SDU University

UDC 004.421:004.93’1

On manuscript rights

AITIMOV ASKHAT KABIDOLLAYEVICH

Analysis and development of algorithm and method for object recognition

6D070400 – Computing Systems and Software

Thesis for the degree of
doctor of philosophy (PhD)

Scientific Advisor,
Doctor of Physical
and Mathematical Sciences,
professor
Amirgaliyeva Saltanat.

Foreign scientific advisor
doctor PhD,
professor,
Krak Iurii

Republic of Kazakhstan
Kaskelen, 2025

CONTENTS

NORMATIVE REFERENCES

This thesis uses references to the following standards:

“Instructions for the preparation of a dissertation and author’s abstract” Ministry of education and science of the Republic of Kazakhstan, 377-3 Zh.

GOST 7.32-2001. Report on research work. Structure and design rules.

GOST 7.1-2003. Bibliographic record. Bibliographic description. General requirements and compilation rules.

GOST 7.32-2017. System of standards of information, librarianship, publishing. Research report. Structure and design rule

ABBREVIATIONS

AI	– Artificial Intelligence
CNN	– Convolutional Neural Network
DT	– Decision Tree
GAN	– Generative Adversarial Network
GDPR	– General Data Protection Regulation
GNB	– Gaussian Naive Bayes
HCI	– Human-Computer Interaction
HIPAA	– Health Insurance Portability and Accountability Act
KNN	– K-Nearest Neighbors
LDA	– Linear Discriminant Analysis
MLP	– Multilayer Perceptron
MSRC	– Modified Sparse Representation based Classification
NB	– Naive Bayes
NCC	– Normalized Cross Correlation
NN	– Neural Network
PCA	– Principal Component Analysis
RBF	– Radial Basis Function
RF	– Random Forest
SGD	– Stochastic Gradient Descend
SOM	– Self-Organizing Map
SRC	– Sparse Representation based Classifier
SVM	– Support Vector Machine

INTRODUCTION

In an era characterized by the proliferation of visual data, the ability to comprehend and interpret the content of images and videos is of paramount significance [1]. Object recognition, a cornerstone of computer vision, plays a pivotal role in this endeavor. It constitutes the bedrock upon which numerous applications, ranging from autonomous vehicles to augmented reality systems, rely for accurate perception and decision-making. The fundamental task of object recognition is to endow machines with the capability to identify and categorize objects within a given visual context, akin to the cognitive abilities of the human visual system [2].

Pattern identification, description, categorization, and grouping provide challenges for Artificial Intelligence (AI), medicine, biology, and other branches of engineering and science. There are numerous definitions of the term "pattern". A pattern is a group of things, happenings, or thoughts that have certain similarities or features. According to Norbert Wiener, the essence of a pattern is an arrangement [3]. It is distinguished by the arrangement of its constituent pieces, rather than by their intrinsic characteristics the pattern is sometimes regarded as the opposite of chaos and a substantially namable item [4]. It can also be defined by a factor shared by multiple instances of the same object. Similarity in fingerprint pictures creates fingerprint patterns; handwriting, audio signals, web pages, and the human face are further examples of patterns [5].

Object recognition remains a challenging problem in computer vision due to various factors that affect the reliability and generalization of recognition systems. According to Bansal et al. in 2021, key challenges include the variability in object appearance, where differences in size, shape, color, illumination, and orientation make it difficult for models to learn consistent representations. Another major issue is scale and resolution, as detecting small objects within large and complex scenes requires algorithms capable of handling multi-scale information effectively. Partial occlusion also hinders detection accuracy when objects are partially hidden by others or by their own parts, while intra-class variability causes significant differences in appearance among objects of the same category. Moreover, Bansal et al. highlight that deep learning approaches often depend on large annotated datasets, and limited training data remain a major obstacle to achieving stable recognition performance [6]. Therefore, the present work focuses on addressing the limitations associated with insufficient data in object recognition tasks.

Limited datasets pose a significant challenge for machine learning models, particularly in object recognition tasks where data diversity and volume are critical for performance. According to Buda, Maki, and Mazurowski in 2018, small and imbalanced datasets often lead convolutional neural networks (CNN) to overfit the majority classes, thereby reducing their ability to generalize effectively to unseen data. [7]. Similarly, Shorten and Khoshgoftaar in 2019 demonstrated that insufficient training data significantly limits the performance of CNN, resulting in degraded recognition accuracy [8].

While there is no universally fixed threshold that defines a “limited dataset,” prior research indicates that datasets with fewer than 500 images per class often fail to generalize effectively and are thus considered insufficient for model training. However, the adequacy of dataset size depends heavily on task complexity—even datasets containing more than 1,000 samples per class may be limited when dealing with fine-grained recognition tasks or high intra-class variability [9]. Furthermore, data quality is as critical as quantity: large datasets with noisy labels, poor annotation, or bias may still functionally behave as limited datasets [10].

In domain-specific applications, such as facial recognition, widely used datasets like FER2013 and LFW contain limited variation in lighting and pose, which diminishes model reliability in real-world conditions [12]. Comparable challenges arise in hand gesture recognition, where Molchanov et al. in 2016 reported a 10–20% decline in model accuracy when trained on small datasets such as SHREC and DHG-14/28 [13]. Likewise, Koller et al. in 2020 emphasized that restricted signer diversity and insufficient gesture classes continue to impede progress in gesture-based recognition systems [14].

In addition, collecting and labeling such datasets is often time-consuming, expensive, and impractical, particularly in specialized or sensitive areas. This persistent dependency on data volume creates a critical bottleneck for achieving consistent model performance under data-scarce conditions. According to recent studies, in security and surveillance domains, legal and operational restrictions often limit both data availability and sharing, making large-scale dataset construction infeasible [15]. Similarly, privacy and ethical considerations in sensitive fields such as biometrics further constrain data collection and dissemination, emphasizing the necessity of developing models that can perform efficiently under limited-data scenarios [16].

In response to these challenges, researchers have introduced various deep learning–based strategies designed to overcome data scarcity. Recent methods include transfer learning, which reuses pretrained network features; generative adversarial networks (GANs), which create synthetic training samples; few-shot learning, which aims to generalize from minimal examples; and federated learning, which enables distributed model training without direct data sharing. For instance, Tomar et al. in 2024 utilized transfer learning with VGG16 and ResNet50 for facial recognition [17], Ghali et al. in 2023 combined data augmentation with CNN and PCA for few-shot setups [18], and Zheng et al. in 2022 implemented federated variational autoencoders with meta-learning for small-sample recognition tasks [19]. Although these approaches have achieved high reported accuracies, the cited authors also consistently identify limitations that restrict real-world applicability.

First, most deep learning–based methods depend on the creation of large quantities of augmented or synthetic data to compensate for limited real samples. Some experiments have produced tens of thousands of artificial images, substantially increasing storage, preprocessing, and training costs. This approach, while improving short-term accuracy, contradicts the objective of designing models that are genuinely effective in low-data environments.

Second, the reliance on complex deep learning architectures leads to high computational and hardware demands. As noted by Tomar et al. in 2024, models based on transfer learning or GANs require powerful GPUs, long training times, and extensive memory resources—factors that make them unsuitable for resource-constrained platforms such as embedded or real-time systems [17, p. 6].

Third, despite strong training performance, generalization and stability issues persist. Models trained heavily on synthetic data may overfit to unrealistic patterns, resulting in inconsistent accuracy when exposed to real-world conditions. Studies such as Ghali et al. in 2023 observed that augmentation improved performance only within the training domain, but failed to guarantee stable results on unseen datasets [18, p. 4].

These findings collectively indicate that although deep learning has revolutionized pattern recognition, it often does so at the expense of computational simplicity, interpretability, and accessibility. In domains such as facial and hand gesture recognition, where data collection is constrained by privacy regulations, annotation cost, and subject diversity, these limitations become particularly evident. Therefore, there remains a strong research need for lightweight, stable, and low-computational algorithms capable of sustaining reliable recognition performance without reliance on massive datasets or specialized hardware. Developing such approaches would not only enhance the sustainability and inclusiveness of AI technologies but also enable their practical deployment in real-world, data-constrained environments supporting broader accessibility and ethical use of recognition systems.

This doctoral thesis focuses on the analysis and development of algorithms and methodologies for object recognition in computer vision. Object recognition refers to the process of detecting, identifying, and classifying objects in digital images or video streams—a fundamental capability in modern AI systems [20]. In recent years, deep learning–based approaches, such as convolutional and transformer networks, have achieved remarkable success in large-scale recognition benchmarks. However, these methods typically rely on massive, well-annotated datasets and extensive computational resources, which limits their applicability in low-data or resource-constrained environments.

To address these challenges, this research emphasizes the study of object recognition under limited-data conditions, using the ORL dataset for face recognition and the Handreader dataset for hand gesture recognition as representative cases. The work examines both classical and modern recognition algorithms, including K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Random Forest (RF), CNN, Support Vector Classifier (SVC), and Sparse Representation Classifier (SRC). In addition, recent advances in deep learning—such as transfer learning, GANs, few-shot learning, and federated learning—are reviewed as representative modern strategies for improving recognition performance under limited data conditions. Building upon this analysis, the study introduces a lightweight algorithm—Principal Component Analysis (PCA) with Modified Triplet Similarity Loss and Projection (PCA-TP)—designed to enhance recognition stability and accuracy while eliminating the need for large datasets or high-performance computing resources.

Problem statement. Considering the challenges discussed above, limited datasets remain a major obstacle to the development of efficient object recognition systems. Deep learning models continue to rely heavily on large, diverse, and well-annotated datasets to achieve stable and accurate performance. However, in many practical domains such as facial and hand gesture recognition, data availability is often restricted due to privacy concerns, high collection costs, and class imbalance. Models trained on small or insufficient datasets tend to suffer from overfitting and weak generalization to unseen samples. Existing approaches aimed at addressing these limitations—such as transfer learning, data augmentation, or generative models—commonly require extensive computational resources or large auxiliary datasets, which makes them unsuitable for data-constrained environments. Therefore, the fundamental problem lies in achieving efficient object recognition performance under limited training data conditions.

Relevance. The relevance of this research lies in the growing demand for reliable and efficient object recognition systems capable of performing under data-constrained conditions. As AI becomes increasingly integrated into domains such as security, healthcare, and human–computer interaction, recognition models must operate effectively even when only limited training data are available. Traditional deep learning approaches often depend on large-scale datasets and high-performance hardware, making them impractical for many real-world applications where data collection is restricted by privacy, cost, or domain specificity. Developing lightweight recognition methods that maintain high accuracy with minimal data is therefore of both theoretical and practical significance. Such approaches not only contribute to the advancement of efficient machine learning but also support broader accessibility, enabling the deployment of recognition technologies in resource-limited environments.

Aim. To develop an algorithm and method for face and hand gesture recognition efficiently capable of achieving high accuracy under limited dataset conditions.

Objectives:

1. To employ limited datasets frequently utilized in research endeavors.
2. To develop a lightweight and efficient recognition approach for face and hand gesture identification that ensures high accuracy and computational efficiency in data-constrained environments.

Research methods:

The research was conducted using a combination of analytical and experimental methods. The analytical component involved the examination of existing approaches to object, face, and hand gesture recognition under limited data conditions, as well as the identification of factors affecting recognition accuracy and generalization. The experimental component focused on developing and evaluating a lightweight recognition algorithm based on PCA, Modified Triplet Similarity Loss, and Projection. For the experimental evaluation, two limited datasets were employed: the ORL face dataset and the HandReader gesture dataset, both commonly used in academic research for small-scale recognition studies. PCA was applied to reduce the dimensionality of input features and eliminate redundant information, while the Modified Triplet Similarity Loss function was used to improve feature discrimination and clustering in the embedding space. Finally, a projection-based classification stage was implemented to enhance recognition stability under limited-data conditions.

Scientific novelty: The scientific novelty of this research lies in the development of a lightweight object recognition algorithm and method that integrate PCA, a Modified Triplet Similarity Loss, and a Projection-based classification stage to ensure accurate performance under limited dataset conditions for face and hand gesture recognition.

Scientific, Practical, and Theoretical Significance of the Study: The theoretical significance of this research lies in the development of a hybrid methodological framework that integrates PCA, a Modified Triplet Similarity Loss, and Projection-based classification into a unified recognition system. This combination expands the theoretical understanding of how dimensionality reduction, metric learning, and projection principles can be jointly optimized to improve recognition accuracy under limited dataset conditions. The study contributes to the theoretical foundations of machine learning by demonstrating how loss function modification and feature-space projection can enhance the discriminative capability of embeddings in data-scarce environments. Moreover, it enriches the methodological tools available to researchers by proposing a lightweight, interpretable recognition architecture that balances performance, computational efficiency, and generalization, an aspect often overlooked in existing deep learning frameworks.

The practical significance of the work is reflected in its direct applicability to face and hand gesture recognition tasks where data availability is constrained. The proposed algorithm can be effectively implemented in systems requiring reliable recognition with minimal computational resources, such as embedded platforms, human–computer interaction interfaces, and security applications. Its use of the ORL and HandReader datasets demonstrates adaptability to small-scale, real-world scenarios, confirming its potential for integration into low-resource recognition systems. The method’s lightweight structure ensures ease of deployment and accessibility, offering practical value for organizations and researchers operating under data or hardware limitations. Overall, the research supports the broader goal of developing efficient and sustainable recognition technologies suitable for diverse and data-constrained environments.

Publications:

1. Sign Language Dactyl Recognition based on machine learning algorithms //Eastern-European Journal of Enterprise Technologies, 2021, Volume 4, Issue:2, pp. 58–72
2. 2D Face Recognition using PCA and Triplet Similarity Embedding // Bulletin of Electrical Engineering and Informatics, 2023, Volume12, Issue: 1, pp 580–586.
3. PCA and Projection based Hand Gesture Recognition // Journal of Theoretical and Applied Information Technology, 2021, Volume 99, Issue:15, pp. 3801–3811
4. An Enhanced Face Recognition Method For Lighting Problem // 2019 15th International Conference on Electronics, Computer and Computation, 2019, Pp. 1-4, (ICECCO)
5. Gesture Recognition Based on Sparse Reconstruction // 2018 14th International Conference on Electronics Computer and Computation, 2018, Pp 206-212 (ICECCO 2018)

6. Comparative Analysis of Recognition Algorithms for Hand Gestures on The Basis of Various Representations of Images // Herald of The Kazakh-British Technical University. 2019; Volume 16, Issue:1 pp. 50-54
7. Information Technology For Sign Language Alphabet Units Modeling And Recognition // Bulletin of Kaznitu 2020;Volume 5, pp 364-371
8. Comparative Analysis Of Face Recognition Algorithms In The Problem Of Visual Identification // Suleyman Demirel University Bulletin: Natural and Technical Sciences 2021; Volume 54, Issue:1, pp 61-67
9. Methods For Handling Frames Of Hand Gesture Recognition // Suleyman Demirel University Bulletin: Natural And Technical Sciences 2021; Volume 54, N. Issue:1 P. 45-51,

Structure of the dissertation. The dissertation consists of an introduction, 3 sections, and a list of sources.

Chapter 1 This chapter delves into the history of object recognition, discussing previous methods, algorithms, and their relevance in contemporary contexts. It explores the challenges faced in object recognition today, particularly in the context of limited datasets. Additionally, the literature review analyzes the evolution of object recognition techniques, highlighting seminal studies and their contributions to the field. It critically examines the limitations of existing methods and identifies gaps in research that necessitate the development of novel approaches to address the challenges of limited dataset recognition.

Chapter 2 The methodology chapter details the datasets used, including ORL for face recognition and Handreader for hand gesture recognition. It discusses the most commonly used algorithms and techniques, including KNN, LDA, RF, SVC, SRC, PCA, Triplet Loss, and Projection. Additionally, it introduces the novel PCA-TP algorithm.

Chapter 3 This chapter presents the experimental setup, including more than 40 different experiments conducted with varying dataset sizes, preprocessing techniques, and algorithm configurations. It discusses the results obtained, highlighting the high performance of the PCA-TP algorithm, achieving 95% accuracy in the Handreader dataset and 10% in the ORL dataset.

Moreover, the chapter provides detailed descriptions of the experimental methodology, including the rationale behind the selection of specific dataset sizes, preprocessing methods, and algorithm configurations. It emphasizes the rigorous nature of the experimentation process, ensuring comprehensive coverage of various scenarios to accurately assess the performance and of the PCA-TP algorithm in object recognition tasks.

Discussion The discussion chapter analyzes and interprets the experimental results, comparing the performance of the PCA-TP algorithm with baseline algorithms. It discusses the implications of the findings and identifies areas for further research.

The thesis's key conclusions are outlined in the last chapter, which also evaluates whether the thesis's goals and objectives were met.

Conclusion In the conclusion chapter, the key findings of the study are summarized, and conclusions are drawn based on the research outcomes. The chapter also discusses the significance of the study and potential future directions in the field of object recognition.

Through this structured approach, the thesis aims to contribute to advancements in object recognition methodologies and provide valuable insights for researchers and practitioners in the field of computer vision.

1 LITERATURE REVIEW

1.1 History of pattern recognition

Since the 1960s, pattern recognition has gained increasing interest. Object recognition, first developed by Duda and his colleagues in 1973, is the study of machine identification for meaning regularities in noisy or complicated situations [21]. Researchers describe object recognition as a field of study whose purpose is to classify objects into several groups or categories. Pattern recognition is a key aspect of the majority of machine intelligence systems that make decisions [22]. In the previous decades, pattern recognition has achieved remarkable results. The most effective techniques include statistical approaches such as parametric and nonparametric Bayesian decision rules, support vector machines (SVM) and boosting algorithms [23]. Typically, the precision of these models is achieved through the use of thoughtfully handcrafted details. In classical approaches, the choice of feature representation has a considerable impact on classification efficiency. Since 2006, the end-to-end deep learning technique, which trains feature and classifier simultaneously from raw data, has emerged as a new cutting-edge solution for numerous pattern recognition problems. Over time, the precision of a great number of tasks has increased dramatically and rapidly. Using CNN and the (handwritten digits of 10 classes) dataset, for instance, it is possible to achieve above 99 percent accuracy without employing conventional manual approaches [24]. Year by year, the precision of a large-scale visual identification of ImageNet 1000-class improved. For example, AlexNet had an accuracy rate of 84.7% in 2012 [25], GoogLeNet reached 93.33%, and ResNet achieved 96.43% [26]. The latest accuracy has already far surpassed that of humans. This type of accuracy improvement and record-breaking occurrence is commonplace for a variety of pattern recognition tasks, including face recognition, voice recognition and identification of handwriting. Accuracy-wise, pattern recognition appears to be a well-solved problem.

1.2 Methods used in pattern recognition

The bulk of conventional procedures are two-step, consisting of a cascade, hand-crafted representation of features followed by pattern categorization. Using feature representation, the raw data should be turned into a feature space that is compact within a class and distinct across classes. First, preprocessing (such as noise reduction and data normalization) is used to minimize variation within a class, whereas feature extraction, which is typically domain-specific, increases variation between classes [27].

Realistically, in order to solve innovative pattern recognition problems, it is necessary to first construct a representation of features, and a good function will significantly reduce the burden of subsequent classifier training. Similar efforts have been undertaken in the fields of iris recognition, gait recognition and action recognition [28].

The second step, following the provision of attributes, is pattern categorization, a substantially broader task. Several textbooks, such as Bishop lay a heavy focus on classification [29]. This process is sometimes referred to as statistical pattern recognition [6, p. 6], and it entails considering numerous distinct issues from a variety of angles. Typically, dimensionality reduction is used to provide a lower-dimensional representation in order to facilitate the subsequent classification task [30]. Another way for selecting features might be considered as a discrete dimension reduction. Then, numerous classic categorization models can be utilized. Bayesian theory [31], which classifies the maximum posterior probability by combining the conditional class density estimate and the prior probability, is the most fundamental. Frequently, machine learning algorithms like Multilayer Perceptron (MLP), radial basis function (RBF), and polynomial networks are used for pattern classification [32]. Also, decision tree (DT) approaches, the classification rule is represented by a tree structure [33]. Kernel approaches [34] were well-liked for extending linear models to nonlinear ones by executing linear operations in a space of greater or even infinite dimension, implicitly translated by a kernel mapping function, SVM [35] being the most prevalent method. By combining the predictions of numerous complimentary models, ensemble methods [36] can further enhance performance. Clustering is frequently used as a tool for unsupervised picture recognition [37]. In two-step approaches, there are typically several options for both feature representation and training classifiers. It is difficult to predict which combination will yield the maximum performance, because different pattern recognition tasks frequently have different optimal configurations based on domain knowledge. In contrast, deep learning approaches [38], study the representation and categorization of features extracted from raw data. Consequently, the explored functions and classifiers are more collaborative in tackling this data-driven problem, which is more adaptable and intuitive than two-step techniques.

Template matching we typically have various options for both feature representation and classifier training in two-step methods. It is difficult to predict which combination will yield the maximum performance, because different pattern recognition tasks frequently have different optimal configurations based on domain knowledge. In contrast, deep learning approaches [39, p. 5], study the representation and categorization of features extracted from raw data. Consequently, the examined functions and classifiers are more cooperative in solving this data-driven problem, which is more flexible and comprehensible than two-step techniques. Although template matching requires a lot of processing, it has become increasingly useful as a result of the development of faster computers. In some cases, rigorous pattern matching is advantageous, but it has a number of downsides. It will fail, for instance, if the templates become corrupted as a result of the rendering process, a change in perspective, or significant class differences between templates. When deformation is difficult to represent or explain explicitly, deformable stencil models [40] or rubber layer deformation [41] can be utilized to compare stencils.

In statistical approach each statistical structure is characterized by d-dimensions or characteristics and regarded like a spot in the d-dimensional area. In a dimensional subspace, one objective is selection of features which makes possible vectors of templates from various classes to occupy small and disjoint areas. Separating templates from distinct classes affects the performance of the presenting area (set of functions). Using a set of train samples for every class, the aim is to build classification rules in the subspace that split class-specific templates. In a statistical decision concept, the selection limits are set by the probability of each class's samples, which must be stated or researched [42]. You may also use a discriminant analyzing classification technique: initially, a parametric shape of the bounding box is produced; next, the boundary of the solution asbestos of the specified type is established by constructing the classification of train samples. Using a criteria like the standard mistake criterion, such limits may be established. The perspective of Vapnik lends credibility to ways of defining separate boundaries: If you have limited knowledge to address an issue, you should attack it directly as opposed to tackling a larger, more general problem first. The offered information may be enough for a clear answer but inadequate for a greater complex intermediate issue [43].

Syntactic technique Numerous recognition issues including complicated patterns favor a hierarchical perspective, in which the pattern is regarded to be constructed of a simple dataset composed of even simpler sub-patterns [44]. Primitives are the simplest/most elementary sub-patterns that can be recognized, and this complicated pattern is defined in terms of these primitives' interactions. During the detection of syntactic patterns, a formal parallel between pattern structure and language syntax is formed. Templates are viewed as language sentences, primitives as the alphabet, and sentences are constructed depending on the grammar. Consequently, a vast variety of complicated patterns can be described with a fewest of grammatical rules. Every template class's grammar should be determined from the provided training examples. Also, to categorize, structural pattern recognition describes how a particular pattern is built from its constituent primitives. This model was used to patterns with a particular structure that may be recorded according to a collection of regulations, like ECG waveforms, textured pictures, and contour analyses [36, p. 187].

Table 1.1 – Pattern Recognition Models

Approach	Representation	Recognition Function	Typical Criterion
Template matching	Samples, pixels, curves	Correlation, distance measure	Classification error
Statistical	Features	Discriminant function	Classification error
Syntactic or structural	Primitives	Rules, grammar	Acceptance error
Neural networks	Samples, pixels, features	Network function	Mean square error

Table 1.1, neural Systems NN are widely used distribution systems, which work with many basic processors with numerous interconnections.

However, the implementation of the syntactic technique faces a number of obstacles, the majority of which are associated with the division of noisy samples(for recognizing data) and the inference of syntax from train [44, p. 22].

Fu suggested attributive grammars, which combine identification of syntactic and statistical patterns. The syntactic technique can result in a combinatorial explosion of research potential, demanding vast training sets and voluminous computational resources [44, p 13].

Certain organizational concepts are tried to be implemented in a network of weighted directed graphs, where the nodes are artificial neurons and the directed edges represent links between neuron outputs and neuron inputs. In addition to computers, these organizational concepts include learning, globalization, flexibility, fault tolerance, and distributed representation. Neural networks (NN) are distinguished by their capacity to learn complex nonlinear I/O connections, apply sequential learning methods, and adapt to new information. A feed-forward network, consisting of RBF networks, and MLP, are the most popular algorithms of NN for image classification applications [45]. These networks are layered and interconnected in a unidirectional fashion. Self-organizing map (SOM) commonly known as the Kohonen network is an additional well-liked network for similar objectives. The network must discover how to adjust its structure and link weights in order to improve its performance on a particular classification or clustering job. NN models for pattern recognition tasks have become more popular due to the accessibility of fast learning algorithms and what seems to be a minimum need for previous knowledge of the relevant subject matter. The term "NN" refers to a new type of nonlinear algorithm that may be used for both feature extraction (through hidden layers) and data classification (e.g. MLP). In order to aid efficient (hardware) implementation, the current methods for feature extraction and classification may be rebuilt as NN architectures [46].

Table 1.2 – Links between Statistical and NN methods

Statistical Pattern Recognition	Artificial Neural Networks
Linear Discriminant Function	Perceptron
PCA	Auto-Associative Network, and various PCA networks
A Posteriori Probability Estimation	MLP
Non-LDA	MLP
Parzen Window Density-based Classifier	RBF Network
Edited K-NN Rule	Kohonen's LVQ

Although the basics of these models may seem to be somewhat dissimilar at first glance, the vast majority of popular NN models are really pretty similar to those used in traditional statistical item identification as shown in table 1.2. In their individual publications, Ripley [47] and Anderson et al. [48] study the link between NNs and the discovery of statistical patterns. It must be mentioned that neuronal networks are amateur statistics... The bulk of NNs hide information from the user. In spite of their apparent similarities, NNs provide a number of benefits that conventional approaches do not. These benefits include a unified approach to the extraction and categorization of features, as well as a flexible way for choosing excellent, somewhat nonlinear solutions. Utilizing NNs also provides a great deal of other advantages [49].

1.3 Relevance of pattern recognition

Pattern recognition has recently been revived due to the emergence of new applications that are not only more complex, but also more computationally intensive [50].

Table 1.3 – Examples of Pattern Recognition Applications

Problem Domain	Application	Input Pattern	Pattern Classes
Bioinformatics	Sequence analysis	DNA/Protein sequence	Known types of genes/patterns
Data mining	Searching for meaningful patterns	Points in multi-dimensional space	Compact and well-separated clusters
Document classification	Internet search	Text document	Semantic categories (e.g., business, sports, etc.)
Document image analysis	Reading machine for the blind	Document image	Alphanumeric characters, words
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective / non-defective nature of product
Multimedia database retrieval	Internet search	Video clip	Video genres (e.g., action, dialogue, etc.)
Biometric recognition	Personal identification	Face, iris, fingerprint	Authorized users for access control
Remote sensing	Forecasting crop yield	Multispectral image	Land use categories, growth pattern of crops
Speech recognition	Telephone directory enquiry without operator assistance	Speech waveform	Spoken words

These applications span multiple fields, demonstrating the versatility and growing importance of pattern recognition in modern data-driven systems. As illustrated in Table 1.3, pattern recognition techniques are applied across diverse domains. Examples include data mining, which focuses on discovering hidden patterns such as correlations and anomalies within large, multidimensional datasets; document classification, which enables efficient text retrieval and organization; financial forecasting, which supports predictive analysis of market behavior; the structure and search of multimedia databases, which facilitates content-based retrieval; and biometrics, which identifies individuals based on physical characteristics such as facial features and fingerprints. Recent developments in affective computing indicate that AI systems are increasingly capable of interpreting and responding to human emotions. Picard's finding paves the way for an application of pattern recognition called emotional computing [51]. Many of these systems' available features have a similar denominator: they must be obtained and developed through data-driven methods, as opposed to being offered by subject matter experts. Many different kinds of systems have this feature in common. Complex and varied data analysis and categorization procedures have been made possible by the fast rise and broad availability of computer power that may expedite the processing of massive datasets. Thanks to the broad availability of computer power and the exponential expansion of computing power, this was realized [52]. At the same time, the availability of enormous datasets and the imposition of strict performance criteria have contributed to an uptick in interest in automated pattern recognition systems. Years of research and development into Pattern Recognition have benefited a wide variety of fields and industries, including AI, computer engineering, nerve biology, medical picture analysis, archaeology, geologic reconnaissance, space navigation, weapons technology, and many others. Computer-assisted diagnosis includes the analysis of DNA sequences, X-ray mammography, and electroencephalogram signals; automatic mail sorting and check processing; human-computer interaction (HCI) and Universal Access in speech recognition; safety-related issues such as face recognition and fingerprint identification[53]. Research in the fields of engineering categorization, aircraft photography, and automation. Make an effort to earn respect among military personnel. The most prevalent real-world examples of object recognition are data processing, segmentation, analysis, computer vision, and stock market analysis. Even if the image is hazy and difficult to detect, a machine-learning algorithm can distinguish dozens of bird species superior to a human [54]. In addition, Neural Talk, an artificial NN that employs pattern recognition for computer vision, can generate real-time explanations of the surrounding environment. Forecasting the stock market is difficult. Even now, there are identifiable and exploitable patterns [55]. Modern investor applications employ AI to provide advisory services to their clients. Several examples include Blumberg, Tinkoff, Kosho, and SofiWealth. In numerous novel applications, it is obvious that there is no common classification technique and that diverse methodologies and approaches must be employed. Consequently, the practice of mixing many perceptual modalities and classifiers is currently widespread in pattern recognition [56].

1.4 Challenges today

Currently, despite significant progress and high recognition rates in various areas, the study of hand gesture recognition faces numerous challenges, including extraction of invariant features, transition models between gestures, minimal sign language recognition units, automatic segmentation of recognition units, recognition approach with scalability about vocabularies, auxiliary information, signer independence, and mixed gesture recognition, among others. In this regard, static gesture recognition based on vision is the current trend in hand gesture recognition and is characterized by the following two technological challenges [57]. Difficulties in target detection arise because the goal is to accurately extract the object of interest—such as a human hand—from an image stream amid diverse and unpredictable visual conditions. In vision-based hand gesture recognition systems, separating the hand region from complex backgrounds remains a major challenge due to varying lighting, clutter, and environmental factors. The recognition process seeks to interpret the high-level semantic meaning conveyed by the posture and dynamic movement of the hand, relying on geometric and motion-invariant feature extraction. However, several intrinsic properties of the human hand make this task complex: the hand is an elastic structure with over twenty degrees of freedom, resulting in substantial variation even among identical gestures, while gestures by different individuals or performed at different times can differ significantly. Furthermore, the hand's non-rigid nature and redundant information—especially between palm and finger features—complicate feature selection and modeling. Projection from three-dimensional space to two-dimensional imagery introduces additional uncertainty depending on viewing angle and illumination. Shadows and surface irregularities further distort visual cues, requiring additional constraints or normalization strategies to achieve reliable recognition [58].

A typical gesture recognition system will usually include a pre-processing step for detection and a classifier for recognition. There are different methods and technologies used for pre-processing and feature extraction. Some approaches apply various filters and algorithms, while others use sensors and extra devices to get more accurate data and results. Because this study focuses on vision-based gesture recognition, we will only address classification processes. Once the hand gesture is detected, an efficient algorithm must be applied to correctly recognize the gesture type and provide the result in minimal time [59].

One of the biggest challenges today in gesture recognition and AI in general is working with limited datasets. Many AI-based models, particularly deep learning models, require massive amounts of labeled training data to perform effectively. However, in real-world applications, acquiring such large datasets is often costly, time-consuming, and in some cases, impossible due to privacy concerns or rare occurrence of certain gestures. In fields like sign language recognition, medical imaging, and HCI, obtaining high-quality datasets remains a fundamental challenge [60].

This work will focus on addressing the issue of limited datasets by investigating methods to improve recognition accuracy with minimal data. Various techniques, including data augmentation, transfer learning, and synthetic data generation, will be explored to enhance model performance even when the available dataset is small. This research will later investigate classification techniques capable of maintaining high recognition accuracy under limited-data conditions

The dataset used in this study consists of static hand postures taken from an existing dataset, which provides an ideal setting to test and analyze model performance under limited data conditions. Various preprocessing and classification algorithms will be applied to measure their efficiency in terms of recognition time and accuracy [61].

AI has been widely used in applications such as image recognition, media classification, object tracking, and gesture recognition, among others. However, the lack of large labeled datasets has made it difficult to develop highly accurate models across various applications. This research aims to contribute to the field by addressing the data scarcity challenge, proposing novel solutions for improving gesture recognition accuracy despite limited training data [62].

The following sections will provide an in-depth discussion of the datasets used, methodology, algorithms compared, and experimental results, emphasizing how limited data impacts recognition performance and the effectiveness of various techniques in overcoming this challenge.

1.5 Limited dataset

A limited dataset refers to a dataset that lacks sufficient quantity, diversity, or quality of data required for training stable machine learning models. In many real-world scenarios, collecting large, high-quality datasets is not feasible due to constraints such as cost, privacy concerns, time limitations, or domain-specific challenges. The availability of high-quality labeled data is often one of the biggest obstacles in developing machine learning models, as the process of data collection and annotation can be labor-intensive and expensive. In cases where obtaining a large dataset is impractical, researchers must adopt alternative techniques to train models effectively. Despite these limitations, machine learning models must still be trained efficiently, requiring researchers and engineers to develop strategies that improve model accuracy with minimal data while ensuring that the model generalizes well to unseen scenarios [63].

Limited datasets are particularly common in fields like medical imaging, satellite sensing, and cybersecurity, where collecting data is either expensive or infeasible [64]. In medical imaging, for example, privacy laws such as General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA) restrict data sharing, making it difficult to obtain large annotated datasets for training AI models. Similarly, in satellite sensing applications, obtaining high-resolution images is costly and dependent on weather conditions, which can limit the availability of data. In cybersecurity, fraud detection relies on rare instances of fraudulent transactions, leading to highly imbalanced datasets that challenge model performance. As a result, machine learning practitioners must employ various techniques to compensate for the lack of large datasets and enhance model performance [65].

Effective methods for handling limited datasets include data augmentation, transfer learning, few-shot learning, self-supervised learning, and synthetic data generation, all of which help models learn meaningful representations despite insufficient training data [66]. Data augmentation techniques such as image rotation, flipping, and adding noise can artificially expand small datasets, improving model generalization. Transfer learning allows models to leverage pre-trained knowledge from large datasets and apply it to a new, limited dataset, reducing the need for extensive labeled data. Few-shot learning and self-supervised learning are powerful techniques that enable models to learn from very few labeled examples by identifying patterns within unlabeled data. Additionally, synthetic data generation using techniques like GANs helps create artificial training samples that mimic real-world data, reducing dependency on real datasets [63, p. 4].

The growing importance of these techniques has led to increased research in handling limited datasets efficiently. Researchers are constantly improving data-efficient machine learning models that can learn effectively from small datasets while maintaining high accuracy. As machine learning expands into various industries, the demand for efficient solutions to work with limited datasets is increasing. In many cases, developing AI systems that require less data for training can help companies reduce costs, accelerate development cycles, and deploy AI models faster. By employing strategies like active learning, semi-supervised learning, and domain adaptation, researchers can maximize the performance of AI models even when dealing with a scarcity of data. Consequently, limited datasets are no longer seen as a major constraint but as an opportunity to explore new, innovative approaches to learning with fewer examples [63, p. 6].

1.5.1 Barriers to Large-Scale Data Collection

Large-scale data collection requires extensive financial resources for data labeling, storage, and processing. The process of manually labeling data, especially in specialized fields such as medical imaging, requires domain experts whose time is costly. In medical diagnostics, for instance, radiologists and pathologists must carefully annotate medical scans and histology slides, which can take hours or even days per dataset. This makes it impractical to obtain large datasets without incurring substantial costs. Additionally, collecting data in certain fields, such as satellite imaging or environmental monitoring, often involves the use of expensive equipment, satellites, or drones, further increasing the financial burden. Storing and maintaining large datasets also requires significant computational resources, including cloud storage, database management, and continuous monitoring. These costs can be prohibitive for smaller research groups, startups, and academic institutions. To address this issue, researchers have explored alternative techniques to reduce dependency on large datasets while maintaining high model performance. Data augmentation techniques, such as flipping, rotation, color adjustments, and noise addition, artificially expand datasets, allowing models to learn more diverse representations from fewer samples [67]. Another widely used approach is transfer learning, where models pre-trained on large datasets are fine-tuned on smaller datasets, significantly reducing the amount of data needed for training. By leveraging these techniques, researchers can achieve comparable accuracy while minimizing the costs associated with data collection and processing [68].

Sensitive data, such as patient records and financial transactions, cannot be shared freely due to strict privacy laws such as the GDPR and the HIPAA. These regulations ensure that personally identifiable information is not misused or exposed, but they also create challenges in developing robust machine learning models that require large amounts of labeled data [69]. In the medical field, for instance, hospitals and research institutions must anonymize and identify patient records before they can be used in AI research, which limits the availability of high-quality labeled datasets. In financial services, datasets related to fraud detection and credit risk assessment are often restricted due to concerns over security breaches and data misuse. These legal restrictions make it difficult to create large, diverse datasets for training machine learning models while maintaining compliance with privacy laws [70]. To overcome these limitations, researchers have developed privacy-preserving techniques such as differential privacy and federated learning. Differential privacy introduces controlled noise into datasets to prevent the identification of individual records while maintaining overall statistical properties. Federated learning allows models to be trained on decentralized datasets, meaning data remains on local devices and is never shared directly, ensuring user privacy. These techniques enable organizations to develop AI models without violating data protection regulations, making it possible to train effective models on limited data while preserving confidentiality [71].

Training deep learning models on large datasets requires high computational resources and takes significant time. Deep learning models, particularly CNNs and transformers, rely on vast amounts of training data to learn complex patterns and representations. However, training on large datasets often requires powerful hardware, including high-performance GPUs or TPUs, and extensive processing time, sometimes taking days or weeks to complete. This results in high energy consumption and increased operational costs, making large-scale model training impractical for many organizations. When working with limited datasets, training is much faster, reducing computational costs and making AI development more accessible. Models trained on smaller datasets require fewer hardware resources, allowing researchers to conduct experiments with standard computing setups [72]. Various techniques have been introduced to help models generalize better even with smaller datasets. Few-shot learning, for example, enables models to make accurate predictions using only a handful of training examples, reducing the need for extensive labeled data. Self-supervised learning leverages large amounts of unlabeled data to pre-train models, which can then be fine-tuned on smaller labeled datasets. These approaches allow AI systems to be deployed more quickly and cost-effectively, making machine learning more scalable for real-world applications where data availability is a limiting factor [73].

Real-world datasets are often imbalanced, where some classes have far fewer samples than others, leading to biased models that perform poorly on underrepresented categories. For example, fraud detection datasets contain very few fraudulent transactions compared to the overwhelming number of legitimate transactions [74]. In healthcare, datasets for rare diseases have significantly fewer samples than those for common conditions, making it challenging for models to recognize and classify rare cases accurately [75]. Similarly, in speech recognition, languages with lower digital presence have far fewer training samples than widely spoken languages such as English or Mandarin [76]. This imbalance can cause models to become biased toward majority classes, leading to poor performance on minority-class predictions. Addressing class imbalance is a critical challenge in machine learning, and several techniques have been developed to mitigate its impact. One commonly used method is SMOTE (Synthetic Minority Over-sampling Technique), which generates synthetic samples for underrepresented classes by interpolating between existing data points. Another approach is cost-sensitive learning, where models assign higher penalties to misclassifications in minority classes, encouraging better representation. Additionally, ensemble learning methods, such as bagging and boosting, improve model performance by training multiple classifiers on different subsets of data. By implementing these techniques, machine learning models can achieve better accuracy and generalization in real-world scenarios, even when faced with highly imbalanced datasets [77].

1.5.2 Domains and Techniques for Managing Limited Data

Limited datasets present significant challenges across multiple fields, requiring specialized approaches to improve machine learning model performance. Some domains are more affected than others due to factors such as privacy regulations, data collection costs, environmental constraints, and the rarity of certain phenomena. In fields like medical imaging, autonomous vehicles, natural language processing, satellite imaging, cybersecurity, and robotics, the availability of large-scale labeled datasets is often restricted, making it difficult to train high-performing AI models without advanced techniques [78].

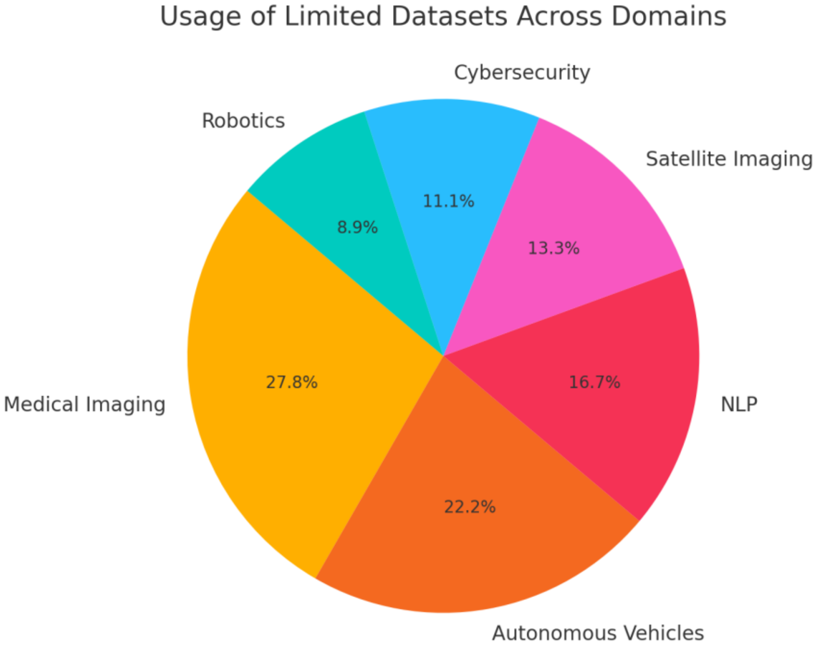

Figure 1.1 - Usage of limited datasets across domains

One of the most critical areas impacted by limited datasets is medical imaging, where patient data is difficult to obtain due to strict privacy laws and the cost of expert annotation. Healthcare data is highly sensitive, and regulations such as the GDPR and HIPAA impose stringent restrictions on data sharing [79]. Additionally, medical datasets require expert labeling by radiologists, pathologists, or clinicians, which increases the cost and time required for dataset creation [80]. To address this, researchers use techniques like data augmentation, transfer learning, and GANs to generate synthetic medical images and expand the dataset without violating privacy laws [81]. GANs, in particular, have been used to synthesize high-quality MRI and CT scans, enabling better training for diagnostic AI models even when real patient data is scarce [82].

In the autonomous vehicle industry, collecting real-world driving data is both expensive and dangerous. Self-driving cars rely on vast amounts of labeled video and sensor data to learn how to navigate in various environments [83]. However, collecting diverse driving scenarios in real-world conditions is time-consuming and costly, as vehicles must be driven for millions of miles to encounter rare but critical situations such as near-collisions, pedestrian crossings, and extreme weather conditions [84]. To overcome these challenges, simulation-based data augmentation and self-supervised learning (SSL) have been widely adopted. Autonomous vehicle companies such as Tesla and Waymo use synthetic environments to simulate complex driving scenarios, allowing AI models to learn from virtual experiences without real-world risk [85].

Similarly, natural language processing (NLP) faces significant data limitations, particularly in less commonly spoken languages and specialized fields such as legal and medical text processing [86]. Large datasets like those used for GPT-based models or BERT are readily available for widely spoken languages like English, Mandarin, and Spanish, but many languages, especially indigenous or low-resource languages, lack sufficient labeled data for effective training [87]. This problem also extends to domain-specific applications, where text datasets related to legal documents, scientific literature, or technical manuals are scarce. To address these challenges, researchers have implemented few-shot learning and pre-trained language models, enabling AI to generalize from a few examples rather than requiring extensive labeled datasets [88].

In the field of satellite imaging, data collection is constrained by environmental factors and high operational costs. Satellite imagery is crucial for applications such as climate monitoring, agricultural analysis, and disaster response, but acquiring high-resolution satellite images is expensive and often limited by weather conditions [89]. Additionally, the manual labeling of satellite images requires experts in geospatial analysis, land classification, and environmental science, making it difficult to create large, labeled datasets [90]. Synthetic data generation, domain adaptation, and transfer learning have been widely used to improve model performance in satellite imaging applications. By generating synthetic satellite images or adapting models trained on different but related datasets, researchers can build robust AI models despite limited labeled data [91].

Cybersecurity and fraud detection also suffer from imbalanced datasets, where malicious activities such as fraudulent transactions, cyberattacks, or data breaches occur far less frequently than normal activity. Machine learning models trained on raw cybersecurity datasets tend to become biased toward majority (non-fraudulent) classes, making it difficult to accurately detect attacks. To combat this issue, anomaly detection and semi-supervised learning techniques are applied, allowing AI models to recognize rare but critical security threats without requiring large amounts of labeled fraud data. These techniques enable AI to flag unusual patterns in network traffic, financial transactions, or system behavior, improving the detection of fraud and cyber threats [92].

Another field heavily affected by limited datasets is robotics, where data collection is constrained by real-world physical interactions. Training robots to perform tasks such as object manipulation, navigation, or industrial automation requires extensive trial-and-error data, which is difficult to obtain in large quantities. Unlike digital data, which can be generated and labeled automatically, robotic learning requires real-world interactions, making it costly and time-consuming to build large datasets [93]. To overcome this challenge, researchers use imitation learning and reinforcement learning, where robots learn from a small number of expert demonstrations or simulated environments before being deployed in real-world applications [94].

To address the widespread issue of limited datasets, researchers have developed various machine learning techniques that allow models to achieve high accuracy even when training data is scarce. Some of the most commonly used methods include data augmentation, transfer learning, synthetic data generation, few-shot learning, self-supervised learning, and anomaly detection, each of which has its strengths and best-suited applications [95].

Table 1.4 – Challenges and Techniques for Addressing Limited Data Across Domains

Domain	Challenges	Common Techniques Used
Medical Imaging	Limited patient data, privacy issues	GANs, Data Augmentation, Transfer Learning
Autonomous Vehicles	Difficult to collect real-world scenarios	Simulation-based Augmentation, SSL
Natural Language Processing	Language diversity, limited labeled text	Few-Shot Learning, Pre-trained Models
Satellite Imaging	Expensive image acquisition, limited labels	Synthetic Data, Domain Adaptation
Cybersecurity	Rare fraud cases, limited attack data	Anomaly Detection, Semi-Supervised Learning
Robotics	Data collection in physical environments	Imitation Learning, Reinforcement Learning

As shown in Table 1.5, these methods address data scarcity in different ways depending on the domain and data type. Among them, data augmentation has been one of the earliest and most widely used strategies, dating back to the 1990s. It artificially expands datasets by applying transformations such as flipping, rotation, cropping, and noise addition to existing samples [96]. This technique is particularly effective in image classification and natural language processing (NLP) tasks, where introducing controlled variation enhances model robustness. However, while traditional augmentation improves diversity, it cannot generate truly novel data. This limitation has led to the emergence of synthetic data generation techniques, especially those based on GANs since 2014, which have shown great promise in domains like medical imaging and cybersecurity [97].

Transfer learning, introduced in 2006, has become a key strategy for training models with limited data [98]. By leveraging pre-trained models on large datasets, AI models can be fine-tuned on smaller datasets to achieve high accuracy with minimal labeled data. This technique is particularly useful in speech recognition, NLP, and medical applications, where pre-trained models can be adapted to domain-specific tasks [99].

Few-shot learning, introduced in 2015, has revolutionized NLP and rare-class classification tasks [100]. Unlike traditional machine learning, which requires thousands of examples per class, few-shot learning enables models to generalize from just a handful of labeled examples. This is particularly useful in low-resource languages, rare disease diagnosis, and fraud detection, where labeled data is difficult to obtain [101].

Self-supervised learning (SSL), which gained popularity in 2016, allows models to leverage large amounts of unlabeled data to learn useful representations before being fine-tuned with minimal labeled samples. SSL has been particularly successful in speech recognition and NLP, where large-scale pre-training on raw text or audio data significantly improves performance [102].

Finally, anomaly detection, introduced in 2020, has become a vital technique in fraud detection and cybersecurity, where traditional machine learning methods struggle with highly imbalanced datasets [103]. By learning patterns of normal behavior and identifying deviations, anomaly detection algorithms can detect rare but critical security threats with minimal labeled data [104].

Table 1.5 – Popular Methods for Working with Limited Datasets

Method	Year introduced	Best for tasks
Data Augmentation	1990s	Image classification, NLP
Transfer Learning	2006	Model fine-tuning, speech recognition
Synthetic Data Generation (GANs)	2014	Medical imaging, security applications
Few-Shot Learning	2015	NLP, rare class classification
Self-Supervised Learning	2016	Speech recognition, NLP
Anomaly Detection	2020	Fraud detection, cybersecurity

Overall, while limited datasets present a significant challenge across various domains, modern machine learning techniques provide effective solutions to mitigate data scarcity.

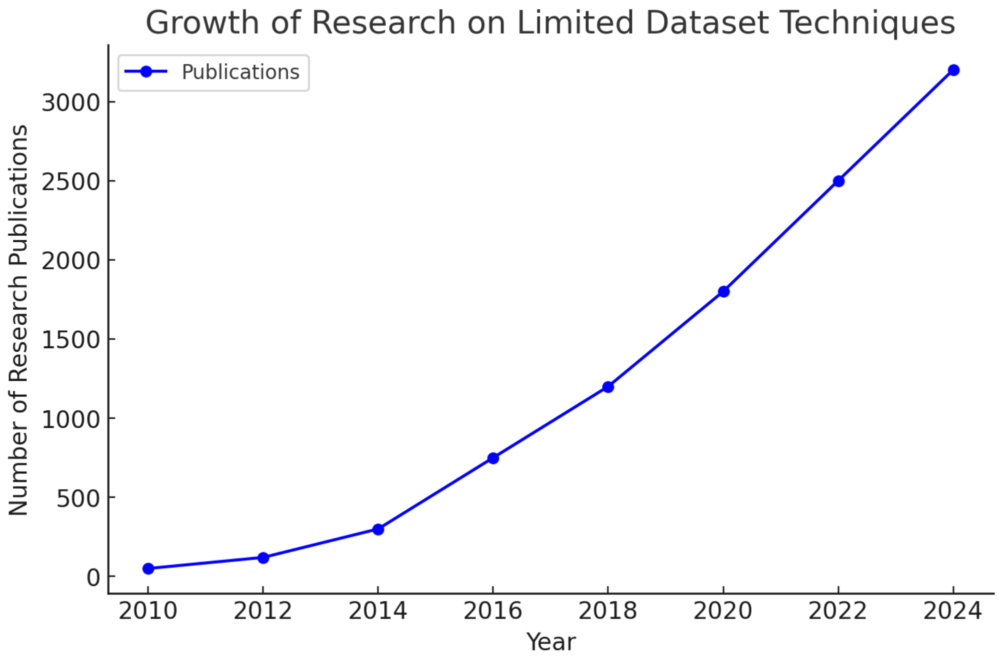

Figure 1.2 – Growth of research on limited dataset techniques

By leveraging data augmentation, transfer learning, synthetic data generation, and self-supervised learning, researchers can develop AI models that achieve high accuracy despite the lack of large-scale labeled datasets. These approaches continue to evolve, with ongoing research focusing on improving model efficiency, reducing dependence on labeled data, and enhancing generalization in real-world applications [105]. The figure 1.2 illustrates the growth of research publications on limited dataset techniques from 2010 to 2024, showing a significant upward trend in academic interest and advancements in this field. The number of publications has increased exponentially, reflecting the growing importance of working with limited datasets in machine learning and AI. Between 2010 and 2014, research on limited dataset techniques remained relatively low, with fewer than 300 publications per year. This period reflects an era when traditional machine learning models relied heavily on large labeled datasets, and methods for overcoming data limitations were not widely explored. During this time, most research focused on basic data augmentation techniques and small-scale transfer learning models, but the demand for solutions to handle limited data was not as urgent [106]. From 2014 to 2018, the number of research publications surged dramatically, reaching over 1,200 publications by 2018. This increase corresponds with the rise of deep learning models, such as CNNs and recurrent neural networks (RNNs), which require vast amounts of labeled training data. As machine learning applications expanded into fields like medical imaging, cybersecurity, and autonomous systems, researchers began investigating ways to improve model performance in data-scarce environments. During this period, GANs were introduced in 2014, revolutionizing synthetic data generation and enabling significant progress in areas where real-world data collection is challenging [107]. Between 2018 and 2022, research publications nearly doubled, surpassing 2,500 publications in 2022. This explosive growth reflects the adoption of self-supervised learning, few-shot learning, and meta-learning techniques, which allowed AI models to learn efficiently from fewer labeled examples. Pre-trained models such as BERT [108] and GPT-3 [109] further fueled this trend, demonstrating that AI systems can generalize well even with limited labeled data when trained using large-scale unsupervised datasets. From 2022 to 2024, research activity in this domain has continued to accelerate, reaching over 3,200 publications in 2024. The rapid increase in publications suggests that handling limited data remains a crucial challenge in AI research, particularly in medical AI, financial fraud detection, and edge computing applications. Techniques like foundation models, federated learning, and AI-driven data augmentation have gained prominence, pushing the boundaries of what machine learning can achieve with limited datasets [110]. This overall trend highlights the growing necessity of effective machine learning solutions for small datasets, driven by privacy regulations, high data collection costs, and real-world constraints. As AI systems continue to be deployed in sensitive domains such as healthcare, finance, and security, researchers will likely explore even more advanced approaches, such as synthetic data generation, self-adaptive learning, and zero-shot learning, to address the persistent issue of data scarcity [111].

In conclusion, the exponential growth in research on limited dataset techniques underscores the critical role of innovative machine learning methods in overcoming data scarcity challenges. This field will remain highly relevant in the coming years, as AI applications continue to expand into areas where data is difficult to obtain.

1.6 Traditional Algorithms for Object Recognition

Before the emergence of deep learning, traditional machine learning algorithms formed the foundation of object recognition research. These classical approaches primarily relied on handcrafted feature extraction and statistical classification, offering interpretable and computationally efficient solutions for pattern and object recognition. Several studies have highlighted algorithms such as KNN, SVM, RF, Naive Bayes (NB), DT, and LDA as among the most widely adopted in various image classification and object recognition tasks [112]. These algorithms served as the basis for early developments in facial, gesture, and texture recognition systems due to their simplicity, reliability, and relatively low computational cost.

Each of these methods follows a distinct learning principle. KNN, one of the simplest non-parametric classifiers, determines the class of an unknown sample based on the majority label of its k nearest neighbors in feature space, making it intuitive but highly sensitive to noise and feature scaling. SVM constructs optimal separating hyperplanes between classes, offering strong generalization when the data is linearly separable or when using appropriate kernel transformations for nonlinear problems. DT and RF rely on recursive partitioning of features, with RF improving performance through ensemble averaging. NB, grounded in probabilistic reasoning, assumes independence among features and is particularly effective when this assumption approximately holds. LDA, in turn, seeks linear combinations of features that maximize class separability, serving as both a dimensionality reduction and classification method [113].

Despite their wide adoption and historical importance, these algorithms exhibit several limitations when applied to complex, high-dimensional visual data. Most depend heavily on handcrafted feature design and perform poorly in recognizing objects under varying illumination, pose, or occlusion. Their ability to generalize is restricted by the quality of manually engineered features rather than by data-driven learning. Furthermore, traditional algorithms often struggle with scalability in large datasets and fail to capture nonlinear relationships inherent in visual patterns. Nonetheless, they remain essential benchmarks for evaluating new methods due to their interpretability, low resource requirements, and effectiveness in small or well-structured datasets [114].

In this research, traditional algorithms are utilized as baseline models for comparative experiments in face and hand gesture recognition, providing a reference point for assessing the improvements introduced by the proposed hybrid algorithm.

1.7 Trending Deep Learning Algorithms for Object Recognition

Deep learning has revolutionized the field of object recognition, marking a decisive shift from handcrafted feature engineering to data-driven representation learning. Since the introduction of CNN in 2012, deep architectures have demonstrated remarkable capabilities in automatically extracting hierarchical features from raw visual data. These methods have achieved state-of-the-art results in tasks such as image classification, facial recognition, gesture detection, and scene understanding [115]. Recent studies have identified deep learning–based approaches such as CNNs, transfer learning, data augmentation, few-shot learning, federated learning, and GANs as the most widely adopted in modern object recognition, especially under limited-data conditions. These developments have significantly enhanced model accuracy and generalization, although they continue to face challenges related to data dependency, computational cost, and overfitting [116]. The emergence of CNNs fundamentally transformed object recognition by replacing handcrafted feature extraction with automated learning from large-scale data. Networks such as AlexNet, VGG, and ResNet demonstrated that deeper architectures with convolutional layers could hierarchically learn low-level edges, textures, and high-level semantic patterns. This concept was further refined by the introduction of residual and inception modules, which improved gradient propagation and network stability. More recently, Vision Transformers (ViTs) have expanded the deep learning paradigm by applying self-attention mechanisms to visual data, enabling long-range dependency modeling and superior scalability. These innovations have become the core of most modern recognition systems and the foundation upon which data-efficient methods such as transfer learning and few-shot learning are built [117].

Transfer learning has emerged as one of the most influential strategies for addressing the limited-data problem. Instead of training from scratch, models pre-trained on large datasets like ImageNet are fine-tuned on smaller, domain-specific datasets. This approach leverages previously learned feature representations, significantly reducing the need for large annotated samples. El Saj et al. (2023) achieved 98.53% accuracy using ResNet18 under transfer learning, while Tomar et al. (2024) obtained 95.18% accuracy with a hybrid setup that combined data augmentation and transfer learning using VGG16 and ResNet50. Such results demonstrate that reusing learned visual representations enables deep networks to perform effectively even when data availability is limited. However, transfer learning requires careful tuning of learning rates and layer freezing strategies, as models pre-trained on general-purpose datasets may not always adapt well to specialized domains with unique visual characteristics [118, 17, p. 5].

Data augmentation represents another critical advancement in deep learning, addressing data scarcity by artificially expanding existing datasets. This technique introduces controlled variations such as flipping, rotation, scaling, and cropping to improve generalization and prevent overfitting. Ghali et al. (2023) reported 98% accuracy by combining CNNs with PCA and few-shot learning enhanced through augmentation [18, p.4], while Parsai et al. (2022) achieved 99.7% accuracy using CNN and SVM models trained with augmented data [119]. Despite these promising outcomes, augmentation-based methods are not without limitations. When excessive or unrealistic transformations are applied, models may overfit to artificial patterns, a phenomenon referred to as synthetic overfitting. Researchers observed that while augmentation improved training performance, the resulting models often failed to generalize effectively to real-world data. Furthermore, large-scale augmentation processes can be computationally demanding. Some experiments have generated between 6,800 and 100,000 augmented images, substantially increasing training time, GPU usage, and storage requirements, which restricts their applicability in resource-limited environments [120].

The development of GANs has further advanced data augmentation by enabling the creation of synthetic yet realistic samples that mimic real-world data distributions. GANs are particularly useful in medical imaging and cybersecurity, where data collection is restricted due to privacy or rarity. However, while they can improve data diversity, GANs also pose challenges related to training instability, high computational cost, and potential bias amplification. When synthetic data dominates the training process, the model may perform well on generated samples but poorly on natural ones, compromising real-world applicability [121].

Recent years have also seen the growing adoption of few-shot and federated learning, which aim to address data scarcity and privacy concerns from different angles. Few-shot learning enables models to learn from only a handful of examples per class by leveraging meta-learning or prototype-based similarity mechanisms. This approach is particularly valuable for rare-class recognition problems, such as sign language or medical image classification, where labeled data is inherently limited . In parallel, federated learning provides a privacy-preserving framework by allowing multiple devices or institutions to collaboratively train a model without sharing raw data [122]. Zhang et al. (2022) demonstrated the effectiveness of this concept through a federated variational autoencoder (VAE) combined with meta-learning, achieving 99.96% accuracy in distributed object recognition tasks. These methods represent a significant step toward decentralized and data-efficient AI systems [19, p. 6]. Despite their success, modern deep learning algorithms face persistent challenges that limit their practicality in constrained environments. One major issue is synthetic overfitting, where excessive augmentation or synthetic sample generation leads to poor generalization. Another challenge is computational intensity where most state-of-the-art deep models require substantial hardware resources, long training times, and large energy consumption, which restricts their deployment in embedded or mobile systems. Additionally, generalization risk remains a concern, as models trained primarily on synthetic or pre-augmented data may fail to adapt to real-world variations in lighting, pose, or occlusion. As highlighted in recent research, while these models achieve impressive benchmark accuracies, their dependency on massive datasets and high-performance hardware makes them less suitable for real-world low-resource scenarios [123]. Overall, trending deep learning algorithms have transformed the landscape of object recognition through automated feature learning and hierarchical representation. Their reliance on large-scale labeled data and computational infrastructure continues to present a barrier to scalability and accessibility. These limitations emphasize the necessity for hybrid and lightweight solutions that can maintain accuracy while minimizing data and computational requirements. The integration of classical dimensionality reduction and projection techniques with discriminative embedding learning represents a promising direction toward achieving stability and efficiency in limited-data conditions. This motivation underpins the development of the proposed hybrid algorithm presented in this dissertation, which seeks to bridge the gap between traditional interpretability and modern deep learning performance.

2. RESEARCH METHODOLOGY

2.1 Datasets

In this chapter, the HandReader dataset is employed as a controlled and limited-data environment to evaluate the performance of both traditional machine-learning techniques and hybrid recognition models. Its publicly available structure, balanced composition, and thorough documentation ensure the reproducibility of experiments and enable consistent benchmarking across various feature-extraction and classification methods, providing a reliable foundation for comparative analysis.

2.1.1 Hand Reader Dataset

The HandReader Dataset, introduced by Ghassem Tofighi et al. (2013) at Ryerson University, is an openly available dataset designed for static hand-posture recognition tasks. The dataset was first presented in their paper “Hand Posture Recognition Using K-NN and Support Vector Machine Classifiers Evaluated on Our Proposed HandReader Dataset” and is accessible through GitHub (https://github.com/tofighi/Hand-Reader-Dataset/tree/master).

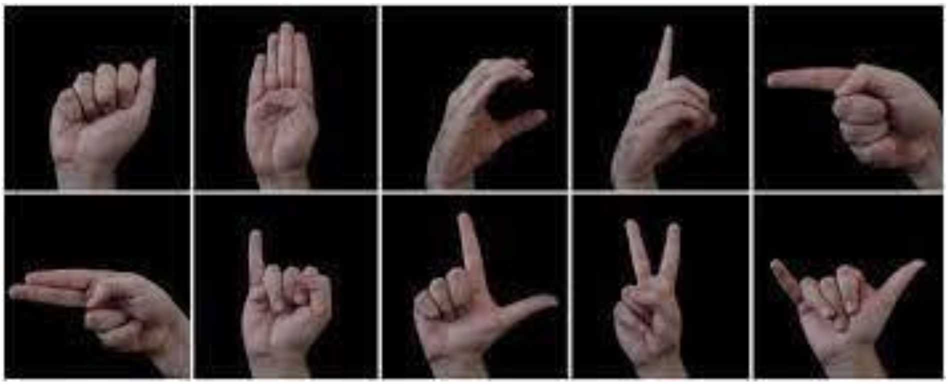

Figure 2.1 – The Hand Reader Dataset Sample

HandReader contains 500 images representing 10 distinct static American Sign Language (ASL) postures: A, B, C, D, G, H, L, I, V, and Y. These gestures were captured from 50 male and female participants positioned in front of a uniform camera setup. Each class comprises 50 samples, all recorded in JPEG format at a resolution of 320 × 240 pixels. All images were acquired against a dark background to simplify segmentation and reduce background interference. However, the creators deliberately introduced variations in illumination to simulate natural lighting differences. This combination of uniform composition and moderate variability makes the dataset well-suited for evaluating the robustness of preprocessing and classification algorithms under limited and slightly noisy conditions. The dataset’s design emphasizes appearance-based static hand recognition, focusing on shape and contour rather than motion or temporal cues. Similar datasets have historically been used to benchmark algorithms for human–computer interaction, sign-language translation, and gesture-based robotic control. In comparison to other gesture datasets, HandReader remains compact but methodologically clean, making it ideal for evaluating algorithms that must generalize from small, homogeneous datasets[124].

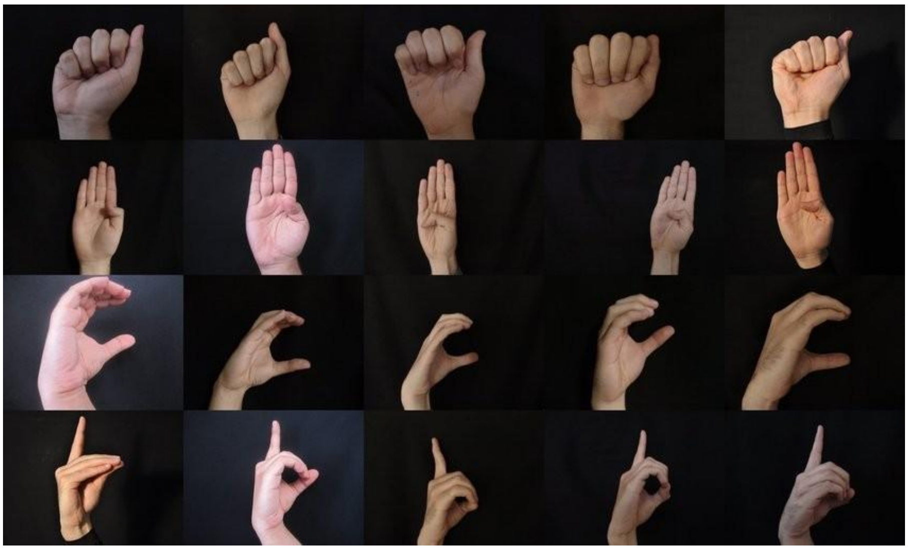

Figure 2.2 – Different images of same sign posture

In this research, the HandReader dataset is used as a controlled, limited-data environment to test both traditional machine-learning and hybrid recognition approaches. Its public availability, balanced structure, and clear documentation support reproducible experimentation and allow for consistent benchmarking of feature-extraction and classification methods.

2.1.2 ORL dataset

The ORL Face Dataset—also referred to as the AT&T Face Database—is one of the most widely recognized benchmark datasets for face recognition research [125]. It was created by the Olivetti Research Laboratory (ORL) in Cambridge, U.K., in collaboration with AT&T Laboratories, and first introduced by Samaria and Harter in 1994. Since its release, it has become a foundational resource for developing, validating, and comparing algorithms in computer vision and pattern recognition. The dataset is publicly available through the official ORL archive at https://cam-orl.co.uk/facedatabase.html [126].

The dataset consists of 400 grayscale facial images of 40 individuals, with 10 samples per subject. Each image has a spatial resolution of 112 × 92 pixels, taken under controlled lighting and a uniform dark background. Despite this controlled setup, there are natural variations in facial expression, pose, and details—such as the presence of glasses, different facial hair, and subtle head rotations. These controlled but realistic variations make ORL an effective testbed for developing and evaluating algorithms that must perform robustly under constrained data conditions [127].

Figure 2.3 – The ORL Dataset sample

Figure 2.4 – 10 different images of 3 individuals

One of the defining features of the ORL dataset is its small sample size, which simulates real-world conditions where only a few labeled images are available per subject. Consequently, it has been extensively employed in evaluating dimensionality-reduction and subspace-based learning methods such as Principal PCA, LDA, and Independent Component Analysis (ICA). These traditional methods often use ORL to assess the relationship between dataset size, feature representation, and recognition accuracy [128]. In recent years, the ORL dataset has remained relevant for testing deep learning and hybrid architectures in low-data environments. Modern studies continue to apply CNN, transfer learning, and embedding-based loss functions on ORL to benchmark lightweight and data-efficient recognition models. Its compact size allows researchers to explore the impact of limited training data on model generalization without requiring large computational resources [129]. The ORL dataset's accessibility, consistent structure, and well-documented characteristics have made it a canonical benchmark in both classical and deep learning–based recognition research. It is available publicly through the University of Cambridge database archives and incorporated into various machine learning frameworks such as scikit-learn, where it continues to serve as a standard dataset for evaluating facial recognition techniques.

2.2 Preprocessing

Preprocessing is a fundamental stage in any pattern recognition or computer vision pipeline, serving as the bridge between raw image acquisition and feature extraction. The primary objective of preprocessing is to enhance the quality of input images, remove irrelevant variations, and standardize data so that subsequent algorithms such as feature extraction, dimensionality reduction, and classification can operate more effectively [130]. In image-based recognition, preprocessing helps to mitigate the influence of illumination differences, sensor noise, scale variations, and background interference, thereby improving both model stability and recognition accuracy. Numerous studies have demonstrated that well-designed preprocessing can significantly improve the performance of object recognition models, particularly when the dataset is limited or collected under varying environmental conditions [131]. Without adequate preprocessing, even powerful feature extraction methods such as PCA or deep learning embeddings may struggle to generalize due to inconsistencies in raw pixel intensity distributions.

In this study, several preprocessing techniques were applied to the ORL and Hand Reader datasets to ensure that the data were consistent and suitable for the subsequent feature extraction and learning stages. The choice and sequence of preprocessing operations depended on the characteristics of each dataset.

The first step involved grayscale conversion, where each RGB image was transformed into a single-channel grayscale image using luminance-based weighting. This step simplified the data representation by reducing three color channels into one intensity channel while preserving essential information about object shape and texture. Such simplification decreases computational complexity and eliminates color variations that are irrelevant to object structure. However, a drawback of this approach is the loss of color information, which could be useful in applications where color is a discriminative feature, such as in natural image segmentation or color-based object detection.

To further enhance image quality, histogram equalization was applied to redistribute pixel intensity values and improve contrast, especially in images captured under uneven or low lighting conditions. This method improves global contrast and enhances the visibility of features, helping to minimize the influence of lighting differences between samples. Nevertheless, it can occasionally amplify noise in regions with uniform brightness or distort brightness relationships in already well-lit images, so its use was carefully controlled.

Another essential step was cropping, which isolated the region of interest—the face or the hand—while removing unnecessary background areas that could introduce noise or bias during feature extraction. Cropping not only reduces irrelevant data but also ensures that the algorithm focuses on the object itself, improving computational efficiency. However, this method requires careful execution, as incorrect cropping—either manual or automatic—can remove valuable contextual information or distort the proportions of the target object.

In some cases, particularly for the Hand Reader dataset, thresholding was employed to convert grayscale images into binary representations that emphasize shape boundaries and simplify contour structures. This technique is highly effective for silhouette-based analysis and gesture recognition, as it highlights the main object outlines and reduces feature complexity. Yet, thresholding is also sensitive to illumination and shadows and can result in the loss of fine texture details that might otherwise contribute to object differentiation.

To reduce high-frequency noise, Gaussian filtering was used. By convolving each image with a Gaussian kernel ($\sigma \approx 0.02 \times$ image size in this study), the method smoothed out noise caused by surface texture and lighting variations. This step was particularly important for the Hand Reader dataset, as it helped stabilize gradient-based features and improved generalization in the training process. However, excessive smoothing can blur important edges and reduce the level of detail necessary for accurate recognition, so the filtering strength was optimized accordingly.

Finally, all images were resized to a uniform dimension of 36 × 36 pixels to ensure consistent input size for the PCA-based feature extraction stage. Resizing not only standardizes the spatial resolution of images from different sources but also reduces computational load and facilitates efficient batch processing. At the same time, this operation can lead to minor loss of fine structural information or introduce interpolation artifacts, but these effects were minimized through careful selection of scaling parameters. Together, these preprocessing steps established a balanced trade-off between data simplification and feature preservation. Each operation contributed to improving the overall quality and consistency of the datasets, providing a reliable foundation for effective feature extraction and object recognition in subsequent stages of the research.

Normalization ensures that all pixel intensities lie within a comparable numerical range, stabilizing the learning process and preventing features with larger values from dominating. In this study, each image was standardized by subtracting the dataset's mean intensity and dividing by its standard deviation [132]. This procedure ensures that the transformed data have zero mean and unit variance, improving convergence and maintaining consistency across both training and testing samples.

Normalization offers key benefits in machine learning. It stabilizes training by preventing features with large magnitudes from dominating, improves convergence during optimization, and enhances comparability across datasets. It also helps models learn more efficiently by ensuring consistent feature scaling. Normalization is especially important when features span different ranges or units. However, it requires using the same normalization parameters—such as mean and standard deviation—from the training set during testing or inference; otherwise, distributional shifts may occur, degrading performance and generalization.

Table 2.1 – Summary of Preprocessing Techniques Used in the Study

Technique	Purpose / Function	Advantages	Disadvantages

Grayscale Conversion	Converts RGB images to single-channel intensity images to simplify data.	Reduces computational cost; removes irrelevant color variations; enhances edge contrast.	Loses chromatic information useful for color-dependent recognition.
Histogram Equalization	Redistributes intensity values to improve image contrast and highlight features.	Enhances visibility in poorly lit images; compensates for illumination variation.	May amplify noise and distort brightness in uniform regions.
Cropping and Centering	Isolates the region of interest (face or hand) and removes background clutter.	Focuses on relevant objects; reduces background interference; speeds up computation.	Risk of cutting relevant regions or distorting object proportions.
Thresholding / Binarization	Converts grayscale image to binary (black–white) for clear shape boundaries.	Simplifies contour detection and segmentation; useful for gesture silhouettes.	Sensitive to light variation; can lose texture and fine detail.
Gaussian Blurring	Applies Gaussian filter to reduce high-frequency noise and smooth images.	Suppresses noise; stabilizes gradient features; improves generalization.	Excessive blur reduces edge sharpness and fine features.
Resizing	Standardizes all images to a uniform dimension (36 × 36 px).	Ensures consistent input size; reduces memory and computation.	Downscaling may remove structural detail; interpolation may distort pixels.
Normalization / Standardization	Centers and scales data so pixel intensities have zero mean and unit variance.	Stabilizes model training; balances feature magnitude; accelerates convergence.	Requires same parameters (μ, σ) for train/test to avoid mismatch.

Through these preprocessing operations - grayscale conversion, histogram equalization, cropping, thresholding, Gaussian smoothing, resizing, and normalization the datasets were transformed into standardized, noise-reduced, and contrast-enhanced forms suitable for subsequent PCA-based dimensionality reduction and Triplet Similarity learning. These operations collectively ensure that the hybrid recognition framework proposed in this study learns from consistent, high-quality data representations, thereby maximizing recognition accuracy.

2.3 Proposed Hybrid Method: PCA-TP Recognition Framework

2.3.1 PCA

PCA is a classical statistical and linear-algebraic method for transforming high-dimensional correlated data into a set of uncorrelated variables known as principal components. Originally proposed by Pearson in 1901 and later generalized by Hotelling in 1933 [133], PCA has become one of the most influential techniques in applied mathematics, data science, and pattern recognition [134]. Its main purpose is to reduce dimensionality while retaining the variance that contributes most to the data's structure. In machine learning and computer vision, PCA serves as a preprocessing step for pattern discovery, feature extraction, and signal denoising [135].

In the context of object recognition, PCA is widely applied to project image data onto a lower-dimensional subspace that captures the most informative patterns. This technique, also known as the Karhunen–Loève transform, is foundational to the well-known Eigenfaces approach, where facial images are represented as linear combinations of eigenvectors (principal components) [136].

Table 2.2 – The features of PCA are shown in the table below

Feature	PCA
Discrimination between classes	PCA manages the entire data for the principal components analysis without taking into consideration the fundamental class structure.
Applications	PCA applications in the significant fields of criminal investigation are beneficial
Computation for large datasets	PCA does not require large computations
Direction of maximum discrimination	The directions of the maximum discrimination are not the same as the directions of maximum variance as it is not required to utilize the class information such as the within class scatter and between class scatter
Focus	Identifies orthogonal axes capturing the widest variation.
Supervised learning technique	Unsupervised
Well distributed classes in small datasets	Effective for dimensionality reduction but less discriminative than supervised methods

A new image is recognized by projecting it into this eigenspace and comparing its coordinates with existing templates, usually via the Euclidean or cosine distance. PCA's strength lies in its ability to compress redundant information, enhance computational efficiency, and suppress noise while preserving discriminative variance.

Table 2.2 summarizes its essential analytical properties [137].

PCA is a typical approach for discovering patterns in high-dimensional data [138]. It has applications in sectors such as face recognition and picture reduction. The whole list of PCA benefits is as follows: Due to the reduced dimensions of the processes, there is less noise sensitivity, less needs for capacity and memory, and more efficiency due to the lack of duplicated data as a result of orthogonal components. The introduction of PCA makes it easier to organize photos. The database representation is reduced since just the trainee pictures in the form of their projections are maintained. Selecting a maximum variation basis lowers noise, so little fluctuations in the background are automatically disregarded [139].

PCA has two major disadvantages. The covariance matrix is difficult to investigate adequately [140]. Without specific training data, the PCA is incapable of capturing even the most basic invariance. PCA has been recreated a number of times. This interpretation emphasizes the modeling capabilities of PCA and is deeply anchored in the theory of regression: variation is explained by the principle components from Pearson’s perspective. Later, in the 1930s, the notion of linear combinations of variables was created, and Hotelling emphasized the significance of key component variation. This is a more complex statistical method. Eventually, the parallels between the two approaches were uncovered.

Principal Component Analysis (PCA) is a statistical method used to transform high-dimensional data into a more compact form while retaining most of its meaningful variance. The process begins with data centering, where the average value of each feature is subtracted to remove bias and align all features around a common origin [133, p. 4]. Next, the variability and relationships among features are analyzed to identify the directions in which the data vary the most. These directions, called principal components, are arranged according to the amount of variance they capture. The data are then projected onto a smaller set of these principal components, forming a reduced representation that preserves the dominant patterns while eliminating noise and redundancy.

In this research, PCA serves as the initial stage of the proposed PCA–Triplet Projection (PCA–TP) hybrid algorithm. It operates as a dimensionality-reduction and noise-suppression step, transforming raw pixel information from the ORL and HandReader datasets into concise and discriminative feature vectors. By keeping only the most informative components, PCA decreases computational complexity, reduces redundancy, and stabilizes the subsequent embedding process based on Triplet Similarity Loss. This ensures that only the most essential spatial and structural information is preserved, improving both convergence and generalization in limited-data scenarios. As a result, PCA provides a robust foundation for efficient and accurate object-recognition performance within the overall framework.

2.3.2 Triplet Loss

Triplet loss learning is a supervised metric learning technique that seeks to construct a discriminative embedding space in which intra-class distances are minimized and inter-class distances are maximized. It operates on groups of three samples—an anchor, a positive, and a negative—to explicitly encode relative similarity constraints in the learned feature representation. The approach enforces that the feature vector of the anchor sample be closer to that of a positive sample (belonging to the same class) than to a negative sample (from a different class) by a predefined margin. This comparative framework allows the model to capture semantic similarity relationships rather than relying solely on categorical labels [141].

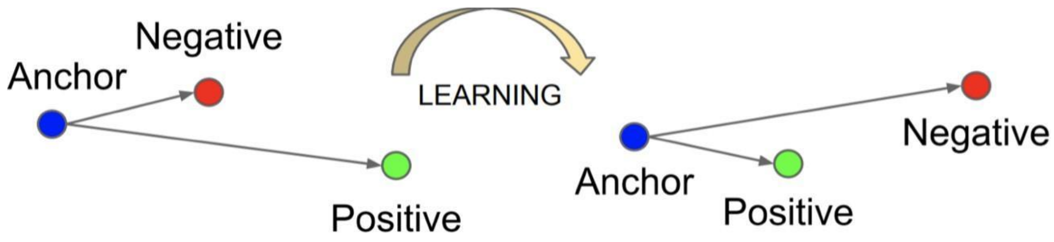

Figure 2.5 – The Triplet Loss minimizes the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the Anchor and a negative of a different identity

Originally introduced in the context of face verification and image retrieval, triplet loss quickly became one of the most widely adopted loss functions in deep representation learning. The seminal work FaceNet demonstrated that training with triplet loss enables the direct learning of a Euclidean embedding space in which simple distances correspond to meaningful visual similarities [142]. Since then, this principle has been extended to a wide range of applications, including person re-identification, signature verification, image retrieval, speaker identification, and hand gesture recognition. The triplet-based learning paradigm has therefore become a standard tool in both academic research and industrial computer vision systems [143].

Triplet loss is particularly effective in deep learning architectures, where it is typically applied to the final embedding layer of a CNN. In this setting, the network learns hierarchical representations that map input samples into a compact feature space optimized for similarity-based comparisons. Compared to classical classification losses such as softmax or cross-entropy, the triplet loss provides a fine-grained supervision mechanism that encourages continuous, structured embedding manifolds instead of discrete class boundaries [144]. As a result, it has proven especially powerful for tasks involving large intra-class variability and inter-class similarity, where small details play a critical role in distinguishing between visually similar samples.

Furthermore, triplet-based methods are known for their ability to generalize across unseen categories, making them valuable for few-shot learning and open-set recognition problems. By training on relative distances rather than explicit labels, the learned embeddings can be reused for novel classes without retraining, simply by computing pairwise distances in the embedding space. This property has made triplet loss one of the key components of modern deep metric learning frameworks, often combined with dimensionality reduction or projection techniques such as PCA, t-SNE, and autoencoders to further refine feature representations [145].

The triplet loss function operates on triplets of samples:

= Anchor (A): a reference image,

= Positive (P): an image of the same class as the anchor,

= Negative (N): an image from a different class.

The goal is to ensure that the distance between the anchor and positive samples is smaller than the distance between the anchor and negative samples by at least a constant margin □. This fundamental constraint is represented as in the equation (1) [146]:

$$||f(A) - f(P)||_2^2 + \alpha < ||f(A) - f(N)||_2^2 \quad (1)$$

Here, □(□) denotes the embedding function that maps the input image □ into a latent feature space, and $\| \cdot \|_2$ represents the Euclidean distance.

The corresponding triplet loss function can then be expressed as in the equation (2):

$$L_{tri} = max(0, ||f(A) - f(P)||_2^2 - ||f(A) - f(N)||_2^2 + \alpha \ ) \quad (2)$$

Where □ is the margin parameter that controls the separation between positive and negative pairs. The hinge-based formulation ensures that only those triplets that violate this margin constraint contribute to the loss, allowing the model to focus on "hard" examples that are more informative during training. The effectiveness of triplet loss learning largely depends on how triplets are selected. Since the number of possible triplets grows cubically with the dataset size, it is computationally impractical to use all combinations. Therefore, hard-negative mining and semi-hard sampling strategies are often adopted, where the negative samples are selected such that they are farther from the anchor than the positive samples but still within the defined margin. This approach accelerates convergence and enhances the discriminative capacity of the learned embeddings. The performance of the model also depends on the choice of the margin □. A small margin may not enforce sufficient separation between classes, while a large margin can lead to unstable optimization. The optimal margin is typically determined empirically through validation experiments. In the PCA–TP hybrid recognition system, the triplet loss plays a key role in achieving discriminative feature separation after PCA-based compression. While PCA performs unsupervised dimensionality reduction to remove redundancy and noise, triplet loss supervises the embedding learning process to ensure that semantically similar samples remain close in the feature space and dissimilar samples are adequately separated. This two-stage integration provides computational efficiency allowing the framework to perform effectively even with limited training data.

2.3.3 Modified Triplet Similarity Loss

The Modified Triplet Similarity Loss (mTSL) enhances the traditional triplet framework by replacing Euclidean distance with cosine similarity and introducing L_2-normalized embeddings to ensure that learning occurs within an angular, scale-invariant space. Each PCA-reduced feature vector is linearly transformed through a learnable projection, defined in equation (3):

$$f(x) = Wx, \quad (3)$$

where $W \in R^{r\times d}$ is a learnable projection matrix mapping a □-dimensional input vector (obtained from PCA) to an □-dimensional embedding space. Following the normalization process described by Zhang et al. in 2020, all embeddings are constrained to lie on a unit hypersphere, making optimization dependent solely on directional relationships between samples [147]. The triplet objective is then reformulated as in given equation (4):

$$L_{mTSL} = max(0, sim(a,n) - sim(a,p) + \alpha) \quad (4)$$

where □□□(□,□) and □□□(□,□) sim(a,n) denote cosine similarities between anchor–positive and anchor–negative pairs, respectively, and □ is the angular margin. This formulation follows the margin-based principle of FaceNet [146, p. 7] but aligns with the angular-margin strategies of CosFace by Wang et al., in 2018 which enhance inter-class separation and intra-class compactness [148]. To overcome the limitation of a fixed margin, an adaptive mechanism is employed in which the margin varies according to sample difficulty, following the approaches of Ha and Blanz in 2021, Tian et al. in 2019 and Nguyen et al in 2022 [149-151]. The final adaptive-margin loss is expressed as in given equation (5)

$$L_{AMTL} = max(0, sim(a,n) - sim(a,p) + \alpha_i) \quad (5)$$

which automatically assigns larger margins to hard triplets and smaller ones to easy cases, improving convergence stability and discrimination without explicit hard-sample mining. Overall, the proposed mTSL integrates the margin-based triplet structure of FaceNet, the hyperspherical normalization of Zhang et al. [146, p.4], and the adaptive-margin principles of recent triplet-ranking studies, yielding a robust, angularly consistent, and data-efficient learning objective suitable for small or imbalanced datasets.

2.3.4 Projection-Based Classifier

Projection-based classification is a widely used technique in pattern recognition, particularly in systems where dimensionality reduction plays a critical role in improving computational efficiency and recognition accuracy. In this approach, the original high-dimensional feature vectors are projected onto a lower-dimensional subspace, preserving the most discriminative information while reducing redundancy and noise. The lower-dimensional subspace serves as a compact representation that enables efficient comparison between feature embeddings through geometric similarity measures [152].

In the context of the PCA–TP hybrid method, projection-based classification represents the final stage of the recognition pipeline. After dimensionality reduction using PCA and discriminative embedding learning through the triplet loss function, the resulting feature vectors are projected onto a subspace defined by principal components. This projection not only reduces computational cost but also stabilizes the classification process in scenarios with limited training data.

Principal Component Analysis (PCA) can be applied as a projection-based metric to create compact and discriminative representations of data. The process begins with data preprocessing, where the dataset is standardized so that each feature has zero mean and unit variance. This step ensures uniform scaling and removes bias caused by differing feature magnitudes. After standardization, the relationships among the features are analyzed through the computation of a covariance matrix, which captures how variables vary together across the dataset. The main directions of variance are then identified through the decomposition of this covariance structure, revealing a set of components that explain most of the data variability. These components are ranked according to their importance, and only the most informative ones are retained to form the projection basis.Using this reduced basis, the data are transformed into a lower-dimensional subspace that preserves the most significant patterns and variations. This transformation provides a compact and noise-resistant representation suitable for subsequent metric learning, particularly when combined with the modified triplet loss function. The projection-based classification strategy introduced here offers distinct advantages for small or noisy datasets. By operating in a reduced-dimensional space, the model becomes less sensitive to irrelevant variations and random noise, leading to more stable and generalized results. Furthermore, because the principal components are orthogonal, the resulting features are uncorrelated, which simplifies distance computation and enhances interpretability in the recognition process [153].

In the PCA–TP method, projection-based classification is employed to map the discriminative embeddings, obtained through triplet similarity learning, into a compact, noise-reduced subspace. Recognition is then performed by computing the minimum projected distance between the test embedding and each class subspace. This mechanism enables the system to achieve stable recognition even when only a small number of training samples are available per class. By combining the statistical compression of PCA with the discriminative power of triplet learning, the projection-based stage transforms the learned embeddings into a geometry-aware representation space, where classes are more separable and classification becomes more stable to noise, scale, and illumination changes.

2.3.5 Integration of the PCA–TP Hybrid Framework

The proposed method aims to achieve efficient object recognition under limited dataset conditions by integrating PCA, a Modified Triplet Similarity Loss, and a Projection-based classification stage into a unified lightweight framework. This approach, referred to as PCA–Triplet–Projection (PCA–TP), is designed to maintain high accuracy and generalization performance while significantly reducing computational complexity.

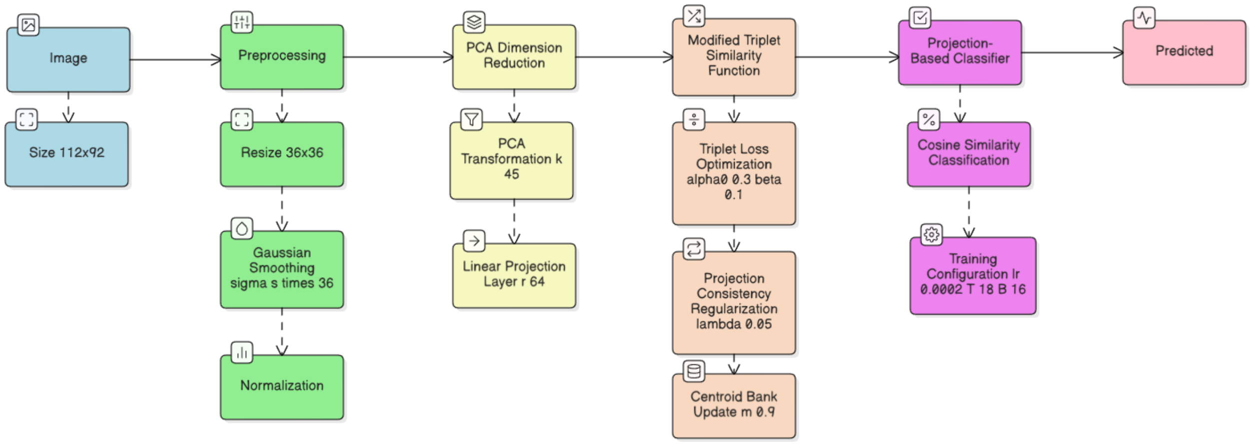

Figure 2.6 – General architecture of the proposed PCA–TP recognition method and algorithm

In this architecture, PCA serves as the initial dimensionality reduction stage, transforming image data into a compact representation that captures the most informative variance. This step reduces redundancy and prevents overfitting, which is particularly important for small datasets. The reduced features are then fed into a linear embedding layer trained using a Modified Triplet Similarity Loss, which replaces traditional Euclidean distance with cosine similarity and operates on L2-normalized embeddings. This modification constrains all features to lie on a unit hypersphere, ensuring that learning depends on angular similarity rather than feature magnitude, thus improving convergence stability with limited training data. The final Projection-based classification stage projects the learned embeddings onto class-specific subspaces or prototypes and assigns labels based on the maximum cosine similarity (or equivalently, minimum projection error). This projection mechanism provides stable classification performance by emphasizing the geometric alignment between samples and class feature directions. Overall, the PCA–TP framework combines the statistical efficiency of PCA, the discriminative power of cosine-based metric learning, and the stability of projection-based decision-making. Its lightweight linear structure and normalization-based optimization make it particularly well-suited for resource-constrained environments and recognition tasks where only small datasets are available, such as face and hand gesture recognition.

The proposed PCA–TP method is designed to achieve accurate object recognition in limited-data environments by integrating three computational stages—PCA, a Linear Embedding model trained with Modified Triplet Similarity Loss, and a Projection-based Classification stage. The architecture emphasizes simplicity, interpretability, and computational efficiency while maintaining high discriminative performance.

Input and Preprocessing: The system receives images of faces and hand gestures, which are preprocessed through resizing, normalization, grayscale conversion, and Gaussian smoothing to reduce noise and ensure consistent image dimensions. This preprocessing improves feature consistency and supports effective dimensionality reduction.

PCA Feature Reduction: The preprocessed images are vectorized and transformed using PCA. PCA extracts the most informative components from the data, removing redundancy and compressing the representation into a lower-dimensional feature space. This stage minimizes overfitting and computational complexity, enabling efficient learning from small datasets.

Linear Embedding with Modified Triplet Similarity Loss (mTSL): The PCA-reduced features are then passed to a linear embedding layer, which learns to project input samples into a discriminative embedding space. Unlike conventional triplet loss that uses Euclidean distance, the Modified Triplet Similarity Loss operates on cosine similarity between L2-normalized embeddings. This modification constrains features to lie on a unit hypersphere, focusing optimization on angular relationships rather than magnitude. During training, triplets of samples (anchor, positive, negative) are constructed such that embeddings of similar samples are drawn closer while dissimilar ones are pushed apart by a defined margin □. This adaptation improves learning stability and generalization under data scarcity.

Projection-Based Classification: After training, the learned embeddings are used to construct class subspaces or prototypes that represent each category in the embedding space. For a new test sample, the embedding is projected onto each class subspace, and classification is determined by the maximum cosine similarity or minimum projection error. This geometric decision mechanism enhances recognition performance by aligning the test embedding with the most representative class direction. The final output of the system is the predicted class label, corresponding to the recognized face or hand gesture. The overall architecture of the proposed PCA–TP method is illustrated schematically in Figure 2.6. The diagram represents the complete data flow through the system. The integration of PCA, cosine-based triplet embedding, and projection classification creates a unified recognition framework that is both lightweight and effective. The method achieves stability and efficiency without the need for deep architectures or extensive training data, making it particularly suitable for applications such as face and hand gesture recognition in resource-constrained environments.

The process begins with input images from the ORL and HandReader datasets, each of which undergoes the preprocessing pipeline described in Section 2.2.

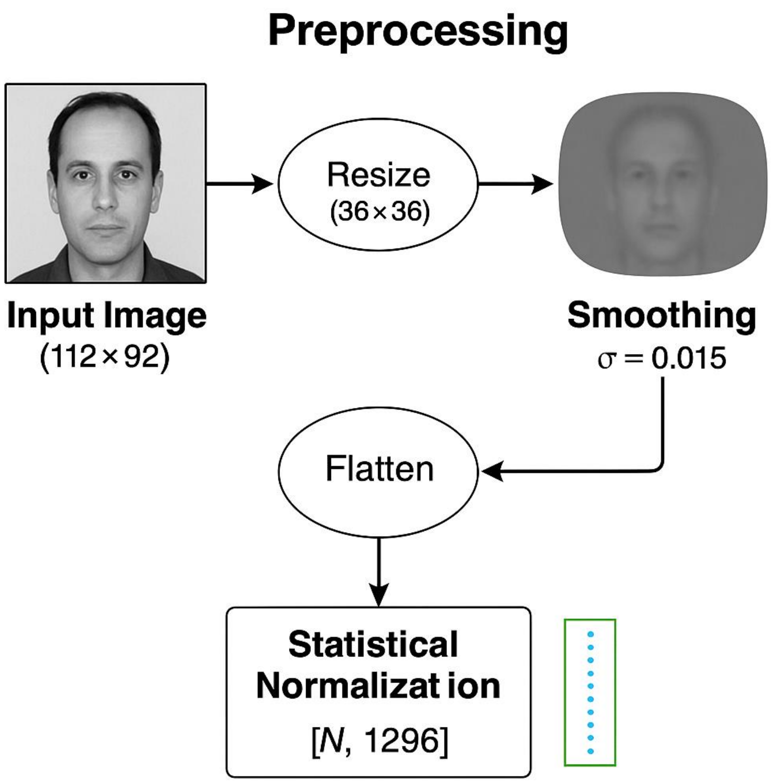

Figure 2.7 – Preprocessing step

After the preprocessing stage, each image is converted into a single continuous line of numerical values that represent its visual features. For example, a 36×36 pixel image becomes a vector containing 1,296 values, each corresponding to the brightness level of one pixel.

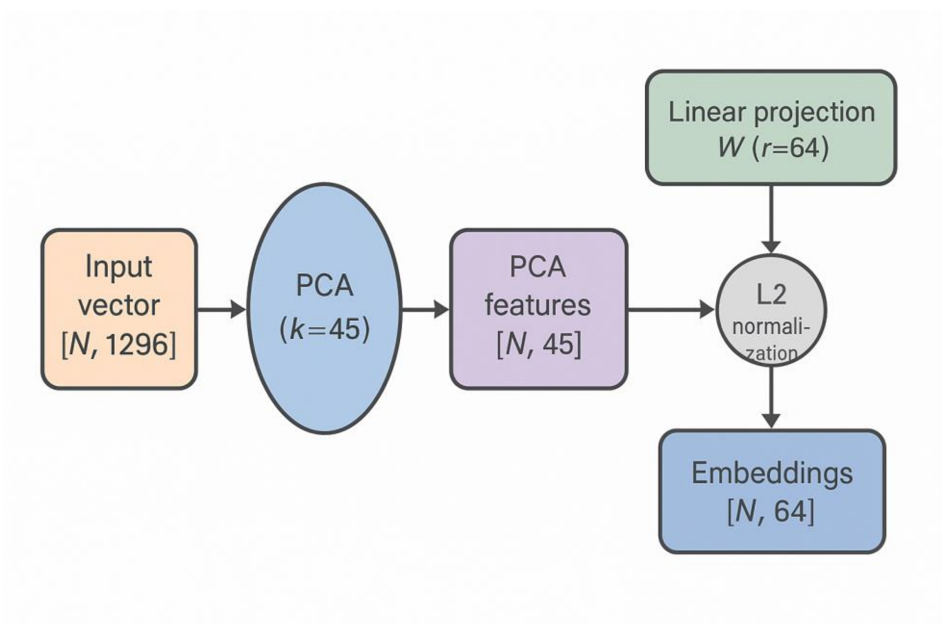

Figure 2.8 – PCA Dimensionality Reduction

When all images are processed in this way, they are arranged together into a single structured dataset, a feature matrix where each row represents one image, and each column represents a specific feature or pixel value.

```python
def apply_pca(X, n_components):  1 usage
    pca = PCA(n_components=n_components)
    X_pca = pca.fit_transform(X)
    return X_pca, pca
```

Figure 2.9 – PCA Dimensionality Reduction code

PCA is applied to the dataset to identify and retain only the most important patterns in the image data. This process reduces the large number of original pixel-based features to a smaller set of key variables that capture the majority of the variation in the images.

By doing so, PCA compresses the data into a lower-dimensional space, removing noise and redundant information while preserving the most discriminative characteristics. Each image is thus represented by a compact feature vector with 45 parameters.

This dimensionality reduction makes the following learning stages more efficient and stable, allowing the model to focus on the most informative features even when working with limited training data.

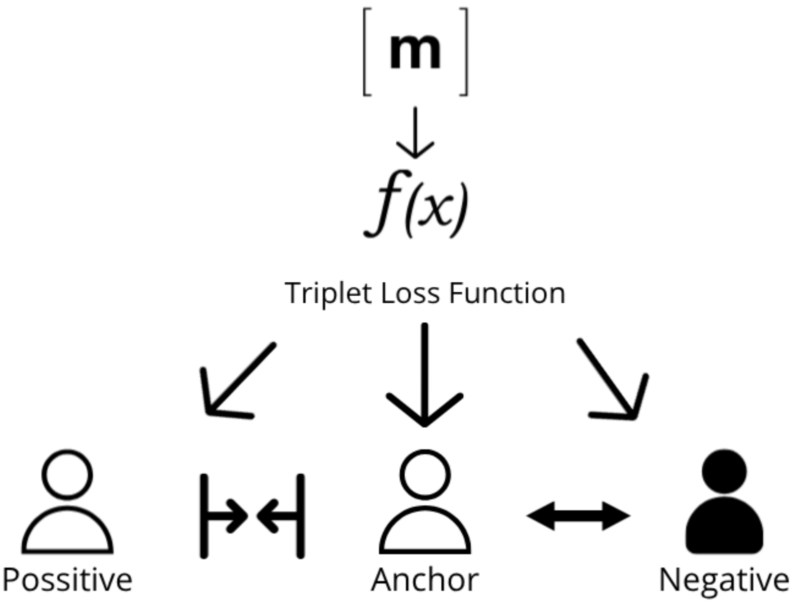

Figure 2.10 – Triplet Loss Function

The PCA-reduced features are passed to the Triplet Similarity Network, which learns a discriminative embedding space through a sequence of adaptive and projection-based operations. The process proceeds as follows:

The compact PCA features are fed into a learnable linear embedding layer. This layer refines the statistical features produced by PCA into a more discriminative form, creating a projection space suitable for similarity learning.

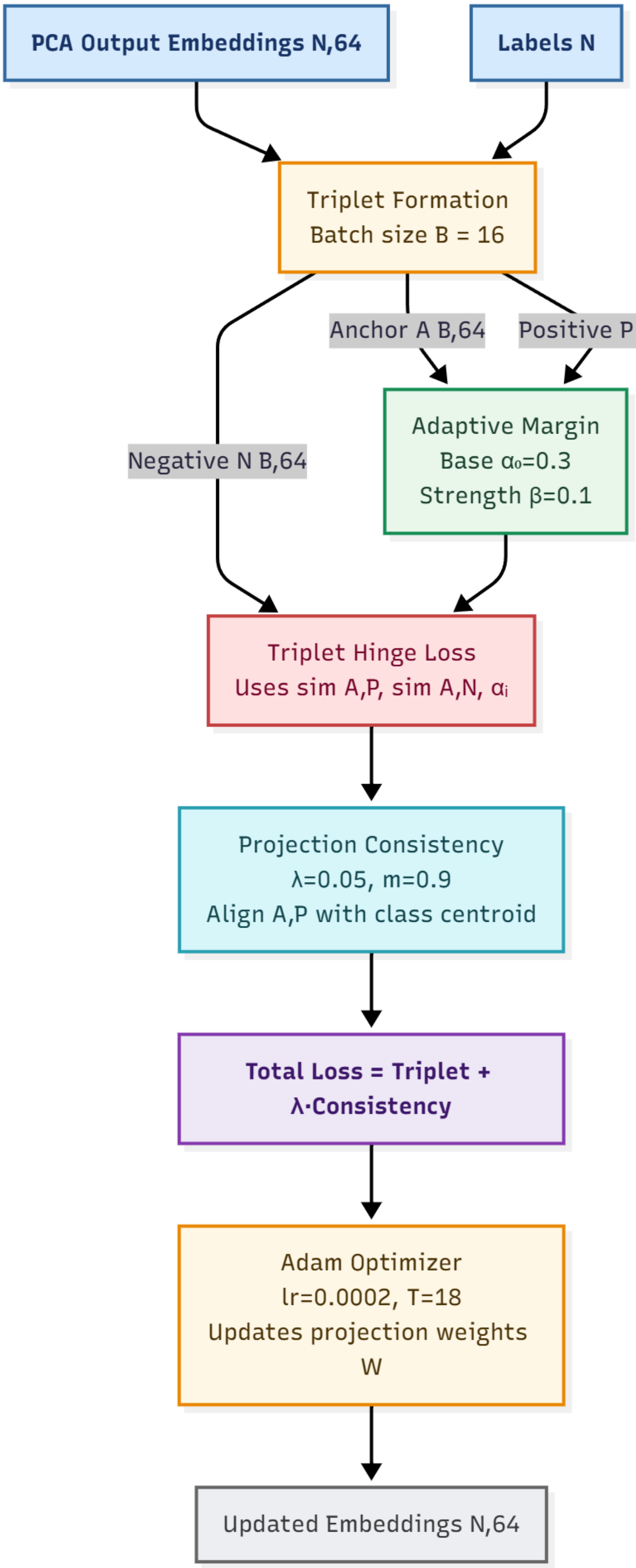

Figure 2.11 – Triplet sampling anchor, positive and negative by labels

The resulting embeddings are normalized to unit length so that similarity is evaluated based solely on angular distance. This normalization enhances training stability and ensures the model learns direction-based relationships between samples.

The training data are organized into triplets composed of an anchor sample, a positive sample from the same class, and a negative sample from a different class. These triplets form the basis of the learning process, allowing the model to distinguish between similar and dissimilar examples. Using a cosine-based triplet similarity loss, the network increases the angular similarity between the anchor and positive embeddings while decreasing the similarity between the anchor and negative embeddings. This encourages the formation of compact intra-class clusters and wider inter-class separation in the embedding space. To handle varying sample difficulty, an adaptive margin strategy is employed. The separation margin between positive and negative pairs is dynamically adjusted according to their similarity level—larger for difficult triplets and smaller for easier ones. This mechanism allows the model to focus learning on more challenging relationships while avoiding overfitting to trivial cases. During training, each embedding is softly aligned with its corresponding class centroid through a projection-consistency constraint. This regularization encourages the anchor and positive embeddings of the same class to share a common directional representation, improving intra-class coherence and geometric stability of the learned space. The parameters of the embedding layer are optimized iteratively to minimize the combined objective, which includes both the adaptive cosine triplet loss and the projection-consistency regularization term. The result is a stable and discriminative embedding space where angular distances reliably reflect semantic similarity. Once the discriminative embeddings are obtained, the final recognition stage employs a projection-based classification mechanism, illustrated in Figure 2.7.

In this stage, both training and testing embeddings are first normalized using L2 normalization, which projects them onto a unit hypersphere. This normalization ensures that classification depends solely on the directional similarity (cosine similarity) between vectors rather than their magnitude. The normalized test embeddings are then compared with all training embeddings using the cosine similarity function, which computes the angular distance between vectors. The test sample is assigned to the class whose training embedding yields the highest cosine similarity, representing the closest match in the learned embedding space. This projection-based classification approach eliminates the need for a separate classifier and instead relies on the intrinsic geometry of the embedding space learned through the Triplet Similarity Network.

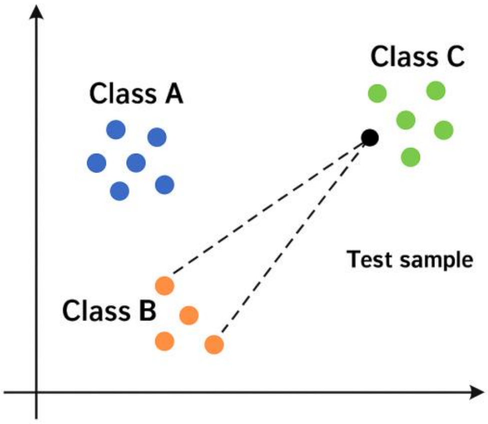

Figure 2.12 - Projection - based classification

By combining PCA-based dimensionality reduction, Triplet Similarity embedding, and projection-based classification, the proposed PCA–TP framework achieves a compact, interpretable, and data-efficient recognition pipeline. This integration ensures that the system remains under limited training data conditions, maintaining stability and high recognition accuracy while minimizing computational complexity.

Figure 2.16 illustrates the geometric interpretation of the projection-based classification process used in the PCA–TP framework. The two-dimensional embedding space (x–y plane) represents the reduced feature space obtained after PCA and Triplet Similarity learning. Each cluster (A, B, C) corresponds to the learned embeddings of samples belonging to distinct classes, and the black point denotes the test sample to be classified. Unlike traditional Euclidean classification, which measures straight-line distances between points, the projection-based approach relies on cosine similarity, which measures the angle between embedding vectors.

In this figure, although the test sample is physically closer to Class C in Euclidean terms, the direction of its vector is more aligned with the centroid vector of Class B, resulting in a smaller angular distance (higher cosine similarity). Therefore, the test sample is correctly projected onto the subspace of Class B and classified accordingly. This illustrates the key advantage of cosine-based projection which emphasizes directional consistency rather than magnitude, making the classifier better able to scale variations and noise in feature intensity. As a result, this projection-based classification ensures stable recognition even under limited data and imperfect feature scaling conditions. The proposed formulation introduces both methodological and algorithmic advancements that together enhance discriminative learning, geometric consistency, and computational efficiency.

Table 2.3 - Summary of Processes and Hyperparameters in the Proposed PCA–TP Algorithm

Process / Sub-Step	Key Hyperparameters	Remarks
PCA Transformation	*k = 45* (number of PCA components)	Non-trainable linear projection; lightweight preprocessing for feature compression.
Linear Projection Layer	*r = 64* (embedding dimension)	Provides low-parameter transformation before similarity learning.
Triplet Loss Optimization	$\alpha_0 = 0.3$ (base margin), $\beta = 0.1$ (adaptive strength)	Employs adaptive angular margin for handling variable sample difficulty.
Projection Consistency Regularization	$\lambda = 0.05$ (projection consistency weight)	Strengthens intra-class compactness and inter-class separation.
Centroid Bank Update	*m = 0.9* (momentum coefficient)	Stabilizes prototype directions; smooths learning dynamics.
Cosine Similarity Classification	—	Ensures consistency between training and inference metrics; low-complexity classifier.
Training Configuration	*lr = 0.0002*, *T = 18*, *B = 16*	Adam optimizer; small batch size enables balanced triplet sampling.

Table 2.3 provides a concise summary of the major processes, key hyperparameters, and their specific roles in the proposed PCA–TP framework. The model is structured as a sequential pipeline consisting of six principal stages—each responsible for transforming input data, enhancing feature discriminability, and maintaining recognition stability under limited data conditions. The framework is optimized for lightweight computation, minimal parameterization, and strong generalization capability.

The process begins with PCA Transformation, where the dimensionality of input feature vectors is reduced to k = 45 components. This non-trainable linear projection serves as a preprocessing step that compresses redundant information while preserving the principal variance of the data. The use of PCA ensures computational efficiency and stability, providing a compact feature representation that reduces overfitting and accelerates subsequent training stages. By reducing noise and emphasizing dominant eigen-directions, PCA also contributes to more stable convergence and improved feature interpretability, making it suitable for small-scale datasets where each sample carries significant informational value.

Following this, a Linear Projection Layer with an embedding dimension of r = 64 refines the PCA-compressed features into a low-dimensional latent space. This layer introduces a lightweight parametric mapping that enhances feature separability and prepares the embeddings for metric-based optimization without adding significant computational complexity. The layer's linearity preserves the geometric relationships learned during PCA while enabling fine-tuning through gradient-based optimization, thereby bridging classical feature extraction with modern learning-based adaptation.

The next stage involves Triplet Loss Optimization, a critical component of the framework. The loss is parameterized by a base margin of α_0 = 0.3 and an adaptive strength factor β = 0.1, enabling the model to handle variable sample difficulty during training. By comparing anchor, positive, and negative samples, the triplet mechanism enforces that features from the same class remain close together while pushing apart those from different classes. The inclusion of an adaptive angular margin further strengthens class boundaries and promotes more robust metric learning under data-scarce conditions. This process ensures discriminative embedding generation, even when the number of available examples per class is small or unevenly distributed.

To reinforce embedding stability, Projection Consistency Regularization is applied with a weighting factor of λ = 0.05. This constraint encourages consistent projection behavior across samples of the same class, enhancing intra-class compactness and inter-class discrimination. By maintaining projection consistency, the framework achieves smoother feature distribution and improved generalization during inference. This mechanism effectively aligns feature vectors toward shared subspaces, mitigating variance introduced by illumination, pose, or scale differences that often degrade recognition accuracy.

The Centroid Bank Update mechanism further stabilizes learning by maintaining running class prototypes with a momentum coefficient of m = 0.9. This process averages class representations over iterations, preventing abrupt fluctuations in centroid direction and ensuring smoother optimization. It effectively integrates both historical and current embedding information, providing a stable reference for ongoing triplet and projection learning. The centroid bank functions as a dynamic memory that captures the evolving structure of the embedding space, allowing the system to continuously refine its internal representations without external supervision.

Finally, the Cosine Similarity Classification stage performs the recognition task using cosine distance as the decision metric. This approach ensures alignment between the training objective and the inference metric, maintaining consistency across the entire learning process. The classifier operates with minimal parameters, contributing to the overall lightweight nature of the system. The cosine-based formulation provides numerical stability and scale invariance, which are advantageous when feature magnitudes differ due to normalization or limited sample diversity. The entire framework is trained using the Adam optimizer with a learning rate of lr = 0.0002, over T = 18 training epochs, and a mini-batch size of B = 16. This configuration supports balanced triplet sampling and stable convergence while maintaining low memory consumption. The combination of small batch processing and gradual parameter updates enhances generalization in limited-data environments.

Table 2.4 summarizes the hierarchical innovations introduced in the proposed PCA–TP (Principal Component Analysis – Triplet Projection) framework and highlights their corresponding contributions. The table presents each methodological and algorithmic advancement from foundational design (method level) to system-level integration, emphasizing how these improvements collectively enhance discriminative learning performance. At the method level, the model integrates PCA-based dimensionality reduction with a triplet similarity learning scheme. This integration enables the extraction of compact, informative features while maintaining discriminative capability in a low-dimensional linear space. By linking classical feature compression with modern metric-learning principles, the PCA–TP method delivers high accuracy without the computational cost or opacity of deep networks, demonstrating that linear representations can still achieve robust separability when properly optimized.

At the algorithmic level, several key modifications strengthen the metric-learning process.

First, the cosine-based modified triplet loss replaces conventional Euclidean distance with an angular similarity metric, making the learning process scale-invariant and more stable under limited data.

Second, the adaptive margin mechanism dynamically adjusts the margin between positive and negative pairs depending on their similarity, allowing the system to focus training on harder examples and reduce overfitting to trivial ones.

Third, the projection-consistency regularization term encourages embeddings to align with their class centroids, improving intra-class compactness and ensuring consistent geometric relationships in the feature space. Together, these elements enhance both discriminative power and stability, especially in small-sample scenarios.

Table 2.4 - Comparison Analysis of the Level

Level	Component	Description of Innovation	Contribution / Impact
1	2	3	4
Method	Integration of PCA and Triplet Similarity Learning (PCA–TP)	PCA-reduced features are directly used as input to a triplet similarity framework, enabling discriminative learning in a compact linear subspace.	Connects classical feature reduction with modern metric learning; achieves strong discrimination without deep networks.
Algorithm	Cosine-Based Modified Triplet Loss Function	Reformulates the traditional Euclidean triplet loss into a cosine-based version, emphasizing angular similarity and scale invariance.	Enhances angular discrimination, improves maintains consistent behavior in low-data environments.
Algorithm	Adaptive Margin Mechanism	Dynamically adjusts the separation margin between similar and dissimilar samples according to their similarity level.	Focuses optimization on difficult samples, stabilizes convergence, and avoids overfitting to easy pairs.
Algorithm	Projection-Consistency Regularization	Aligns embeddings with their class centroid directions to maintain geometric consistency and intra-class cohesion.	Strengthens feature compactness, stabilizes angular geometry, and improves inter-class separation.
Overall	Unified Lightweight Metric-Learning Framework	Combines PCA-based representation and the modified triplet loss function within a linear, interpretable system.	Achieves high recognition accuracy with minimal complexity and strong generalization for small datasets.

Finally, at the overall framework level, these components are unified into a lightweight, interpretable metric-learning system. The resulting PCA–TP framework achieves strong angular discrimination, stable convergence, and high recognition accuracy while maintaining minimal parameterization. This design successfully bridges the gap between traditional linear feature extraction and advanced metric-learning strategies, offering an efficient alternative to complex deep-learning models for object recognition tasks under data-constrained conditions.

3. EXPERIMENT AND RESULT

3.1 Hand gesture recognition algorithms and picture representation methods

The objective of this experiment is to evaluate the performance of several classical machine learning algorithms for static hand gesture recognition using different image representation methods. Specifically, five algorithms—KNN, RF, LDA, DT, and Gaussian Naïve Bayes (GNB)—are tested to identify which provides the most reliable classification accuracy for limited, small-scale datasets.

The experiment utilizes the HandReader dataset in Section 2.1.2 as the primary data source. To reduce computational complexity and ensure controlled evaluation, only four gesture categories (A, B, C, and D) were selected. Each class includes 50 samples, resulting in a total of 200 images.

Experiments were implemented in Python using scikit-learn on the same dataset splits and preprocessing steps. Each algorithm was trained and tested using all three representation methods. The recognition accuracy was calculated for every image size configuration.

3.1.1 Preprocessing

In this experiment, three image preprocessing techniques were applied to enhance recognition accuracy and ensure consistent feature extraction across samples: raw pixel representation, histogram-based features, and grayscale transformation. All images were cropped and resized to fixed dimensions to maintain uniform input size, which helps improve the reliability of classification results. Since the HandReader dataset contains images with a uniform black background, no background subtraction or noise removal was required.

The raw pixel method represents each image as a matrix of pixel intensity values, where each pixel contains red, green, and blue (RGB) components. This matrix is flattened into a one-dimensional feature vector and stored as a column in the data matrix, forming the direct input for the learning algorithms.

The histogram method captures color distribution information by converting each image from the RGB color space to HSV (Hue, Saturation, Value) format using the OpenCV library. A histogram is then computed to represent the frequency of pixel intensities, producing a feature vector that emphasizes color variation rather than spatial structure.

The grayscale method converts images into a single-channel intensity format, simplifying the data while preserving essential shape and texture features. Similar to the raw pixel approach, the grayscale matrix is flattened into feature vectors, resulting in reduced dimensionality and faster computation.

Each preprocessing method provides a distinct representation of visual information, allowing for comparative evaluation of how data encoding influences the classification performance of traditional machine learning algorithms.

3.1.2 Experiment result

This section analyzes and assesses the performance of five algorithms, namely KNN, RF, LDA, DT and GNB employing Raw pixel, histogram, and grayscale methods. The Python platform was used for experiments. Before simulation, every image in the dataset was cropped to extract hand motions and rescaled into six groups of varying sizes: 20x15, 24x18, 28x21, 32x24, 36x27, and 40x30 as given in table 3.1.

Table 3.1 – Result of algorithms using Raw Pixel method

Raw Pixels Size	KNN Accuracy (%)	RF Accuracy (%)	LDA Accuracy (%)	DT Accuracy (%)	GNB Accuracy (%)
20x15	95	97	90	83	84
24x18	97	97	92	80	82
28x21	96	97	95	86	89
32x24	97	96	90	78	85
36x27	97	94	92	84	87
40x30	96	96	95	78	85

Table 3.2 – Result of algorithms using Histogram and Grayscaled methods

Histogram Size	KNN Accuracy (%)	RF Accuracy (%)	LDA Accuracy (%)	DT Accuracy (%)	GNB Accuracy (%)
20x15	62	59	56	59	54
24x18	62	59	56	59	54
28x21	62	59	56	59	54
32x24	62	59	56	59	54
36x27	62	59	56	59	54
40x30	62	59	56	59	54
Grayscaled Size	KNN Accuracy (%)	RF Accuracy (%)	LDA Accuracy (%)	DT Accuracy (%)	GNB Accuracy (%)
20x15	95	95	90	82	84
24x18	97	94	92	84	82
28x21	96	94	95	89	89
32x24	97	96	90	78	85
36x27	97	95	92	80	87
40x30	96	97	95	76	85

As shown in table 3.2 KNN and RF have the highest accuracy of 97%. Additionally, LDA achieves 95% accuracy at 28x21 and 40x30 sizes. In addition, it is necessary to note that algorithm performance is less affected by different sizes. Another experiment demonstrates that the histogram method is not recommended due to its low rate of accuracy. The maximum result exhibited by the KNN algorithm was only 62%. This experiment also demonstrates that the size of the dataset has no effect on the performance of the histogram approach.

The final grayscale approach produced results comparable to raw pixel methods. 97% is the maximum recognition rate for KNN and RF. The maximum performance of the DT and GNB algorithms at 28x21 is 89%, which is also an excellent result. In addition, we calculate the average accuracy of all algorithm sizes in this experiment as shown in figure 3.1 in terms of average 90% accuracy, KNN, RF, and LDA improved by 96.33, 96.15, and 92.33%, respectively. GNB's performance is 85.33 percent, and DT's is the lowest at 81.5%.

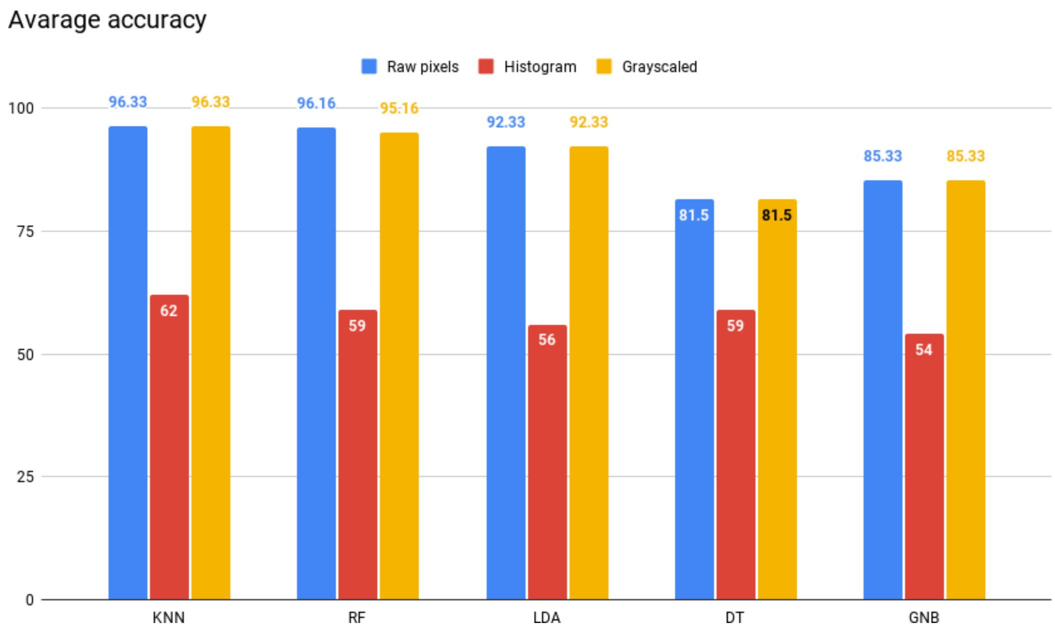

Figure 3.1 – Average accuracy of algorithms

This experiment demonstrates that KNN and RF perform better than other algorithms. Also, the histogram approach is not suggested for experiments of this nature. Two additional methods function as intended. Looking at the average outcomes of algorithms, the difference between raw pixel and grayscale algorithms is only 1%. Thus, the Raw pixels approach produces the best results. On the other hand, a matrix generated with the raw pixel approach has three times as many elements as a matrix constructed with the grayscale method. It is anticipated that the grayscale approach will produce faster results. Also recommended is the calculation of the speed of algorithms and approaches.

3.2 Recognition of gestures with the use of sparse reconstruction

This experiment aims to evaluate whether a sparsity-based classification approach can enhance recognition accuracy and computational efficiency in static hand gesture recognition tasks. Specifically, a Modified Sparse Representation Classifier (MSRC), which relies on image intensity features, was compared with two traditional machine learning algorithms KNN and RF to determine whether sparse reconstruction could achieve faster and more accurate gesture recognition. The dataset used for this study was derived from the open-source Hand Reader Dataset described in Section 2.1.2. For the present experiment, only four gestures (A, B, C, and D) were selected, totaling 200 images. Each image was cropped to isolate the hand region and converted to grayscale to reduce computational complexity before being resized into six groups: 20×15, 24×18, 28×21, 32×24, 36×27, and 40×30 pixels. The dataset was divided into

120 training and 80 testing samples for consistent evaluation. All experiments were implemented in Python, and two performance metrics were examined: recognition accuracy and execution time. Three classifiers—KNN, RF, and MSRC—were trained on identical training data and evaluated on the same test set. KNN served as the baseline for distance-based classification, RF represented an ensemble learning approach, and MSRC reconstructed each test image as a sparse linear combination of training samples, enabling classification based on the minimum reconstruction error for each gesture class.

3.2.1 Experiment Result and Discussion

The experimental results summarized in Table 3.3 demonstrate the comparative performance of the three classification algorithms—KNN, RF, and MSRC—in terms of both recognition rate (RR) and execution time across six image resolutions. The results clearly indicate that KNN provides a stable and reliable performance, maintaining a consistent recognition accuracy of 95% across all image sizes, with an average execution time of approximately 0.1 seconds. This confirms KNN's efficiency for small-scale datasets, where its simplicity and low computational overhead yield results.

Table 3.3 – Recognition rate (RR) and execution time for KNN, RF and MSRC

Algorithms	KNN		RF		MSRC	
SIZE	RR(%)	TIME(s)	RR(%)	TIME(s)	RR(%)	TIME(s)
20x15	95	0.104	87.5	0.216	97.5	0.054
24x18	95	0.104	92.5	0.222	95.1	0.056
28x21	95	0.104	88.75	0.228	96.25	0.056
32x24	95	0.105	88.75	0.233	96.25	0.061
36x27	95	0.105	87.5	0.236	95	0.061
40x30	95	0.107	91.25	0.281	97.5	0.081

The RF classifier, on the other hand, exhibits more variable behavior. Its recognition rate fluctuates between 87.5% and 92.5%, and its execution time ranges from 0.216 to 0.281 seconds, indicating that it is computationally more expensive. The additional time complexity arises from the ensemble nature of RF, which relies on the training and evaluation of multiple DT. While RF can handle nonlinear data distributions more effectively than KNN, it requires significantly more processing time, making it less practical for applications demanding real-time performance.

In contrast, the MSRC method achieves the highest recognition accuracy among all three algorithms, reaching up to 97.5% while maintaining the lowest average execution time—less than 0.1 seconds even for the largest image size. This superior performance demonstrates the effectiveness of sparse reconstruction in capturing discriminative features from grayscale gesture images. By representing each test image as a sparse linear combination of training samples, MSRC focuses on the most informative components of the input, thereby improving recognition precision and computational speed.

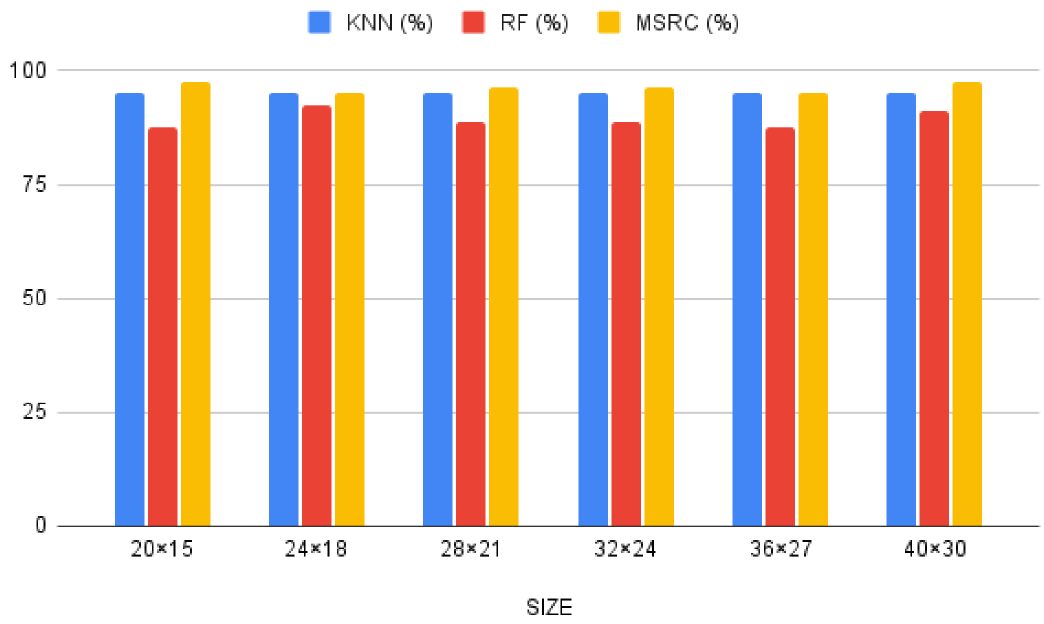

Figure 3.2 –Recognition rate vs image size for all algorithms of KNN, RF and MSRC

Although MSRC provides the best overall results, it also presents a notable limitation: as the size of the training matrix increases, the computational burden associated with solving the sparse optimization problem grows rapidly. Therefore, while MSRC performs exceptionally well on small- to medium-scale datasets, its scalability to very large datasets remains limited. Nonetheless, the results, illustrated in Figures 3.2 and 3.3, confirm that MSRC outperforms both KNN and RF in terms of recognition rate and efficiency, making it a strong candidate for hand gesture recognition applications where high accuracy and low latency are required.

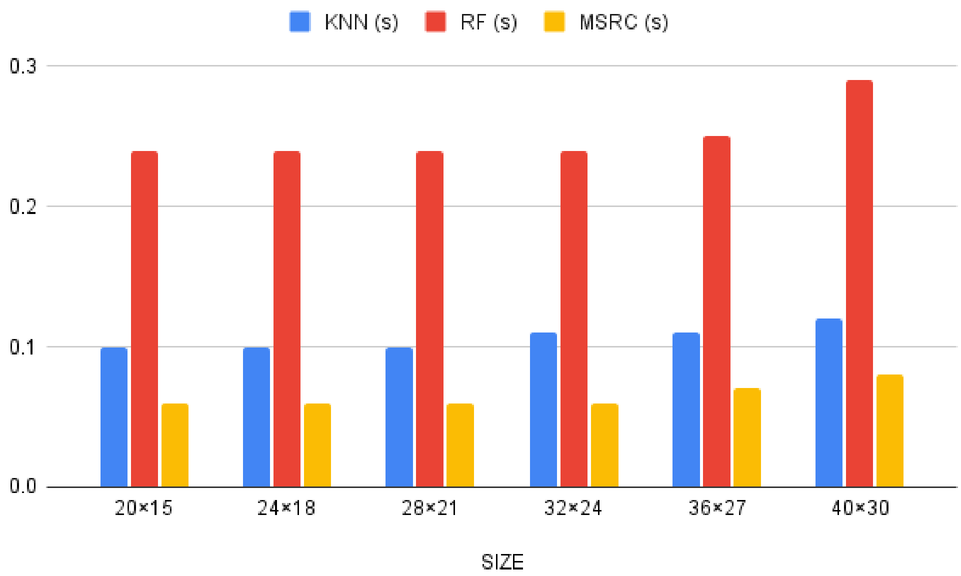

Figure 3.3 – Execution time vs image size for all algorithms of KNN, RF and MSRC

Recognition rate (RR) and execution time for KNN, RF and MSRC algorithms for different image sizes, (ranging between 87.50 and 92.50), (between 0.216 and 0.281). In spite of the fact that the amount of time required to complete a task grows proportionately with the size of the picture, as shown in Figure 3.3 and Figure 3.4, MSRC offers the best performance, with recognition rates higher than 95% and timeframes of less than 0.1 second for all sizes. Due to the astronomically high size of the training matrix, MSRC cannot be used for big datasets; as a result, it is best suited for pictures of a modest size. As a limitation of MSRC, it should be mentioned that it is best suited for small- sized images. All of the data gathered from simulation are depicted in figures 3.2, 3.3.

3.3 PCA and projection based hand gesture recognition

The primary objective of this experiment is to evaluate the effectiveness of a hybrid recognition framework that integrates PCA, Triplet Similarity learning, and Projection-based classification for static hand gesture recognition. Specifically, the experiment aims to determine whether combining dimensionality reduction, discriminative embedding learning, and projection-based decision-making can enhance recognition accuracy and computational efficiency compared to traditional approaches such as KNN and MSRC.

Additionally, the experiment investigates the impact of preprocessing techniques—including grayscale conversion, thresholding, and contour-based cropping—on the performance of the recognition pipeline. The goal is to validate whether PCA coupled with Triplet Similarity and Projection can form a more stable and discriminative representation for gesture classification using the HandReader dataset.

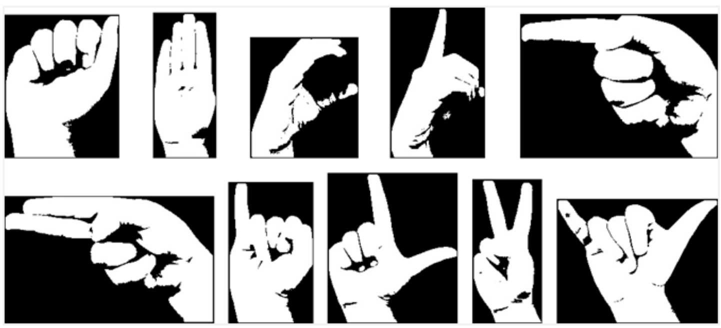

Figure 3.4 – Examples from preprocessed dataset

Every utilizing threshold of 60 in the spectrum [0, 255], the picture is transformed to grayscale and subsequently to black and white, having white symbolizing a human skeleton. The hand is then trimmed to produce a rectangle shaped picture, which is subsequently contoured to create a 30x30 image. Figure 3.6 depicts the image prior to being resized to 30x30.

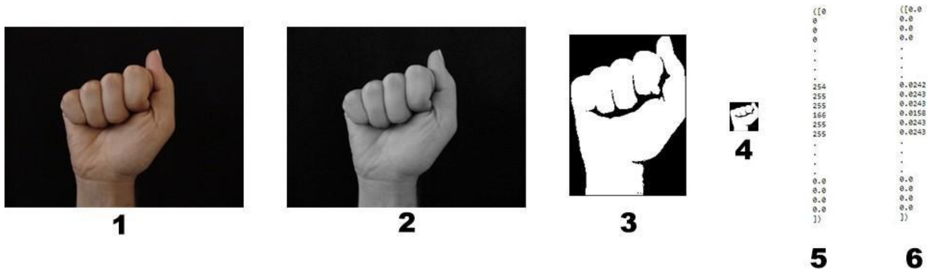

Figure 3.5 – Dataset preprocessing steps

3.3.1 Experiment Result

In this part, we evaluate the effectiveness of KNN, MSRC, and the suggested PCA- TP using the findings of our experiments

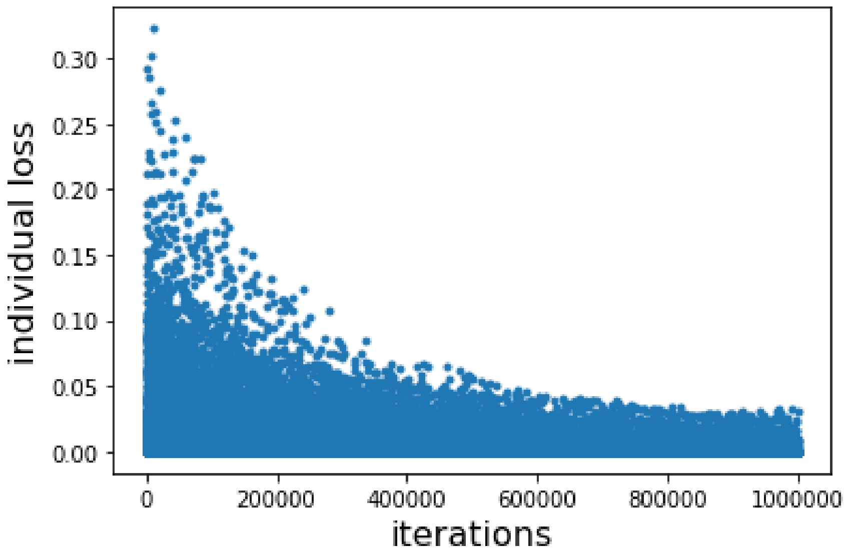

Figure 3.6 – Loss function trained on Raw dataset

. Studies on the HandReader database were done under two scenarios: without any preparation and with the preparation mentioned in the methods section. Each image feature vector is decreased to 99 dimensions after PCA has been applied .

Figures 3.6 and 3.7 show the loss function graphs after 1,000,000 iterations.

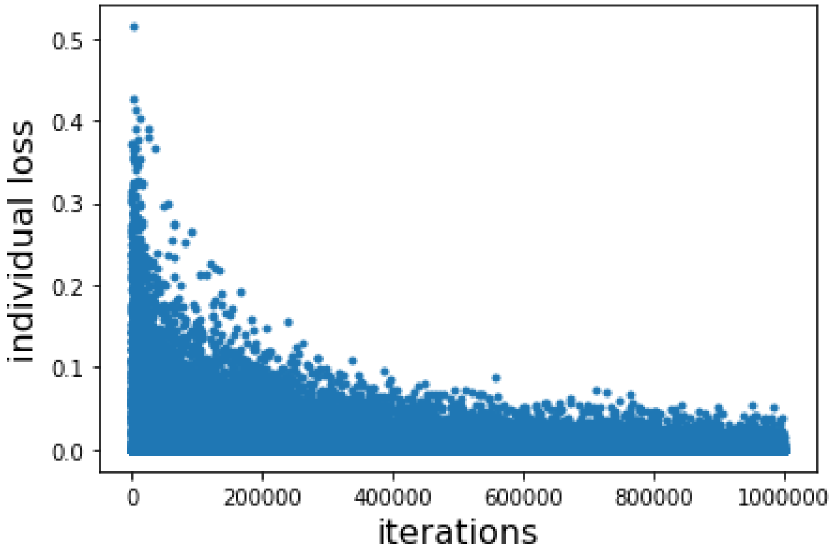

Figure 3.7 – Loss function trained on Preprocessed dataset

After learning on raw data, the loss stays close to 0.1 while falling below 0.05 from before the data. We also observe that the loss function remains stable in both scenarios after 600,000 workshops.

Finally, we give the results of the experiment for the original data and analyze them to KNN and MSRC. The k Nearest Neighbor approach was utilized in three not same situations where k had the variables 1, 3, and 5. According to table 3.4, it is important to subtract k from odd integers, and since the accuracy reduced as k grew, we ended at k = 5.

Table 3.4 – Result of KNN, MSRC, PCA-TP classifier algorithms on HandReader dataset without preprocessing

Classifier	KNN(k=1)	KNN(k=3)	KNN(k=5)	MSRC	PCA-TP
Accuracy	70.25%	65.25%	64.50%	64.75%	77.25%

Our recommended PCA-TP model clearly exceeds the competition with a detection performance of 77.25 percent. Further research with PCA-TP without triplet homology produced a defection rate of 76%. Table 3.4 depicts the result of implementing the weighted measure (4) using PCA-TP. In following studies, the recognition rate for PCA-TP with metric and (3) was 75% and 77%, correspondingly. Figure 3.13 demonstrates PCA-letter-specific TP's recognition accuracy.

The recommended strategy is most accurate for the letter A (97.5%), while it is least accurate for the letter D (52.5%).

As indicated in the previous section, we now report the results of our experiments conducted on precompiled data. Table 3.5 provides a summary of the key results.

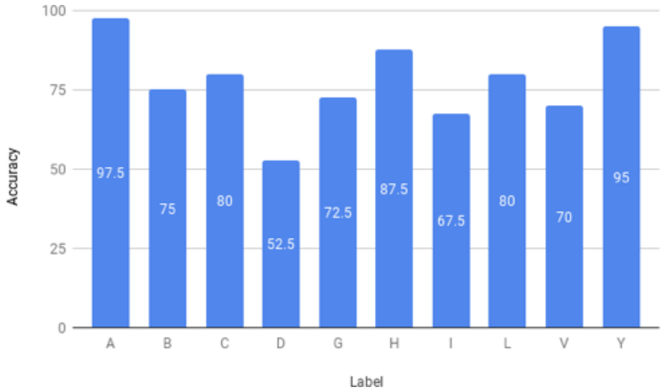

Figure 3.8 – Recognition accuracy for individual signs on a raw dataset

Table 3.5 – Result of KNN, MSRC, PCA-TP classifier algorithms on HandReader dataset with preprocessing

Classifier	KNN(k=1)	KNN(k=3)	KNN(k=5)	MSRC	PCA-TP
Accuracy	91%	88.50%	84.75%	90.75%	95%

Our suggested model outperforms standard recognizers with an average accuracy of 95%. Both raw and precompiled dataset evaluations demonstrate that KNN outperforms MSRC. Excluding triplet homology, the experimental PCA-TP result would be 94% accurate. In comparison, considering metrics (2) and (3), the accuracy of PCA-TP is 93.75 and 94.25%, correspondingly. The letter-specific accuracy of PCA-TP identification is seen in figure 3.14.

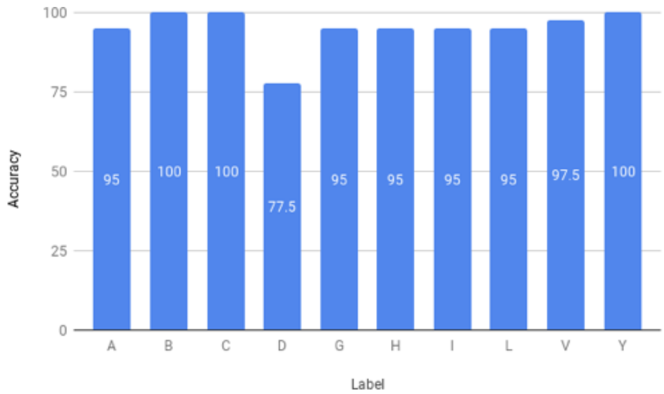

Figure 3.9 – Recognition accuracy for individual signs on a preprocessed dataset

In the maximum cases, PCA-TP fails to detect the letter "D," as seen in figure 3.9 According to a detailed investigation, "D" is wrongly identified as "I" in the majority of situations.

3.3.2 Experiment Discussion

In this research, we analyzed k-NN, MSRC, and our novel approach (PCA-TP) for hand gesture detection. Results indicate that PCA-TP beats both other methods. Tofighi et al [154]. examined having similar dataset using a more comprehensive preparation and a version of Support Vector Machine classifier with straight and RBF kernel, showing that in the linear situation, the accuracy goes to 95% [], but our PCA-TP obtains 95% accuracy as shown in table 3.5. In addition, they examined the use of Gaussian Blur, the squares picture canvas, etc. In much of their research, the letter "G" was recognized wrongly as the letter "H." In our situation, PCA-TP misidentified the letter "D" as the letter "I." Using both techniques will certainly increase precision. As mentioned earlier, a recent research investigated the topic of hand gesture recognition utilizing a subset of the HandReader dataset. They examined the detection accuracy of KNN, MSRC, and RF classifications employing four out of ten letters, namely A, B, C, and D. In particular, they evaluated the effectiveness of the resized images and the identification speed. MSRC fared the best through each case, led by KNN, while RF achieved the worst. Based on the size of the photos, the stated MSRC recognition rate is between 95 and 97%. MSRC's performance dropped to 90.75 percent when the whole HandReader dataset was added to our situation, but our suggested model obtained 95 percent accuracy. In addition to offering a unique technique, our strategy may be considered as an effort to combine processes using linear algebra with nonlinear approaches based on machine learning. We see that the impact from triplet identity embedding (machine learning) is roughly 1%, which is satisfactory; nevertheless, it would be intriguing to investigate if various kinds of triplet embedding functions have the right performance in future studies. As stated earlier, although deep learning builds such as NN convolutional models offer positive outcomes in machine vision, the models suggested in this research have two significant benefits: resilience and efficacy on restricted databases.

3.4 Two-dimensional facial recognition with PCA and triplet similarity implantation

The objective of this experiment is to evaluate the effectiveness of a hybrid two-dimensional facial recognition framework that combines PCA, Triplet Similarity Learning, and Projection-based classification (PCA-TP). Using the ORL dataset introduced in Section 2.1.1, this experiment aims to assess how the integration of dimensionality reduction, similarity-based embedding, and projection mechanisms improves recognition performance compared to conventional classifiers such as (KNN). The specific goals of this experiment investigate how different training-to-testing split ratios influence recognition accuracy. Compare the classification performance of the proposed PCA-TP model with PCA-only and KNN methods. Determine whether Triplet Similarity Learning enhances the discriminative capability of PCA features, particularly when the amount of training data is limited. Overall, this experiment seeks to validate that the PCA-TP approach provides generalizable framework for facial recognition on the ORL dataset, yielding superior accuracy and stability across multiple evaluation settings.

3.4.1 Experiment results

Now, we shall discuss the outputs of this experiment. We contrast the accuracy of our finding with that of the k-Nearest Neighbor classifier (KNN). For further details regarding KNN, because k values bigger than one underperform, we only show the accuracy of KNN when k=1 because k values larger than one underperform. The outputs of our experiments are shown in table 3.6.

Table 3.6 – Result of KNN, PCA-T, PCA-TP classifier algorithms on ORL dataset

Train/Test split % PCA-TP		PCA-T	KNN(k=1)
80/20	98.75	98.75	95
60/40	97.5	97.5	95
40/60	93.33	92.5%	92.5
20/80	84.375	81.56	81.88

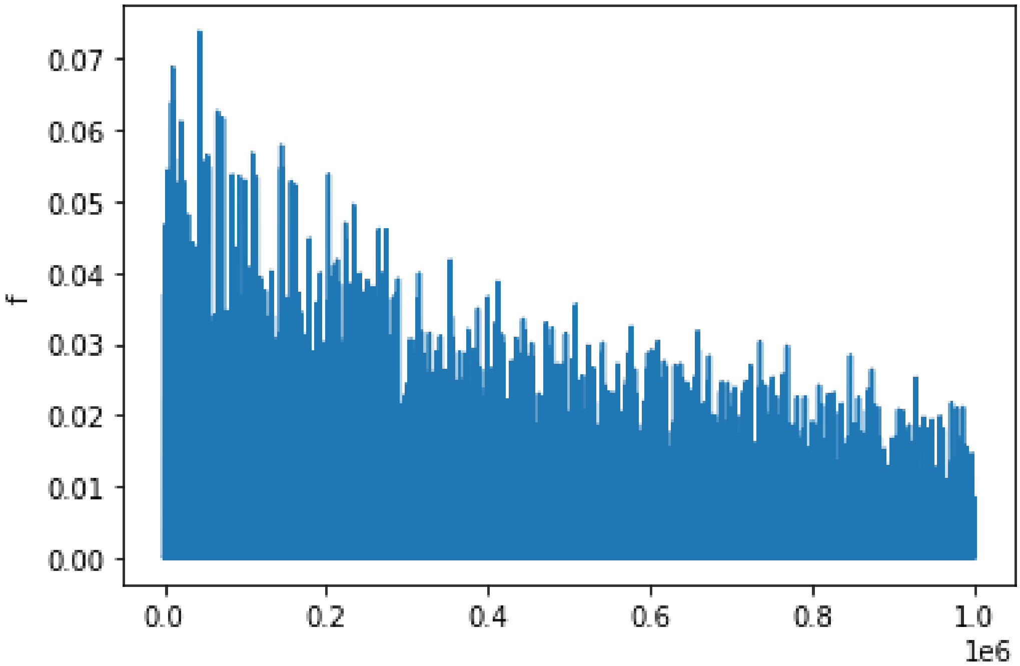

Figure 3.10 – Similarity embedding with SGD (80%)

Figures 3.10, 3.11, 3.12, 3.13, 3.14 illustrate the demonstrations of the loss function f (similarity embedding transformation) as a consequence of learning the similarity embedding using SGD.

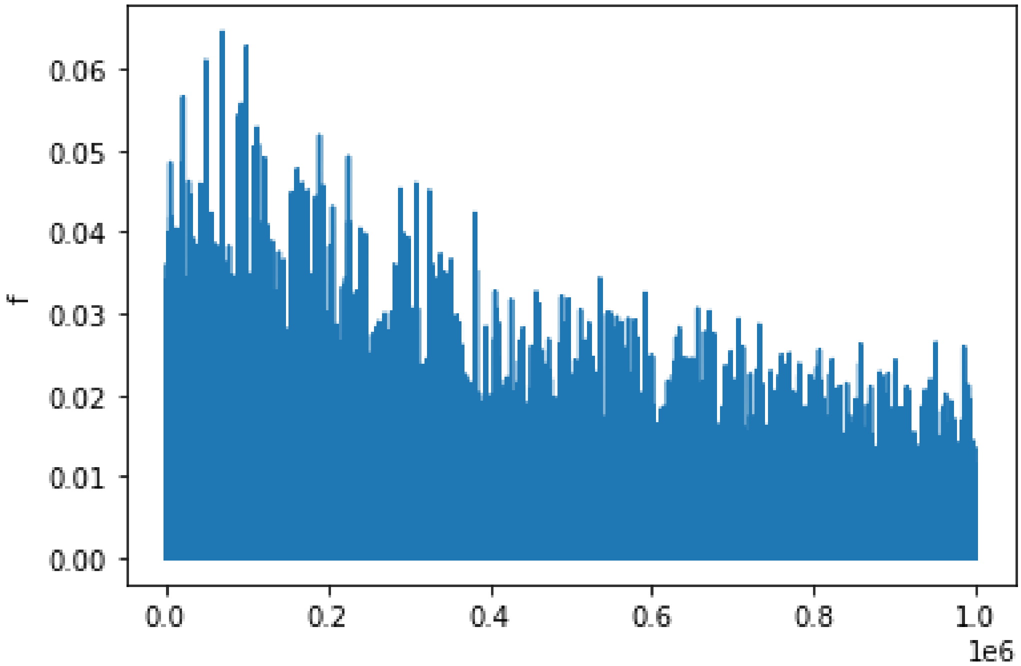

Figure 3.11 – Similarity embedding with SGD (60%)

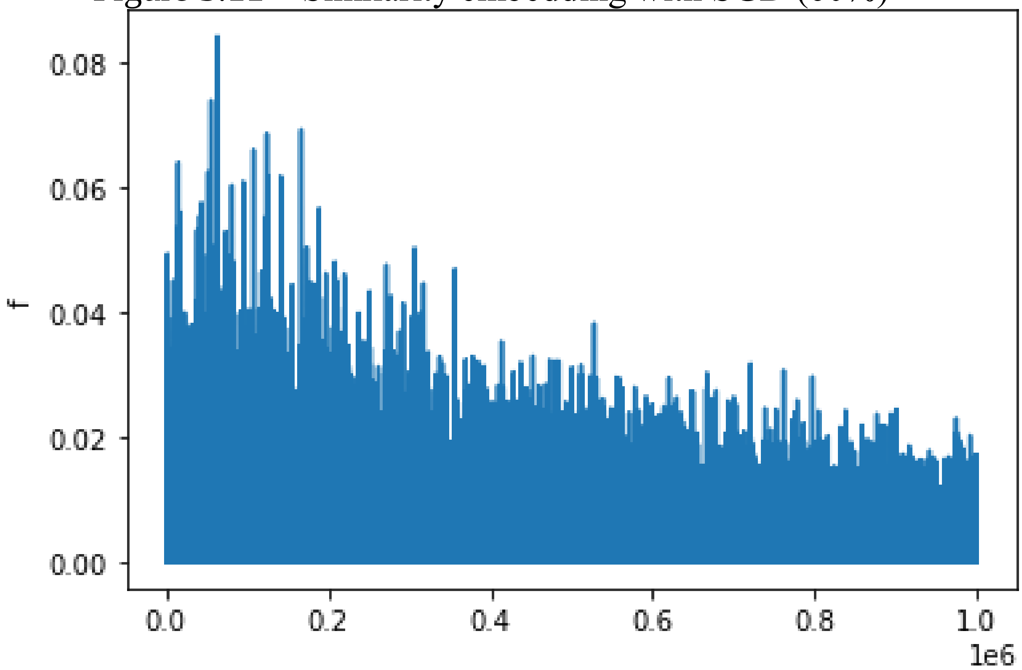

Figure 3.12 – Similarity embedding with SGD (40%)

This research evaluates the performance of KNN with PCA-TP on the ORL dataset. Under four separate conditions, the dataset was randomly split into training and test sets with ratios of 80/20, 60/40, 40/60, and 20/80.

PCA-TP outperforms KNN on a consistent basis, as illustrated in Table 1 above. With up to 98.75% accuracy when 80% of the information is given to the training set, it is obvious that the accuracy score rises as more training data is included. We observe that as the teaching set develops, the similarity embedding technique does not considerably enhance the classifier.

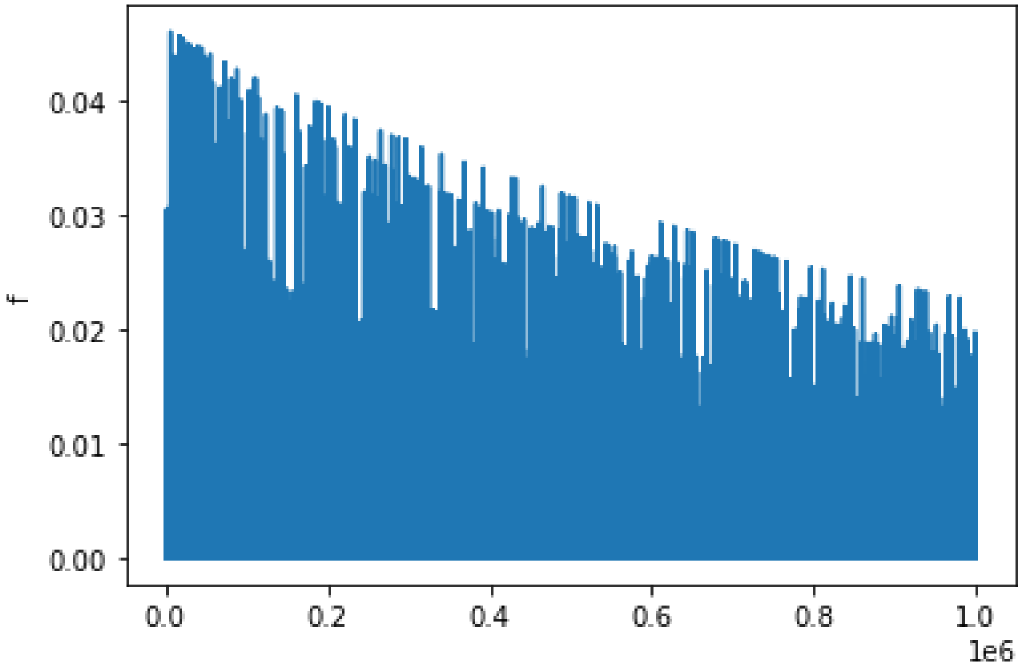

Figure 3.13 – Similarity embedding with SGD (20%)

Figure 3.14 displays some of the most often mislabeled pictures and associated false positive things.

Figure 3.14 – The commonly mislabeled images and their false positives

In comparison, the benefit from triplet similarity encoding is roughly 3% when the training set comprises as little as 20% of the database.

3.4.2 Experiment discussion

Recent research by Hu et al. [155] evaluates the effectiveness of their CNN-based facial recognition method using the ORL dataset. They claim a recognition accuracy of 95% when 80% of the dataset is used for training. Our suggested model surpasses the competition, since our identification accuracy for the same case of splitting was 98.75%. The recommended classifier PCA-TP may be improved for future work by modifying the triplet similarity embedding function.

3.5 Face Recognition in Low-Data Regimes with PCA-TP

This section presents a comprehensive experimental study comparing a proposed facial recognition method with two baseline algorithms using the ORL dataset. The aim is to evaluate the performance of a novel approach that combines PCA, Triplet Similarity Loss, and a projection-based (PCA_TP) decision metric under controlled and reproducible conditions. The experiment is designed to follow the exact preprocessing procedures of a previously published CNN-GRNN method, which reported high classification accuracy on small-sample data. By applying the same image preprocessing and evaluating performance at various training ratios, the study provides a clear and fair comparison of algorithmic effectiveness [156].

3.5.1 Baseline Overview

The first baseline used in this study is a previously developed classifier that combines PCA with Triplet Similarity Embedding and a projection method (PCA-TP). In this earlier configuration explained in 3.4 chapter, the model achieved strong performance across various train-test splits. For instance, when trained on 80% of the data, PCA-TP achieved 98.75% accuracy, and 97.5% accuracy on a 60% training split. These results were already significantly better than a KNN classifier, which only achieved 95% at best. However, this earlier model did not incorporate the preprocessing strategies used in the CNN-GRNN paper and was not evaluated under the exact same training scenarios as that method.

The second baseline is a CNN-based architecture enhanced with dictionary filters and a General Regression Neural Network (GRNN), as described in the second referenced paper. This model was evaluated on the ORL dataset after resizing the images to 36×36 pixels and applying a smoothing filter. It achieved a reported accuracy of 93.96% using 30% of the dataset for training, a strong result for such a small training set. Given its preprocessing pipeline and relatively high accuracy, this method was selected as the main benchmark for comparing the proposed method under equivalent experimental conditions [156, p. 6].

In the following sections, the experimental setup and methodology for the proposed model are described, followed by a detailed analysis of results, including performance metrics, tuning configurations, and misclassification insights.

3.5.2 Experimental Setup and Results

Continuing from the previous section, this part outlines the experimental setup used to evaluate the proposed model, followed by a detailed presentation and analysis of the results.

In order to ensure fair and consistent comparison with the CNN-GRNN method, the proposed model was tested under identical preprocessing steps. Specifically, all images were resized from their original resolution of 112×92 pixels to 36×36 pixels. A smoothing filter was also applied prior to feature extraction to enhance contrast and reduce noise, just as performed in the baseline CNN experiment. After preprocessing, features were extracted using PCA, followed by a triplet similarity embedding technique. A projection-based classifier was then used to compute final classification scores. Unlike the earlier PCA-TP implementation, this updated experiment introduced hyperparameter tuning at each training ratio to identify the most optimal settings for learning rate, PCA components, embedding size, triplet loss margin (alpha), smoothing intensity, and number of training epochs.

To evaluate the performance of the model under different data availability conditions, experiments were conducted using training sets ranging from 20% to 90% of the full dataset. For each split, the model was tuned using multiple configurations, and the best-performing result was recorded. The evaluation metric used was classification accuracy, defined as the ratio of correctly identified images to the total number of test samples. Additionally, misclassified image indexes were recorded to observe patterns in prediction errors.

Table 3.7 – Result of PCA-TP with resize and smoothing, PCA-TP and CNN+GRNN classifiers algorithms on ORL dataset

Train %	Resize + Smoothing (PCA-TP)	PCA-TP (early experiment, chapter 3.4)	CNN+GRNN
90	100%	—	—
80	100%	98.75%	—
70	99.17%	—	—
60	99.38%	97.5%	—
50	96.00%	—	—
40	95.83%	93.33%	—
30	95.00%	—	93.96%
20	91.25%	84.375%	—

In the case of 90% and 80% training data, the proposed model achieved perfect classification accuracy of 100%. These results demonstrate the ability of the algorithm to generalize well when ample training data is available. When training data was reduced to 70% (280 training, 120 testing), only one sample was misclassified, resulting in an accuracy of 99.17%. The best configuration for this split used 35 PCA components, 48-dimensional embedding, smoothing value of 0.01, learning rate of 0.0005, and 12 training epochs. A similar result was observed with 60% training data (240 training, 160 testing), where again only one image was incorrectly classified, yielding an accuracy of 99.38%. The configuration that produced this outcome included a slightly adjusted learning rate and embedding dimension.

At 50% training data (200 training, 200 testing), accuracy dropped to 96% with eight mislabeled images. Misclassification analysis revealed consistent confusion between certain identities, with subjects 23 and 35 frequently mistaken for subject 16, and subject 36 misclassified as 27. These cases suggest that specific individuals in the ORL dataset exhibit overlapping features or pose similarities, which become more challenging to separate with reduced training data.

With 40% training (160 training, 240 testing), the model misclassified ten images, achieving an accuracy of 95.83%. At this stage, further drops in performance began to emerge. Using 30% of the dataset for training (120 images), the model still achieved a notable 95% accuracy, surpassing the CNN-GRNN baseline (93.96%) reported under the same conditions. This demonstrates the effectiveness of combining dimensionality reduction with triplet-based distance learning, especially when combined with appropriate preprocessing. However, the misclassification count increased to 14, and several identities were confused multiple times, including subject 26, which was misclassified into three different identities.

In the lowest data scenario with only 20% of the data for training (80 images), the model's performance dropped to 91.25%, with 28 mislabeled samples. Misclassified pairs in this scenario reflected a mix of frequently confused subjects and some sporadic errors, likely due to underrepresentation of those identities in the training set. Nevertheless, even under this most constrained condition, the model maintained relatively high accuracy, demonstrating good generalization for small-sample learning tasks.

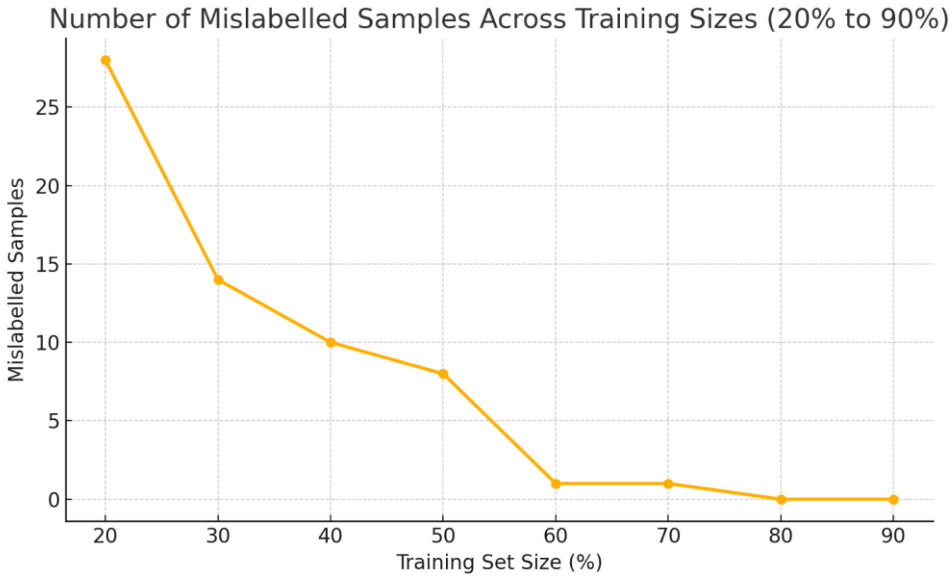

Figure 3.15 – Number of mislabeled samples across training sizes (20% to 90%)

The figure 3.15 illustrates how the number of classification errors made by the proposed PCA + Triplet Similarity + Projection method changes as the proportion of training data increases. The overall trend is clearly downward, showing a strong inverse relationship between training data volume and misclassification rate. This is an expected outcome in supervised learning: as the model is exposed to more representative samples during training, its ability to generalize improves significantly.

At the lowest training level of 20%, the model misclassified 28 images, which reflects the challenge of learning complex decision boundaries from very limited data. Interestingly, the drop in errors from 20% to 30% training is relatively steep, with errors reducing from 28 to 14, demonstrating the model's rapid adaptation when exposed to just a slightly larger training set. From 30% to 60%, the rate of decline slows, indicating that while more data improves performance, the marginal gains begin to taper as the model stabilizes.

At 70% training data, only a single image was misclassified, and at both 80% and 90% training levels, the model achieved zero classification errors. This confirms the model's capacity to fully memorize or generalize correctly when the majority of the dataset is used for training.

3.5.3 Comparison of three works

The comparison of classification accuracy across different training proportions reveals several significant observations regarding the performance of the evaluated algorithms. Among the three methods, the proposed model combining PCA, Triplet Similarity Loss, and a projection-based metric consistently delivers the highest accuracy, particularly under limited training conditions.

At the 30% training level, which is a common benchmark for small-sample learning, the CNN+GRNN method achieved a reported accuracy of 93.96%, demonstrating its strength as a deep learning-based model enhanced by preprocessing techniques such as image resizing and smoothing. The previously proposed PCA-TP method did not report performance for this specific ratio, but under other similar constraints (20% and 40% training), it achieved 84.375% and 93.33% respectively. In contrast, the newly proposed model reached 95.00% accuracy at 30% training data, exceeding the CNN+GRNN benchmark and significantly outperforming the earlier PCA-based approach. This suggests that integrating metric learning through triplet loss provides a more discriminative embedding space, even with limited data.

As the proportion of training data increases, the accuracy gap between the methods becomes more pronounced. At 60% training data, both the PCA-TP model and the new method achieve 97.5% or higher, but only the new method was explicitly tuned and evaluated at this ratio with preprocessing included, where it achieved 99.38%. The CNN+GRNN method did not report results at this level, leaving the comparison inconclusive for that setting.

At higher training ratios, the superiority of the proposed method becomes evident. While the previous PCA-TP method reached 98.75% accuracy at 80% training, the new model achieved a perfect 100% classification rate for both 80% and 90% training configurations. These results illustrate that the enhancements introduced through fine-tuned triplet similarity embedding and data preprocessing contribute significantly to performance and generalization. Moreover, the new method's strong performance at both low and high training data levels reflects its adaptability across different sample size scenarios.

Importantly, while deep learning approaches like CNN+GRNN offer compelling performance, they often require more complex architectures and larger datasets to generalize effectively. In contrast, the proposed method remains computationally lightweight, yet manages to surpass the deep learning benchmark under similar experimental settings. This makes it highly suitable for resource-constrained environments or applications where labeled data is limited.

In conclusion, across all reported training proportions, the newly proposed PCA + Triplet Similarity + Projection method outperforms both the CNN+GRNN and the original PCA-TP baseline. The consistent improvement, especially in small-sample scenarios, confirms the effectiveness of combining dimensionality reduction, similarity-based embedding, and projection-based classification in facial recognition tasks using the ORL dataset.

3.5.4 Mislabeled individual report

Now in this section, we analyze the misclassification behavior of the proposed model using 20% of the ORL dataset for training. This scenario represents the most limited data condition in the experiment, where generalization is expected to be challenging. The analysis focuses on identifying which individuals were most frequently misclassified and which identity pairs exhibited the highest confusion frequency.

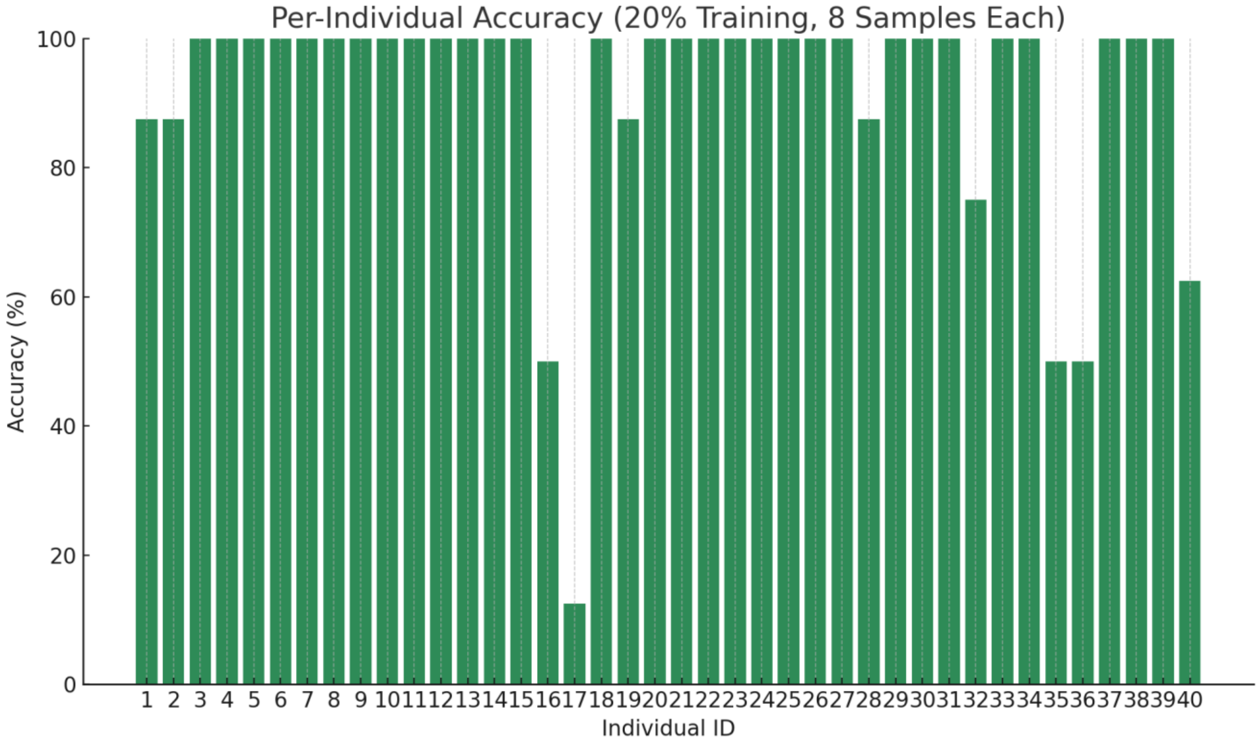

Figure 3.16 – Per-individual accuracy (20% training, 8 samples each)

The results illustrated in figure 3.23 that some individuals consistently present challenges for the classifier. According to the Most Misclassified Individuals table, Individual 17 was the most misclassified, appearing as the true identity in 7 incorrect predictions. This suggests that the model has difficulty learning a reliable representation for this individual under limited training conditions. Similarly, Individuals 16, 35 and 36 were misclassified 4 times each. These values indicate not just random noise, but systematic confusion, possibly due to overlapping facial features or variations in image conditions (e.g., lighting, pose).

More specific insight comes from the table 3.8. The most frequent mislabelled pair was '1 - 16', which occurred 3 times. This means that images of Individual 16 were incorrectly predicted as Individual 1 in three separate instances. Such repetition strongly implies a feature similarity in the embedding space between these two individuals, making them visually confusable for the classifier. Similarly, '5 - 40', '24 - 17' and '24 - 36' also occurred multiple times, further suggesting instability in recognizing Individuals 17, 36, and 40. These consistent misclassification patterns point toward possible refinement needs in the embedding learning process, especially under constrained data regimes.

Table 3.8 - Most Common Misclassification Pairs (20% Training)

Predicted - Actual (Ind #)	Count
5 - 40	3
24 - 17	3
24 - 36	3
1 - 16	3
10 - 17	2
36 - 32	1
7 - 36	1
40 - 35	1
27 - 35	1
12 - 35	1
25 - 35	1
32 - 1	1
2 - 32	1
27 - 2	1
11 - 19	1
6 - 17	1
12 - 17	1
8 - 16	1
37 - 28	1

This analysis highlights not only the overall drop in performance under small training sets but also the fact that certain identities contribute disproportionately to the classification errors. Addressing these may involve localized training enhancements, sample re-weighting, or more discriminative embedding strategies. It also shows the importance of per-identity evaluation, as some individuals maintain perfect accuracy while others dominate the error distribution.

3.6 Evaluation of the Modified Triplet Loss Function

The objective of this final experiment is to evaluate the performance improvement introduced by the modified triplet loss function, which incorporates adaptive margin and projection-consistency regularization within the PCA-based facial recognition framework. While previous experiments focused on the fixed-margin PCA + Triplet + Projection (PCA-TP) method, this experiment isolates and examines the contribution of the proposed algorithmic enhancement described in Section 2.3.3.4. Testing was conducted on the ORL dataset using identical preprocessing and evaluation protocols to ensure a direct and fair comparison. The assessment concentrated on the most challenging training conditions—20 % and 30 % of the total data—where generalization is typically most difficult. Under such limited-data scenarios, discriminative embedding quality, and the ability to avoid overfitting are the key performance indicators.

Experiments with higher training ratios (≥ 60 %) were intentionally omitted here because both models already achieved near-perfect classification accuracy under abundant data, leaving the low-data regime as the most informative for algorithmic comparison. The experimental pipeline followed the same preprocessing and learning procedure as described in Section 3.5: Input images resized to 36 × 36 pixels and smoothed with a Gaussian filter to reduce noise. PCA applied for dimensionality reduction, producing compact linearly decorrelated features. Linear embedding layer trained using the modified cosine-based triplet loss with adaptive margin and projection-consistency regularization. Hyperparameters tuned for base margin (α_0), adaptive coefficient (β), and projection weight (λ). Classification performed using cosine 1-nearest-neighbor (1-NN) in the learned embedding space. Two training ratios were evaluated: 20 % (80 training / 320 testing) and 30 % (120 training / 280 testing). The main metric was classification accuracy, calculated as the ratio of correctly classified samples to the total number of test images.

3.6.1 Experimental Results

The quantitative results of both algorithms under identical preprocessing and training configurations are summarized in Table 3.11.

Table 3.9– Comparative Results of PCA-TP and Modified PCA-TP

Training Ratio	PCA-TP Accuracy (%)	Modified PCA-TP Accuracy (%)	Absolute Gain (%)
30 %	95.00	96.79	+1.79
20 %	91.25	92.19	+0.94

The proposed modification yields a measurable performance improvement in both training scenarios, with the most notable gain observed at the 30 % training level.

The results confirm that incorporating an adaptive margin improves inter-class separability by dynamically adjusting the margin for difficult examples, while projection-consistency regularization further refines intra-class alignment. These effects collectively lead to enhanced recognition accuracy even when the number of training samples per subject is small.

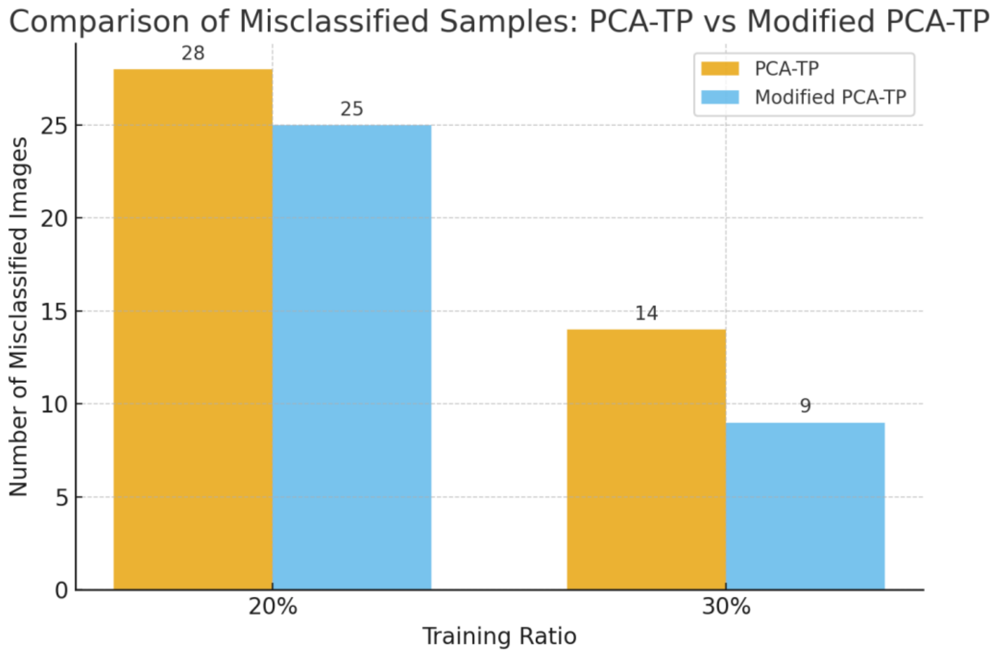

Figure 3.17 – Comparison of Misclassified Samples between PCA-TP and Modified PCA-TP

To better illustrate the improvement, the total number of misclassified images was also compared between the two algorithms. The Modified PCA-TP consistently reduced misclassification counts at both training levels, as shown in Figure 3.27. At 20 % training, the number of mislabelled samples decreased from 28 to 25, while at 30 % training, it dropped from 14 to 9. This corresponds to an overall 18 % reduction in classification errors, confirming that the proposed adaptive mechanism provides greater resilience under limited-data conditions.

The statistical comparison clearly demonstrates that the modified formulation delivers consistent, measurable gains in both accuracy and error reduction. By introducing an adaptive margin, the model allocates greater learning emphasis to challenging sample pairs, while the projection-consistency term ensures that learned embeddings maintain stable class geometry. As a result, the Modified PCA-TP exhibits superior discrimination and improved performance when trained on small subsets of the ORL dataset, effectively addressing one of the major limitations of traditional fixed-margin triplet formulations.

CONCLUSION

The primary goal of this dissertation was to analyze and develop an algorithm and method for object recognition capable of achieving high accuracy under limited-data conditions.

The research began by exploring the two dominant paradigms in object recognition, traditional machine learning algorithms and modern deep learning approaches and identifying their respective advantages and constraints. Traditional algorithms such as KNN, SVM, and Linear Discriminant Analysis (LDA) provided interpretable and computationally efficient solutions but were heavily dependent on handcrafted features and struggled to generalize to complex, high-dimensional data. In contrast, deep learning architectures such as CNN and transfer learning models demonstrated remarkable performance on large-scale datasets but remained data- and resource-intensive, often leading to overfitting when applied to small datasets.

Given these contrasting characteristics, this study focused on bridging the gap between traditional and deep learning paradigms by developing a hybrid, lightweight, and data-efficient recognition framework. The proposed method combines PCA for dimensionality reduction with Triplet Similarity Learning and a Projection-Based Classifier, effectively integrating statistical projection techniques with discriminative metric learning. This hybrid design leverages the interpretability and low computational cost of PCA while incorporating the representational power of embedding learning to improve class separability in feature space.

In addition to the methodological integration, the research introduced a Modified Triplet Loss Function that employs Adaptive Margin Adjustment and Projection-Consistency Regularization. Unlike the fixed-margin triplet formulation, the adaptive margin dynamically modifies the decision boundary according to sample similarity, emphasizing harder examples and reducing overlap between classes. The projection-consistency term further stabilizes the embedding space by aligning learned representations with their respective class centroids, thereby enhancing intra-class compactness and inter-class separation. Together, these modifications yield a more discriminative and geometrically consistent embedding representation suitable for limited-data scenarios.

Experimental evaluations were performed using two datasets, the ORL facial dataset and the HandReader gesture dataset, both representing small-sample environments. The baseline PCA–TP model demonstrated excellent performance, achieving up to 100% accuracy at higher training ratios (80–90%) and maintaining with reduced data (95% at 30%, 91.25% at 20%). The Modified PCA–TP algorithm further improved performance under constrained data, reaching 96.79% accuracy at 30% training and 92.19% at 20% training, with approximately 18% fewer misclassifications than the baseline. These outcomes confirm that the adaptive margin and projection-consistency components effectively enhance recognition accuracy, stability, and generalization even with minimal training samples.

Through this research, both a method (PCA + Triplet + Projection) and an algorithm (Modified Triplet Loss with Adaptive Margin and Projection Regularization) were developed and validated. The findings demonstrate that the combination of classical dimensionality reduction and metric-based learning can achieve competitive or superior accuracy compared to deep networks while remaining computationally efficient and interpretable. The approach thus represents a practical alternative for real-world applications where data availability and hardware resources are limited, such as embedded systems, security devices, and small-scale industrial automation.

In conclusion, this dissertation provides a comprehensive analysis of object recognition techniques and presents an effective solution for limited-data learning. By uniting the principles of statistical projection, similarity-based embedding, and adaptive optimization, the proposed method offers a balanced trade-off between accuracy, efficiency, and interpretability. The results confirm that high-performance object recognition does not necessarily require large-scale deep architectures which can also be achieved through well-structured hybrid models that intelligently combine traditional and modern paradigms.

REFERENCES

1 Sharma P., Kaur M. Classification in pattern recognition: A review // International Journal of Advanced Research in Computer Science and Software Engineering. – 2013. – Vol. 3, Issue 4. – P. 298-306.

2 Watanabe S. Pattern recognition: human and mechanical. – Hoboken: John Wiley & Sons, Inc., 1985. – 520 p.

3 Jain A.K., Duin R.P.W. Introduction to pattern recognition // In book: The Oxford Companion to the Mind. – Oxford, 2004. – P. 698-703.

4 Nagy G. State of the art in pattern recognition // Proceedings of the IEEE. – 1968. – Vol. 56, Issue 5. – P. 836-863.

5 Fu K.-S. Recent developments in pattern recognition // IEEE Transactions on computers. – 1980. – Vol. 100, Issue 10. – P. 845-854.

6 Bansal M., Kumar M., Kumar M. 2D object recognition techniques: state-of-the-art work //Archives of Computational Methods in Engineering. – 2021. – Т. 28. – №. 3. – С. 1147-1161.

7 Mateusz Buda, Atsuto Maki, Maciej A. Mazurowski A systematic study of the class imbalance problem in convolutional neural networks // Neural Networks. – 2018. – Vol. 106, – P. 249-259

8 Shorten, C., & Khoshgoftaar, T. M. A survey on image data augmentation for deep learning // Journal of Big Data. – 2019. – Vol. 6, No. 1. – P. 60.

9 Schneider, S., Greenberg, S., & Taylor, G. W. Three critical factors affecting automated image species recognition performance for camera traps // Ecology and Evolution. – 2020. – Vol. 10, No. 7. – P. 3503–3517.

10 Hinton G.E., Salakhutdinov R.R. Reducing the dimensionality of data with neural networks // Science. – 2006. – Vol. 313, Issue 5786. – P. 504-507.

11 Lecun Y. et al. Deep learning // Nature. – 2015. – Vol. 521, Issue 7553. – P. 436-444.

12 Nan F. et al. Feature super-resolution based facial expression recognition for multi-scale low-resolution images //Knowledge-Based Systems. – 2022. – Т. 236. – С. 107678.

13 Molchanov D., Ashukha A., Vetrov D. Variational dropout sparsifies deep neural networks //International conference on machine learning. – PMLR, 2017. – С. 2498-2507.

14 Koller O. Quantitative survey of the state of the art in sign language recognition //arXiv preprint arXiv:2008.09918. – 2020.

15 Bugaje A. A. B., Cremer J. L., Strbac G. Generating quality datasets for real-time security assessment: Balancing historically relevant and rare feasible operating conditions //International Journal of Electrical Power & Energy Systems. – 2023. – Т. 154. – С. 109427.

16 Mittal S. et al. On responsible machine learning datasets emphasizing fairness, privacy and regulatory norms with examples in biometrics and healthcare //Nature Machine Intelligence. – 2024. – Т. 6. – №. 8. – С. 936-949.

17 Tomar V. et al. Two-Level Image Augmentation with Deep-Learning for Single Sample Face and Ear Recognition //2024 IEEE Silchar Subsection Conference (SILCON 2024). – IEEE, 2024. – С. 1-6.

18 Ghali A. et al. Few-shot learning CNN optimized using combined 2D-DWT injection and evolutionary optimization algorithms for human face recognition //International Journal of Wavelets, Multiresolution and Information Processing. – 2023. – Т. 21. – №. 06. – С. 2350024.

19 Zheng W., Yang S., Zhang X. A Novel Privacy-Preserving Computing System Based on VAE Federated Meta-Learning //Journal of Physics: Conference Series. – IOP Publishing, 2022. – Т. 2363. – №. 1. – С. 012027.

20 Trigka M., Dritsas E. A comprehensive survey of machine learning techniques and models for object detection //Sensors. – 2025. – Т. 25. – №. 1. – С. 214.

21 Hall D. J. et al. Development of New Pattern-Recognition Methods. – 1973.

22 Ludmila I Kuncheva. Combining pattern classifiers: methods and algorithms. – New Jersey, 2014. – 384 p.

23 Amancio D. R. et al. A systematic comparison of supervised classifiers //PloS one. – 2014. – Т. 9. – №. 4. – С. e94137.

24 Ahlawat S. et al. Improved handwritten digit recognition using convolutional neural networks (CNN) //Sensors. – 2020. – Т. 20. – №. 12. – С. 3344.

25 Zhai J. et al. A review of the evolution of deep learning architectures and comparison of their performances for histopathologic cancer detection //Procedia Manufacturing. – 2020. – Т. 46. – С. 683-689.

26 Andri R. et al. YodaNN: An ultra-low power convolutional neural network accelerator based on binary weights //2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI). – IEEE, 2016. – С. 236-241.

27 Lampert C.H. Kernel methods in computer vision. – Hanover, 2009. – 112 p.

28 Lu J., Zhang E. Gait recognition for human identification based on ICA and fuzzy SVM through multiple views fusion //Pattern Recognition Letters. – 2007. – Т. 28. – №. 16. – С. 2401-2411.

29 Bishop C. M. Neural networks for pattern recognition. – Oxford university press, 1995.

30 Ahad M. et al. Lower-dimensional feature sets for template-based motion recognition approaches //Journal of Computer Science. – 2010. – Т. 6. – №. 8. – С. 920.

31 Escobar M. D., West M. Bayesian density estimation and inference using mixtures //Journal of the american statistical association. – 1995. – Т. 90. – №. 430. – С. 577-588.

32 Wilamowski B. M., Jaeger R. C. Implementation of RBF type networks by MLP networks //Proceedings of International Conference on Neural Networks (ICNN'96). – IEEE, 1996. – Т. 3. – С. 1670-1675.

33 Duda R.O., Hart P.E., Stork D.G. Pattern classification and scene analysis. – NY.: Wiley, 1973. – Vol. 3. – 520 p.

34 Liu H.-H., Fu K.S. A syntactic pattern recognition approach to seismic discrimination // Geoexploration. – 1982. – Vol. 20, Issue 1-2. – P. 183-196.

35 Fu K.S., Albus J.E. et al. Syntactic pattern recognition, applications. – Berlin, 2012. – 272 p.

36 Perlovsky L.I. Conundrum of combinatorial complexity // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1998. – Vol. 20, Issue 6. – P. 666-670.

37 Kohonen T. Self-Organizing Maps. – Berlin, 1995. – 362 p.

38 Ripley B.D. Statistical aspects of neural networks // In book: Networks and chaos-statistical and probabilistic aspects. – London, 1993. – P. 40-123.

39 Anderson J.A., Rosenfeld E., Pellionisz A. Neurocomputing. – Cambridge: MIT press, 1988. – 583 p.

40 Zhao W. Robust image based 3D face recognition. College Park, MDЖ University of Maryland, 1999. – 296 p.

41 Fromherz T. Face recognition: A summary of 1995-1997: tech. rep. – Berkeley, 1998. – 8 p.

42 Bruner J. The perception of people // In book: Handbook of social psychology. – NY., 1954. – Vol. 2. – P. 634-654.

43 Bledsoe W.W. The model method in facial: techn. Report. – Palo Alto, 1966. – 25 p.

44 Fu K. S. Syntactic pattern recognition //Applications of Pattern Recognition. – CRC Press, 2019. – C. 37-64.

45 Chelali F. Z., Djeradi A. Face recognition using MLP and RBF neural network with Gabor and discrete wavelet transform characterization: a comparative study //Mathematical Problems in Engineering. – 2015. – T. 2015. – №. 1. – C. 523603.

46 Naskath J., Sivakamasundari G., Begum A. A. S. A study on different deep learning algorithms used in deep neural nets: MLP SOM and DBN //Wireless personal communications. – 2023. – T. 128. – №. 4. – C. 2913-2936.

47 Ripley B. D. Pattern recognition and neural networks. – Cambridge university press, 2007.

48 Anderson B. Pattern recognition: An introduction. – Scientific e-Resources, 2019.

49 Tamura S., Kawai H., Mitsumoto H. Male/female identification from 8×6 very low resolution face images by neural network // Pattern recognition. – 1996. – Vol. 29, Issue 2. – P. 331-335.

50 Koutroumbas K., Theodoridis S. Pattern recognition. – Academic Press, 2008.

51 Picard R. W. Affective computing. – MIT press, 2000.

52 Voronov V. I. et al. Designing a neural network identification subsystem in the hardware-software complex of face recognition //T-Comm-Телекоммуникации и Транспорт. – 2020. – T. 14. – №. 5. – C. 69-76.

53 Abiodun O. I. et al. Comprehensive review of artificial neural network applications to pattern recognition //IEEE access. – 2019. – T. 7. – C. 158820-158846.

54 Weiss S. M., Kapouleas I. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods //IJCAI. – 1989. – T. 89. – C. 781-787.

55 Bauskar S. View of Unveiling the Hidden Patterns AI-Driven Innovations in Image Processing and Acoustic Signal Detection //Journal of Recent Trends in Computer Science and Engineering. – 2020. – T. 8. – №. 1. – C. 10.70589.

56 Sharma P. Machine Learning and Pattern Recognition: Techniques, Applications, Challenges and Future Perspectives //International Journal of Engineering, Science and Humanities. – 2021. – Т. 11. – №. 2. – С. 18-25.
57 Chen L. et al. A survey on hand gesture recognition // Procced. internat. conf. on computer sciences and applications. – Wuhan, 2013. – P. 313-316.
58 Mitra S., Acharya T. Gesture recognition: A survey // IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). – 2007. – Vol. 37, Issue 3. – P. 311-324.
59 Yasen M., Jusoh S. A systematic review on hand gesture recognition techniques, challenges and applications // PeerJ Computer Science. – 2019. – Vol. 5. – P. e218-1-e218-30.
60 Vaibhavi S.G. et al. A review of various gesture recognition techniques // International Journal of Engineering and Computer Science. – 2014. – Vol. 3. – P. 8202-8206.
61 Chanu O.R. et al. Comparative study for vision based and data based hand gesture recognition technique // Procced. internat. conf. on Intelligent Communication and Computational Techniques (ICCT). – Jaipur, 2017. – P. 26-31.
62 Yusnita L. et al. Implementation of real-time static hand gesture recognition using artificial neural network // Procced. 4th internat. conf. on Computer Applications and Information Processing Technology (CAIPT). – Kuta Bali, 2017. – P. 1-6.
63 Alzubaidi L. et al. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications //Journal of Big Data. – 2023. – Т. 10. – №. 1. – С. 46.
64 Yan R. et al. Label-efficient self-supervised federated learning for tackling data heterogeneity in medical imaging //IEEE Transactions on Medical Imaging. – 2023. – Т. 42. – №. 7. – С. 1932-1943.
65 Lopez-Ledezma M., Velarde G. Cyber Security Data Science: Machine Learning Methods and Their Performance on Imbalanced Datasets //International Scientific-Practical Conference. – Cham : Springer Nature Switzerland, 2024. – С. 569-578.
66 Mumuni A., Mumuni F. Data augmentation: A comprehensive survey of modern approaches //Array. – 2022. – Т. 16. – С. 100258.
67 Kuo W. et al. Cost-sensitive active learning for intracranial hemorrhage detection //International Conference on Medical Image Computing and Computer-Assisted Intervention. – Cham : Springer International Publishing, 2018. – С. 715-723.
68 Park Y., Hauschild A. C., Heider D. Transfer learning compensates limited data, batch effects and technological heterogeneity in single-cell sequencing //NAR genomics and bioinformatics. – 2021. – Т. 3. – №. 4. – С. lqab104.
69 Said A., Yahyaoui A., Abdellatif T. HIPAA and GDPR compliance in IoT healthcare systems //International conference on model and data engineering. – Cham : Springer Nature Switzerland, 2023. – С. 198-209.
70 Sun T., Wang M., Han X. Deep Learning in Insurance Fraud Detection: Techniques, Datasets, and Emerging Trends //Journal of Banking and Financial Dynamics. – 2025. – Т. 9. – №. 8. – С. 1-11.

71 Boiano A. et al. FederNet: A Network and Device-Aware Emulation Platform for Federated Learning Benchmarking //Available at SSRN 5615841.

72 Giagias D. A comparative analysis of data parallelism and model parallelism for deep learning-based text classification. – 2025.

73 Anastasiou D. et al. Exploring Pre-training Across Domains for Few-Shot Surgical Skill Assessment //Data Engineering in Medical Imaging: Third MICCAI Workshop, DEMI 2025, Held in Conjunction with MICCAI 2025, Daejeon, South Korea, September 27, 2025, Proceedings. – Springer Nature, 2025. – С. 212.

74 Baisholan N. et al. A Systematic Review of Machine Learning in Credit Card Fraud Detection Under Original Class Imbalance //Computers. – 2025. – Т. 14. – №. 10. – С. 437.

75 Shao Y. Imbalance Learning and Its Application on Medical Datasets : дис. – Dissertation, Göttingen, Georg-August Universität, 2021, 2021.

76 Wang C., Pino J., Gu J. Improving cross-lingual transfer learning for end-to-end speech recognition with speech translation //arXiv preprint arXiv:2006.05474. – 2020.

77 Mienye I. D., Sun Y. Performance analysis of cost-sensitive learning methods with application to imbalanced medical data //Informatics in Medicine Unlocked. – 2021. – Т. 25. – С. 100690.

78 Rather I. H., Kumar S., Gandomi A. H. Breaking the data barrier: a review of deep learning techniques for democratizing AI with small datasets //Artificial Intelligence Review. – 2024. – Т. 57. – №. 9. – С. 226.

79 Fareed S. et al. Fedsegnet: A federated learning framework for 3d medical image segmentation //International Journal of Ethical AI Application. – 2025. – Т. 1. – №. 2. – С. 30-46.

80 Meng D. et al. CmFNet: Cross-modal Fusion Network for Weakly-supervised Segmentation of Medical Images //arXiv preprint arXiv:2506.18042. – 2025.

81 Kozah N. et al. Data Augmentation Techniques for Medical Image Segmentation–A Review //2024 International Conference on Computer and Applications (ICCA). – IEEE, 2024. – С. 1-8.

82 Yi Z. et al. Dualgan: Unsupervised dual learning for image-to-image translation //Proceedings of the IEEE international conference on computer vision. – 2017. – С. 2849-2857.

83 Bai X. et al. Bridging the domain gap between synthetic and real-world data for autonomous driving //Journal on Autonomous Transportation Systems. – 2024. – Т. 1. – №. 2. – С. 1-15.

84 Kalra N., Paddock S. M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? //Transportation research part A: policy and practice. – 2016. – Т. 94. – С. 182-193.

85 Li Y. et al. Pre-training on synthetic driving data for trajectory prediction //2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – IEEE, 2024. – С. 5910-5917.

86 Ariai F., Mackenzie J., Demartini G. Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges //arXiv preprint arXiv:2410.21306. – 2024.

87 Hedderich M. A. et al. A survey on recent approaches for natural language processing in low-resource scenarios //arXiv preprint arXiv:2010.12309. – 2020.

88 Yang M. A survey on few-shot learning in natural language processing //2021 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA). – IEEE, 2021. – С. 294-297.

89 He Y. et al. Spatial-temporal super-resolution of satellite imagery via conditional pixel synthesis //Advances in Neural Information Processing Systems. – 2021. – Т. 34. – С. 27903-27915.

90 Bastani F. et al. Satlaspretrain: A large-scale dataset for remote sensing image understanding //Proceedings of the IEEE/CVF International Conference on Computer Vision. – 2023. – С. 16772-16782.

91 Xu M. et al. The eyes of the gods: A survey of unsupervised domain adaptation methods based on remote sensing data //Remote Sensing. – 2022. – Т. 14. – №. 17. – С. 4380.

92 Yang Z. et al. A systematic literature review of methods and datasets for anomaly-based network intrusion detection //Computers & Security. – 2022. – Т. 116. – С. 102675.

93 Saxena V. et al. What matters in learning from large-scale datasets for robot manipulation //arXiv preprint arXiv:2506.13536. – 2025.

94 Celemin C. et al. Interactive imitation learning in robotics: A survey //Foundations and Trends® in Robotics. – 2022. – Т. 10. – №. 1-2. – С. 1-197.

95 Abufadda M., Mansour K. A survey of synthetic data generation for machine learning //2021 22nd international arab conference on information technology (ACIT). – IEEE, 2021. – С. 1-7.

96 Alomar K., Aysel H. I., Cai X. Data augmentation in classification and segmentation: A survey and new strategies //Journal of Imaging. – 2023. – Т. 9. – №. 2. – С. 46.

97 Yang S. et al. Image data augmentation for deep learning: A survey //arXiv preprint arXiv:2204.08610. – 2022.

98 Pan S. J., Yang Q. A survey on transfer learning //IEEE Transactions on knowledge and data engineering. – 2009. – Т. 22. – №. 10. – С. 1345-1359.

99 Zu W. et al. Pre-trained Models Succeed in Medical Imaging with Representation Similarity Degradation //arXiv preprint arXiv:2503.07958. – 2025.

100Zeng W., Xiao Z. Few-shot learning based on deep learning: A survey //Mathematical Biosciences and Engineering. – 2024. – Т. 21. – №. 1. – С. 679-711.

101Alsentzer E. et al. Few shot learning for phenotype-driven diagnosis of patients with rare genetic diseases //npj Digital Medicine. – 2025. – Т. 8. – №. 1. – С. 380.

102 Mohamed A. et al. Self-supervised speech representation learning: A review //IEEE Journal of Selected Topics in Signal Processing. – 2022. – Т. 16. – №. 6. – С. 1179-1210.

103Alabadi M., Celik Y. Anomaly detection for cyber-security based on convolution neural network: A survey //2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). – IEEE, 2020. – С. 1-14..

104Saabith S., Thangarajah V., Fareez M. A survey of machine learning techniques for anomaly detection in cybersecurity //International Journal of Research in Engineering and Science. – 2023. – Т. 11. – №. 10. – С. 183-193.

105Yang S. et al. Image data augmentation for deep learning: A survey //arXiv preprint arXiv:2204.08610. – 2022.

106 Alzubaidi L. et al. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications //Journal of Big Data. – 2023. – Т. 10. – №. 1. – С. 46.

107Song Y. et al. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities //ACM Computing Surveys. – 2023. – Т. 55. – №. 13s. – С. 1-40.

108Song Y. et al. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities //ACM Computing Surveys. – 2023. – Т. 55. – №. 13s. – С. 1-40.

109Brown T. et al. Language models are few-shot learners //Advances in neural information processing systems. – 2020. – Т. 33. – С. 1877-1901.

110Kabashkin I. Integration of Foundation Models and Federated Learning in AIoT-Based Aircraft Health Monitoring Systems //Mathematics. – 2024. – Т. 12. – №. 21. – С. 3428.

111Lu Y. et al. Machine learning for synthetic data generation: a review //arXiv preprint arXiv:2302.04062. – 2023.

112 Fawagreh K., Gaber M. M., Elyan E. Random forests: from early developments to recent advancements //Systems Science & Control Engineering: An Open Access Journal. – 2014. – Т. 2. – №. 1. – С. 602-609.

113Boutaba R. et al. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities //Journal of Internet Services and Applications. – 2018. – Т. 9. – №. 1. – С. 1-99.

114Jin L. et al. Hand-crafted features or machine learnt features? together they improve RGB-D object recognition //2014 IEEE International Symposium on Multimedia. – IEEE, 2014. – С. 311-319.

115Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks //Advances in neural information processing systems. – 2012. – Т. 25.

116Aziz L. et al. Exploring deep learning-based architecture, strategies, applications and current trends in generic object detection: A comprehensive review //Ieee Access. – 2020. – Т. 8. – С. 170461-170495.

117 He K. et al. Deep residual learning for image recognition //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – С. 770-778.

118El Saj R. et al. Generative classifier pix2pix //Pattern Recognition and Tracking XXXIV. – SPIE, 2023. – Т. 12527. – С. 85-92.

119Parsai S., Ahmadi M. A low error face recognition system based on a new arrangement of convolutional neural network and data augmentation //TENCON 2022-2022 IEEE Region 10 Conference (TENCON). – IEEE, 2022. – С. 1-5.

120Scheunemann C. et al. Data Collection and Annotation Pipeline for Social Good Projects //AI4SG@ AAAI Fall Symposium. – 2020.

121 Chen Y. et al. Generative adversarial networks in medical image augmentation: a review //Computers in Biology and Medicine. – 2022. – Т. 144. – С. 105382.

122 Gharoun H. et al. Meta-learning approaches for few-shot learning: A survey of recent advances //ACM Computing Surveys. – 2024. – Т. 56. – №. 12. – С. 1-41.

123 Zhang J. et al. Federated learning with label distribution skew via logits calibration //International Conference on Machine Learning. – PMLR, 2022. – С. 26311-26329.

124 Tofighi G. et al. Hand posture recognition using K-NN and Support Vector Machine classifiers evaluated on our proposed HandReader dataset //2013 18th International Conference on Digital Signal Processing (DSP). – IEEE, 2013. – С. 1-5.

125 Lionnie R., Andika J., Alaydrus M. A New Approach to Recognize Faces Amidst Challenges: Fusion Between the Opposite Frequencies of the Multi-Resolution Features //Algorithms. – 2024. – Т. 17. – №. 11. – С. 529.

126 Wilson I. D. Recent advances from Olivetti research //IEE Colloquium on Interactive Multimedia: A Review and Update for Potential Users. – IET, 1992. – С. 7/1-7/4.

127 Bazatbekov B. et al. 2D face recognition using PCA and triplet similarity embedding //Bulletin of Electrical Engineering and Informatics. – 2023. – Т. 12. – №. 1. – С. 580-586.

128 Peprah F., Asante M. Comparative Analysis Of The Performance Of Principal Component Analysis (PCA) And Linear Discriminant Analysis (LDA) As Face Recognition Techniques //International Journal of Scientific & Technology Research. – 2017. – Т. 6. – №. 10. – С. 286-291.

129 Mahouachi D., Akhloufi M. A. Recent advances in infrared face analysis and recognition with deep learning //AI. – 2023. – Т. 4. – №. 1. – С. 199-233.

130 Bow S. T. Pattern recognition and image preprocessing. – CRC press, 2002.

131 Bindal N. et al. A systematic review of state-of-the-art noise removal techniques in digital images //Multimedia Tools and Applications. – 2022. – Т. 81. – №. 22. – С. 31529-31552.

132 Horwood A. et al. Image normalization: a basic requirement for computer-based automatic diagnostic applications. – 2001.

133 Hotelling H. Analysis of a complex of statistical variables into principal components //Journal of educational psychology. – 1933. – Т. 24. – №. 6. – С. 417.

134 Jolliffe I. T., Cadima J. Principal component analysis: a review and recent developments //Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences. – 2016. – Т. 374. – №. 2065. – С. 20150202.

135 Abiodun O. I. et al. Comprehensive review of artificial neural network applications to pattern recognition //IEEE access. – 2019. – Т. 7. – С. 158820-158846.

136 Tuffour P. A. Enhancing facial recognition: A comparative analysis of image preprocessing techniques in PCA, KL expansion and SVD. – 2023.

137 Aggarwal R. et al. Face recognition system using image enhancement with PCA and LDA //2022 6th International Conference on Computing Methodologies and Communication (ICCMC). – IEEE, 2022. – C. 1322-1327.

138 Tufail S. et al. Advancements and challenges in machine learning: A comprehensive review of models, libraries, applications, and algorithms //Electronics. – 2023. – T. 12. – №. 8. – C. 1789.

139 Aggarwal R. et al. Face recognition system using image enhancement with PCA and LDA //2022 6th International Conference on Computing Methodologies and Communication (ICCMC). – IEEE, 2022. – C. 1322-1327.

140 Kherif F., Latypova A. Principal component analysis //Machine learning. – Academic Press, 2020. – C. 209-225.

141 Nithya K., Rajamani V. Triplet Label Based Image Retrieval Using Deep Learning in Large Database //Computer Systems Science & Engineering. – 2023. – T. 44. – №. 3.

142 Ren Z. et al. Exploring simple triplet representation learning //Computational and Structural Biotechnology Journal. – 2024. – T. 23. – C. 1510-1521.

143 Wan Q., Zou Q. Learning metric features for writer-independent signature verification using dual triplet loss //2020 25th international conference on pattern recognition (ICPR). – IEEE, 2021. – C. 3853-3859.

144 Ren Z. et al. Exploring simple triplet representation learning //Computational and Structural Biotechnology Journal. – 2024. – T. 23. – C. 1510-1521.

145 Shi D., Orouskhani M., Orouskhani Y. A conditional Triplet loss for few-shot learning and its application to image co-segmentation //Neural Networks. – 2021. – T. 137. – C. 54-62.

146 Schroff F., Kalenichenko D., Philbin J. Facenet: A unified embedding for face recognition and clustering //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2015. – C. 815-823.

147 Wang H. et al. Cosface: Large margin cosine loss for deep face recognition //Proceedings of the IEEE conference on computer vision and pattern recognition. – 2018. – C. 5265-5274.

148 Ha M. L., Blanz V. Deep ranking with adaptive margin triplet loss //arXiv preprint arXiv:2107.06187. – 2021.

149 Tian Y. et al. Outlier-suppressed triplet loss with adaptive class-aware margins for facial expression recognition //2019 IEEE International Conference on Image Processing (ICIP). – IEEE, 2019. – C. 46-50.

150 Nguyen K., Nguyen H. H., Tiulpin A. Adaptive gradient triplet loss with automatic margin learning for forensic medical image matching //Medical Imaging with Deep Learning. – 2022.

151 Snell J., Swersky K., Zemel R. Prototypical networks for few-shot learning //Advances in neural information processing systems. – 2017. – T. 30.

152 Liu J. Feature dimensionality reduction for myoelectric pattern recognition: A comparison study of feature selection and feature projection methods //Medical engineering & physics. – 2014. – T. 36. – №. 12. – C. 1716-1720.
153 Turk M., Pentland A. Eigenfaces for recognition //Journal of cognitive neuroscience. – 1991. – T. 3. – №. 1. – C. 71-86.
154 Tofighi M. et al. Prior information guided regularized deep learning for cell nucleus detection //IEEE transactions on medical imaging. – 2019. – T. 38. – №. 9. – C. 2047-2058.
155 Hu X., Yang J., Yang J. A CNN-based approach for lung 3D-CT registration //IEEE Access. – 2020. – T. 8. – C. 192835-192843.
156 Li G., Zhang T., Leng F. The Research on Classification of Small Sample Data Set Image Based on Convolutional Neural Network //2021 33rd Chinese Control and Decision Conference (CCDC). – IEEE, 2021. – C. 2317-2324.

APPENDIX A: Code view

```
import os
import cv2
import numpy as np
import random
from itertools import product

import torch
import torch.nn as nn
import torch.optim as optim
from torch.utils.data import DataLoader, Dataset
from sklearn.decomposition import PCA

# ---------------------------
# Device & Reproducibility
# ---------------------------
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

def set_seed(seed=42):
```

```
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    if torch.cuda.is_available():
        torch.cuda.manual_seed(seed)
        torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False

set_seed(42)

# ---------------------------
# Paths & Settings
# ---------------------------
base_train_folder = "train20"
base_test_folder  = "test20"
img_size = (36, 36)

# ---------------------------
# Data Loading & Preprocessing
# ---------------------------
def load_dataset(folder_path, smoothing_factor):
    images, labels = [], []
    if not os.path.isdir(folder_path):
        raise FileNotFoundError(f"Folder not found: {folder_path}")

    person_folders = sorted(os.listdir(folder_path))
    for person_folder in person_folders:
        person_path = os.path.join(folder_path, person_folder)
        if not os.path.isdir(person_path):
            continue

        try:
            label = int(person_folder[1:]) - 1
        except:
            # Fallback: use folder index if naming not s#
            label = len(labels) // 10

        file_names = sorted(os.listdir(person_path))
        for img_name in file_names:
            img_path = os.path.join(person_path, img_name)
            img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
            if img is None:
                continue
            img = cv2.resize(img, img_size)

            # smoothing_factor is relative to image size
            sigma = float(smoothing_factor) * min(img_size)
            smoothed = cv2.GaussianBlur(img, (0, 0), sigmaX=sigma,
sigmaY=sigma)

            images.append(smoothed)
            labels.append(label)

    images = np.array(images, dtype=np.float32)
    labels = np.array(labels, dtype=np.int64)

    # Statistical normalization (zero-mean, unit-std) - image-level
    images = (images - images.mean()) / (images.std() + 1e-8)
    return images, labels
```

```python
def apply_pca(X, n_components):
    pca = PCA(n_components=n_components)
    X_pca = pca.fit_transform(X)
    return X_pca, pca


# ----------------------------
# Triplet Dataset (returns labels)
# ----------------------------
class ORLTripletDataset(Dataset):
    def __init__(self, features, labels):
        """
        features: numpy array [N, D] (PCA features)
        labels:   numpy array [N]
        """
        self.features = torch.tensor(features, dtype=torch.float32)
        self.labels   = torch.tensor(labels, dtype=torch.long)
        # Build index lists per class for fast triplet sampling
        self.label_dict = {}
        for idx, y in enumerate(self.labels.tolist()):
            self.label_dict.setdefault(y, []).append(idx)

    def __len__(self):
        return len(self.features)

    def __getitem__(self, idx):
        a = self.features[idx]
        ya = int(self.labels[idx])

        # Positive from same class
        pos_idx = random.choice(self.label_dict[ya])
        p = self.features[pos_idx]
        yp = ya

        # Negative from a different class
        neg_label_choices = [c for c in self.label_dict.keys() if c != ya]
        yn = int(random.choice(neg_label_choices))
        neg_idx = random.choice(self.label_dict[yn])
        n = self.features[neg_idx]

        return a, p, n, ya, yp, yn

# ----------------------------
# Linear Embedding Model
# ----------------------------
class PCA_Triplet(nn.Module):
    def __init__(self, input_dim, reduced_dim):
        super().__init__()
        # Linear projection W: [r, d]
        self.W = nn.Parameter(torch.randn(reduced_dim, input_dim) * 0.01)

    def forward(self, x):
        # x: [B, d]
        z = torch.matmul(x, self.W.t())  # [B, r]
        # L2-normalization (feature-level; required for cosine)
        z = z / (z.norm(dim=1, keepdim=True) + 1e-8)
        return z

# ----------------------------
# Centroid Bank (Prototypes)
# ----------------------------
class CentroidBank:
    def __init__(self, num_classes, emb_dim, device, momentum=0.9):
```

```
        self.m = momentum
        self.device = device
        self.centroids = torch.zeros(num_classes, emb_dim, device=device)
        self.initialized = torch.zeros(num_classes, dtype=torch.bool,
device=device)

    @torch.no_grad()
    def update(self, z, y):
        # z: [B, r] (assumed unit-norm); y: [B]
        for c in y.unique():
            mask = (y == c)
            if mask.any():
                mean_c = z[mask].mean(dim=0)
                mean_c = mean_c / (mean_c.norm() + 1e-8)
                cidx = int(c.item())
                if not self.initialized[cidx]:
                    self.centroids[cidx] = mean_c
                    self.initialized[cidx] = True
                else:
                    self.centroids[cidx] = self.m * self.centroids[cidx] + (1 -
self.m) * mean_c
                    self.centroids[cidx] = self.centroids[cidx] /
(self.centroids[cidx].norm() + 1e-8)

    def get(self, y):
        # Returns centroids for labels y: [B, r]
        return self.centroids[y]

# ----------------------------
# Loss: Cosine Triplet + Adaptive Margin + Projection-Consistency
# ----------------------------
def cosine_triplet_loss_with_adapt_and_proj(a, p, n, ya, yp, yn, centroids,
alpha0, beta, lam):
    """
    a, p, n: [B, r] (L2-normalized embeddings)
    ya, yp, yn: [B] labels
    centroids: CentroidBank
    alpha0: base margin
    beta:   adaptive margin strength
    lam:    projection-consistency weight
    """
    # Cosine similarities
    pos_sim = (a * p).sum(dim=1)  # sim(a,p)
    neg_sim = (a * n).sum(dim=1)  # sim(a,n)

    # Adaptive margin per sample: alpha_i = alpha0 + beta*(1 - pos_sim)
    alpha_i = alpha0 + beta * (1.0 - pos_sim)
    triplet = torch.clamp(neg_sim - pos_sim + alpha_i, min=0.0)

    # Projection-consistency (anchors & positives to their class directions)
    ca = centroids.get(ya)  # [B, r]
    cp = centroids.get(yp)  # [B, r]
    # Fallback if centroid is zero (uninitialized): use current embedding
    ca = torch.where(ca.norm(dim=1, keepdim=True) > 0, ca, a.detach())
    cp = torch.where(cp.norm(dim=1, keepdim=True) > 0, cp, p.detach())

    Pa = (a * ca).sum(dim=1, keepdim=True) * ca  # projection onto class
direction
    Pp = (p * cp).sum(dim=1, keepdim=True) * cp
    proj_term = 0.5 * ((a - Pa).pow(2).sum(dim=1) + (p - Pp).pow(2).sum(dim=1))

    return triplet.mean() + lam * proj_term.mean()
```

```
# ----------------------------
# Training (Proposed)
# ----------------------------
def train_pca_triplet_cosine_adapt_proj(train_loader, input_dim, reduced_dim,
                                        alpha0, beta, lam, epochs, lr,
num_classes):
   model = PCA_Triplet(input_dim, reduced_dim).to(device)
   optimizer = optim.Adam(model.parameters(), lr=lr)
   bank = CentroidBank(num_classes=num_classes, emb_dim=reduced_dim,
device=device, momentum=0.9)

   model.train()
   for epoch in range(epochs):
       for a, p, n, ya, yp, yn in train_loader:
           a = a.to(device); p = p.to(device); n = n.to(device)
           ya = ya.to(device); yp = yp.to(device); yn = yn.to(device)

           optimizer.zero_grad()
           a_emb = model(a)
           p_emb = model(p)
           n_emb = model(n)

           loss = cosine_triplet_loss_with_adapt_and_proj(
               a_emb, p_emb, n_emb, ya, yp, yn, bank, alpha0=alpha0,
beta=beta, lam=lam
           )
           loss.backward()
           optimizer.step()

           # Update centroids using current (post-update) embeddings
           with torch.no_grad():
               bank.update(a_emb, ya)
               bank.update(p_emb, yp)

   return model

# ----------------------------
# Evaluation: Cosine 1-NN
# ----------------------------
def projection_classification_cosine_nn(model, X_train, X_test, y_train,
y_test):
   # X_*: torch.FloatTensor [N, d_pca]
   model.eval()
   with torch.no_grad():
       Z_train = model(X_train.to(device))  # [N_train, r], unit-norm already
       Z_train = Z_train / (Z_train.norm(dim=1, keepdim=True) + 1e-8)
       Z_test  = model(X_test.to(device))
       Z_test  = Z_test / (Z_test.norm(dim=1, keepdim=True) + 1e-8)

       correct = 0
       for i in range(Z_test.shape[0]):
           z = Z_test[i]                        # [r]
           sims = torch.matmul(Z_train, z)      # cosine sims to all train
items
           pred = y_train[torch.argmax(sims).item()]
           if pred == y_test[i].item():
               correct += 1

   return correct / len(y_test)

# ----------------------------
# Grid Search
# ----------------------------
```

```
if __name__ == "__main__":
   # Hyperparameter grids (feel free to adjust)
   pca_components   = [45, 55, 60, 70]
   embedding_dims   = [64, 80, 96]
   alpha_values     = [0.3, 0.4, 0.5]        # base margin alpha0
   learning_rates   = [0.0002, 0.0003, 0.0005]
   epochs_list      = [18, 22, 25]
   smoothing_values = [0.015, 0.02]

   # Proposed-method params
   betas            = [0.1, 0.3]             # adaptive margin strength
   lambdas          = [0.05, 0.1, 0.2]       # projection-consistency weight

   best_accuracy = 0.0
   best_params = {}
   results = []

   for s_factor, pca_comp, emb_dim, alpha0, lr, epochs, beta, lam in product(
       smoothing_values, pca_components, embedding_dims, alpha_values,
learning_rates, epochs_list, betas, lambdas
   ):
       print(f"\nTesting: smoothing={s_factor}, PCA={pca_comp},
Embedding={emb_dim}, "
             f"alpha0={alpha0}, beta={beta}, lambda={lam}, LR={lr},
Epochs={epochs}")

       # Load data with current smoothing
       X_train_img, y_train = load_dataset(base_train_folder, s_factor)
       X_test_img,  y_test  = load_dataset(base_test_folder,  s_factor)

       # Vectorize
       X_train_vec = X_train_img.reshape(X_train_img.shape[0], -1)
       X_test_vec  = X_test_img.reshape(X_test_img.shape[0],  -1)

       # PCA
       X_train_pca, pca_model = apply_pca(X_train_vec, n_components=pca_comp)
       X_test_pca  = pca_model.transform(X_test_vec)

       # Tensors
       X_train_pca_t = torch.tensor(X_train_pca, dtype=torch.float32)
       X_test_pca_t  = torch.tensor(X_test_pca,  dtype=torch.float32)
       y_train_t     = torch.tensor(y_train, dtype=torch.long)
       y_test_t      = torch.tensor(y_test,  dtype=torch.long)

       # Dataloader for triplets
       train_dataset = ORLTripletDataset(X_train_pca, y_train)
       train_loader  = DataLoader(train_dataset, batch_size=16, shuffle=True,
drop_last=True)

       num_classes = len(np.unique(y_train))

       # Train proposed model
       model = train_pca_triplet_cosine_adapt_proj(
           train_loader,
           input_dim=X_train_pca_t.shape[1],
           reduced_dim=emb_dim,
           alpha0=alpha0,
           beta=beta,
           lam=lam,
           epochs=epochs,
           lr=lr,
           num_classes=num_classes
       )
```

```
        # Evaluate with cosine 1-NN
        acc = projection_classification_cosine_nn(
            model, X_train_pca_t, X_test_pca_t, y_train_t, y_test_t
        )
        print(f"Accuracy: {acc:.4f}")

        results.append({
            "smoothing": s_factor, "PCA": pca_comp, "Embedding": emb_dim,
            "alpha0": alpha0, "beta": beta, "lambda": lam, "LR": lr, "Epochs":
epochs,
            "Accuracy": acc
        })

        if acc > best_accuracy:
            best_accuracy = acc
            best_params = results[-1]

    print("\n========== Best Configuration (Proposed: PCA-TP-AMPR) ==========")
    for k, v in best_params.items():
        if k != "Accuracy":
            print(f"{k}: {v}")
    print(f"Best Accuracy: {best_accuracy:.4f}")
```