

Article

Face Recognition System Using CLIP and FAISS for Scalable and Real-Time Identification

Antonio Labinjan , Sandi Baressi Šegota *, Ivan Lorencin  and Nikola Tanković *

Faculty of Informatics, Juraj Dobrila University of Pula, Alda Negrija 6, 51000 Pula, Croatia;
antonio.labinjan@unipu.hr (A.L.); ivan.lorencin@unipu.hr (I.L.)

* Correspondence: sandi.baressi.segota@unipu.hr or sandi.baressi.ssegota@unipu.hr (S.B.Š.);
nikola.tankovic@unipu.hr (N.T.)

Abstract

Face recognition is increasingly being adopted in industries such as education, security, and personalized services. This research introduces a face recognition system that leverages the embedding capabilities of the CLIP model. The model is trained on multimodal data, such as images and text and it generates high-dimensional features, which are then stored in a vector index for further queries. The system is designed to facilitate accurate real-time identification, with potential applications in areas such as attendance tracking and security screening. Specific use cases include event check-ins, implementation of advanced security systems, and more. The process involves encoding known faces into high-dimensional vectors, indexing them using a vector index FAISS, and comparing them to unknown images based on L2 (euclidean) distance. Experimental results demonstrate a high accuracy that exceeds 90% and prove efficient scalability and good performance efficiency even in datasets with a high volume of entries. Notably, the system exhibits superior computational efficiency compared to traditional deep convolutional neural networks (CNNs), significantly reducing CPU load and memory consumption while maintaining competitive inference speeds. In the first iteration of experiments, the system achieved over 90% accuracy on live video feeds where each identity had a single reference video for both training and validation; however, when tested on a more challenging dataset with many low-quality classes, accuracy dropped to approximately 73%, highlighting the impact of dataset quality and variability on performance.

Keywords: attendance tracking; CLIP; embeddings; face recognition; FAISS; real-time identification; security screening; vector search

1. Introduction

In today's digital age, facial recognition technologies play a key role in transforming interactions with computer systems, securing spaces, and optimizing everyday processes [1]. Facial recognition is one of the fastest growing fields in biometric authentication [2,3], enabling fast and reliable identification of individuals without the need for physical contact or additional interaction [4]. With the increasing availability of advanced computational resources [5,6] and sophisticated algorithms, implementing facial recognition systems has become an indispensable tool across various industries [4,7].

The applications of this technology are diverse and span multiple sectors. In education, facial recognition can be used for automated student attendance tracking, allowing for more efficient monitoring and record-keeping. In the security sector, this technology enables the



Received: 14 January 2026

Revised: 4 February 2026

Accepted: 24 February 2026

Published: 1 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

identification of suspicious individuals and enhances surveillance in public and private spaces, significantly contributing to crime prevention and increased safety. Additionally, in personalized services, facial recognition helps tailor user experiences by providing access to relevant information and services based on the user's identity [8].

Traditional face recognition models, such as FaceNet [5], DeepFace [6], VGG-Face [9], and ArcFace [10], achieve very high accuracy on standard benchmarks (often exceeding 97–99%), but typically require large-scale training, fine-tuning, and substantial computational resources, which can limit their scalability and real-time deployment. YOLO-Face [11,12] offers a fast, all-in-one solution, but its performance can degrade on more complex or occluded faces.

The goal of this research is to develop and evaluate an efficient facial recognition system that leverages the capabilities of the CLIP (Contrastive Language-Image Pre-Training) model to extract high-dimensional features from images and video recordings, minimizing the need for extensive fine-tuning [13]. Combined with FAISS (Facebook AI Similarity Search) vector search [14–18], the system supports fast and accurate real-time facial identification. While its accuracy ranges from ~87–100% depending on dataset characteristics, the approach is fast, scalable, and requires minimal training, making it suitable for real-time applications such as attendance tracking and security screening.

Through experimental evaluation, this research demonstrates how the combination of advanced machine learning techniques and optimized search algorithms can improve both the accuracy [13,19] and deployment efficiency of facial recognition systems. Despite slightly lower peak accuracy compared to fully-trained CNN models, the proposed approach offers a better trade-off between performance, real-time responsiveness, and adaptability in realistic application scenarios.

The main contributions of this work can be summarized as follows:

- We propose a novel face recognition pipeline that leverages a pre-trained CLIP model for facial embedding extraction combined with FAISS-based similarity search, demonstrating that multimodal vision-language models can be effectively repurposed for high-performance face recognition without task-specific fine-tuning.
- We present, to the best of our knowledge, the first systematic evaluation of the CLIP-FAISS combination for face recognition in both controlled and unconstrained video-based scenarios, achieving recognition accuracy exceeding 90% on live video feeds and maintaining robust performance across challenging datasets.
- We introduce a deterministic, threshold-based voting mechanism with configurable parameters (k_1 , k_2 , and distance threshold), enabling transparent, reproducible, and interpretable decision-making in the embedding space.
- We provide a comprehensive resource utilization analysis, demonstrating that the proposed zero-shot architecture achieves superior computational efficiency—reducing CPU load by nearly 50% and significantly lowering memory overhead compared to traditional deep CNN-based frameworks, thus facilitating scalable deployment on consumer-grade hardware.
- We establish a standardized benchmarking protocol on the YouTube Faces (YTF) dataset, providing a fair comparative analysis that highlights the trade-off between the extreme flexibility of vision-language embeddings and the peak accuracy of identity-specific supervised models.
- We conduct an extensive grid search analysis to investigate the impact of neighborhood size and distance thresholds on recognition performance, providing practical insights into cost-effective and scalable deployment under limited computational resources.

- We perform a detailed dataset quality analysis, identifying and excluding extremely low-quality classes, and demonstrate how dataset characteristics such as resolution, blur, and intra-class variability directly affect embedding-based face recognition performance.

Despite the strong performance of conventional face recognition models, most existing approaches are fundamentally constrained by their reliance on large-scale, task-specific training datasets and rigid class definitions. These limitations significantly reduce their adaptability in real-world scenarios where new identities appear frequently, labeled data is scarce, and deployment conditions vary over time. As a result, high benchmark accuracy does not necessarily translate into practical robustness or scalability in live systems. In contrast, CLIP-based representations introduce a fundamentally different paradigm for face recognition. Rather than optimizing embeddings exclusively for closed-set identity classification, CLIP learns a semantically structured, high-dimensional embedding space through large-scale contrastive vision–language pre-training. This property enables strong zero-shot and few-shot recognition behavior, allowing new identities to be incorporated without retraining or fine-tuning the underlying model. From a recognition perspective, this leads to improved generalization across unseen subjects, varying illumination conditions, and heterogeneous data sources. The key recognition effect exploited in this work is the stability and semantic consistency of CLIP embeddings under domain shifts. While traditional CNN-based face recognition models tend to overfit to dataset-specific distributions, CLIP embeddings preserve inter-class separability even when facial images differ in resolution, pose, blur, or capture device. This makes CLIP particularly well suited for video-based and real-time recognition scenarios, where input quality is inherently inconsistent. Furthermore, by decoupling feature extraction from identity classification and pairing CLIP embeddings with FAISS-based approximate nearest neighbor search, the proposed system enables scalable open-set face recognition. New identities can be added dynamically by inserting embeddings into the index, eliminating the need for costly retraining cycles. This architectural separation represents a significant practical and conceptual shift compared to end-to-end trained face recognition pipelines.

The remainder of this paper is organized as follows. Methodology section details the proposed methodology, focusing on the integration of the CLIP model for high-dimensional feature extraction and the FAISS index for efficient similarity search. Results section presents the experimental results and provides an analysis of system performance across different datasets. Discussion section discusses the practical implications of the results, addresses testing limitations, and explores potential real-life applications and live-feed implementation. Finally, conclusion section concludes the paper by summarizing the key findings and suggesting directions for future research

2. Related Work

FAISS (Facebook AI Similarity Search) [14,18] and CLIP (Contrastive Language-Image Pretraining) [20–22] represent two advanced approaches in the fields of computer vision [23] and natural language processing [24] for data processing and retrieval. FAISS is a highly efficient system for vector similarity search [17], optimized to handle large databases. Its applications include image retrieval based on visual similarity, biometric facial recognition systems, recommendation systems, and semantic search for textual data and genetic sequences. It is particularly useful in IoT data analysis and anomaly detection [25], enabling fast comparisons of time series [26] data and sensor patterns. Furthermore, FAISS is utilized in accelerated document search systems, user behavior analysis, and real-time data classification, significantly improving the scalability and efficiency of machine learning models.

On the other hand, CLIP is a transformer-based model trained on a large dataset combining images and descriptive texts, allowing it to understand visual content in the context of natural language. Its applications include image retrieval based on textual queries, automatic image captioning, zero-shot classification [27–29], and improving generative image models such as DALL-E [30] and Stable Diffusion [31]. More recently, CLIP-based zero-shot learning has been successfully applied in domain-specific and safety-critical scenarios, such as maritime object recognition and monitoring, demonstrating strong performance even with minimal labeled data and highlighting the model’s potential for real-world deployment in dynamic environments [32]. CLIP is also used for content moderation, video analysis, autonomous system interactions with the environment, and in creative industries for generating artistic visuals based on semantic descriptions.

Although FAISS and CLIP have been applied individually in various domains, no implementation has been recorded that combines them for scalable real-time facial recognition. This work introduces the first such approach, leveraging CLIP’s capabilities to generate robust facial representations [10] and FAISS’s efficiency for extremely fast searches within large databases [15]. The integration of these two models enables the development of advanced multimodal data processing systems, providing more efficient search and classification methods across a wide range of applications, from security systems to creative industries. Proposed implementation not only enhances the speed and accuracy of facial identification, but also unlocks new possibilities in the field of computer vision that have not yet been explored.

3. Proposed Methodology

The described system is implemented in Python version 3.12 using CLIP for embedding extraction from images and FAISS for fast similarity search. Video recordings of human faces are processed into individual frames prior to embedding extraction. In the initial experimental setup, frames extracted from the same video recording were used for both training and validation. While this configuration allowed for a preliminary assessment of system performance, it represents an optimistic scenario due to the strong visual similarity between samples originating from the same video stream.

To obtain more realistic and robust evaluation results, a second dataset configuration was adopted in which each class (individual) was represented by two entirely separate video recordings [33], one used exclusively for training and the other for validation. This setup better simulates real-world deployment scenarios, where faces are observed under varying conditions across different recordings. Figure 1 shows a schematic representation of the system highlighting all the key steps of facial recognition, from the initial image input to the final classification using FAISS. The system first loads known face images using the `load_known_faces()` function, which iterates through directories labeled with class names. Each class contains a subdirectory named `train`, where images used for embedding extraction are stored. For each image in the training directory, the function `add_known_face()` is invoked to extract image embeddings using the CLIP model. Specifically, the `openai/clip-vit-base-patch32` variant was employed. Images are pre-processed using the corresponding `CLIPProcessor`, including resizing and normalization to the model’s required input resolution of 224×224 pixels. The resulting embeddings are 512-dimensional vectors and are computed on a GPU when available. Batch processing is used to improve computational efficiency, with the batch size determined by available GPU memory. Each embedding is associated with its corresponding class, forming a database of known identities used during the recognition phase.

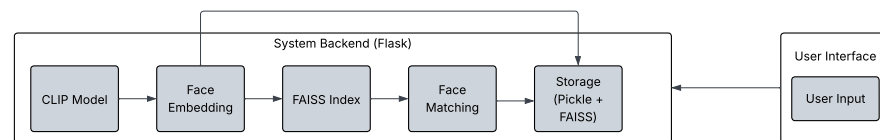


Figure 1. Schema overview.

After extracting embeddings, the computed vectors are organized into a FAISS index using the `build_index` function. FAISS enables fast vector search in high-dimensional space using the L2 distance, also known as Euclidean distance [34,35]. The Euclidean (ℓ_2) distance between two vectors $\mathbf{A} = (a_1, a_2, \dots, a_n)$ and $\mathbf{B} = (b_1, b_2, \dots, b_n)$ is defined as

$$d(\mathbf{A}, \mathbf{B}) = \|\mathbf{A} - \mathbf{B}\|_2 = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (1)$$

This metric measures the physical distance between two points in an n -dimensional space and is used to compare face embeddings. The rationale for choosing the Euclidean distance is summarized in Table 1. Figure 2 illustrates a system flowchart depicting all crucial steps required for successful face classification. In the proposed system, FAISS is employed to store all known face embeddings within an `IndexFlatL2` index, enabling efficient similarity search based on L2 distance [36–38]. When a new embedding is added, it is directly inserted into the index together with its corresponding class label. During recognition, the embedding of the input face image is compared against the stored embeddings, and the system retrieves the k_2 nearest neighbors with the smallest L2 distances

Table 1. Rationale for using Euclidean (L2) distance in the proposed facial recognition system.

Reason	Description
Native FAISS support	The <code>IndexFlatL2</code> index natively implements L2 distance, allowing extremely efficient similarity search without custom metric overhead.
Geometrical interpretability	Euclidean distance corresponds to the “straight-line” distance between two points in the feature space, making it intuitive for measuring similarity of embeddings.
Consistency with feature extraction	The embedding vectors produced by the chosen feature extractor are normalized in a way that L2 distance accurately reflects similarity; in many CV pipelines, L2 distance correlates well with cosine similarity after normalization.
Computational efficiency	Compared to other metrics like Mahalanobis distance or learned metrics, L2 distance is simple to compute and scales efficiently to large datasets, which is crucial for real-time recognition.

A similarity threshold is applied directly to the L2 distances to filter out embeddings that are not sufficiently similar. Only neighbors whose distance is less than or equal to the defined threshold are considered valid candidates for classification. Among these candidates, a voting mechanism is applied using the first k_1 nearest neighbors, where the predicted class is determined by the most frequently occurring label [39–41]. If none of the top k_1 neighbors satisfy the threshold, additional neighbors in the range from k_1 to k_2 are evaluated as a fallback to improve robustness. If no neighbor meets the threshold criterion, the face is deterministically classified as Unknown, which occurs when the embedding does not sufficiently match any known face in the database. This threshold-based filtering represents the decisive criterion in the classification pipeline, while the parameters k_1 and k_2 control the size of the primary and backup voting sets. The procedure is fully deterministic and reproducible: given the same embeddings and threshold value, identical predictions

are always produced. While the threshold remains a fixed and stable value during real-time (live) inference to ensure consistent recognition, it was dynamically adjusted during the grid search phase to identify the optimal cutoff. Specifically, all combinations of k_1 , k_2 , and threshold values were systematically evaluated, ensuring transparency and interpretability of the reported results. The threshold thus represents an explicit cutoff in the embedding space rather than an arbitrary heuristic, and can be readily transferred to new datasets. Described voting process is visualized on Figure 3 which depicts the process of finding k_1 and k_2 nearest matches.

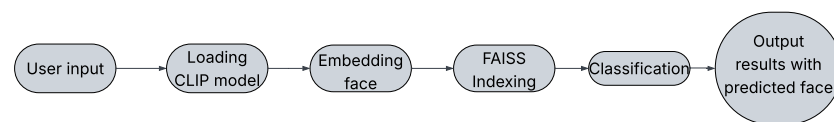


Figure 2. System flowchart.

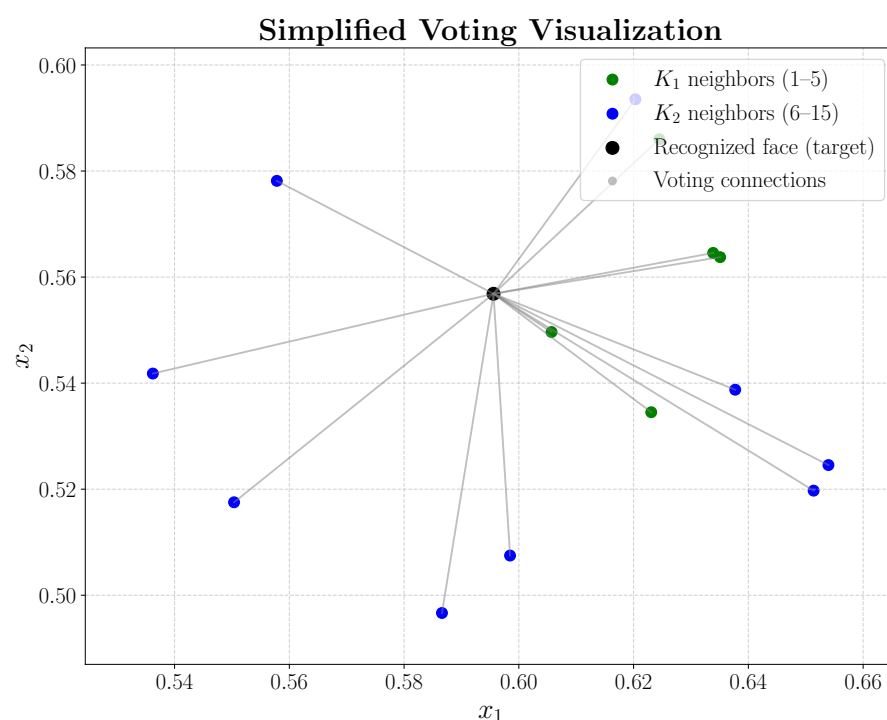


Figure 3. Simplified voting visualization.

First, the system was tested on a dataset of approximately 500 classes, where each person had one video, using some frames for training and the rest for validation (80–20 split). In this case, the results showed 100% accuracy. This high performance was attributable to the minimal difference between the training and validation images, as both were derived from the same recording and context, failing to test the model's ability to generalize to new environments. To further enhance the realism of our evaluation and ensure robust performance assessment, we utilized the YTFaces dataset [42], which comprises videos of individuals captured in unconstrained and diverse environments (Figures 4 and 5). This dataset effectively simulates the type of input the system would encounter in a live camera setting, including variations in pose, lighting, and background.

As described in Table 2, the dataset, in total, includes 3425 videos representing 1595 unique individuals, with most subjects having two or more videos available. Video lengths vary substantially, ranging from very short clips of under 50 frames to longer sequences exceeding 6000 frames, with an average duration of around 180 frames per clip. For evaluation, two distinct videos per person were selected to ensure that frames used for training

and validation came from completely separate recordings, providing a realistic scenario for testing the model's ability to recognize faces across different conditions.

Table 2. Summary of the YTFaces Dataset Used for Evaluation.

Property	Value
Total number of videos	3425
Number of unique individuals	1595
Average videos per individual	2.15
Minimum video length	48 frames
Maximum video length	6070 frames
Average video length	181 frames
Training-validation split	Two separate videos per individual
Evaluation protocol	No frame overlap between training and validation



Figure 4. Training dataset image.



Figure 5. Validation dataset image.

3.1. FAISS Implementation

Face embeddings were visualized in a two-dimensional space using Principal Component Analysis (PCA) [43,44]. PCA reduces dimensionality by projecting high-dimensional data onto a smaller set of principal components while preserving as much variance as possible. This visualization provides qualitative insight into the relative distribution of embeddings and their mutual relationships within the CLIP embedding space. Figure 6 illustrates a realistic arrangement of identity clusters in two-dimensional space, where partial overlap between different classes can be observed.

Such overlap is particularly noticeable for identities with similar facial features, illumination conditions, or expressions, and becomes more pronounced as the number of classes and intra-class samples increases [35,45]. While this visual overlap suggests potential ambiguity in low-dimensional projections, it does not directly reflect the behavior of the recognition system operating in the original embedding space.

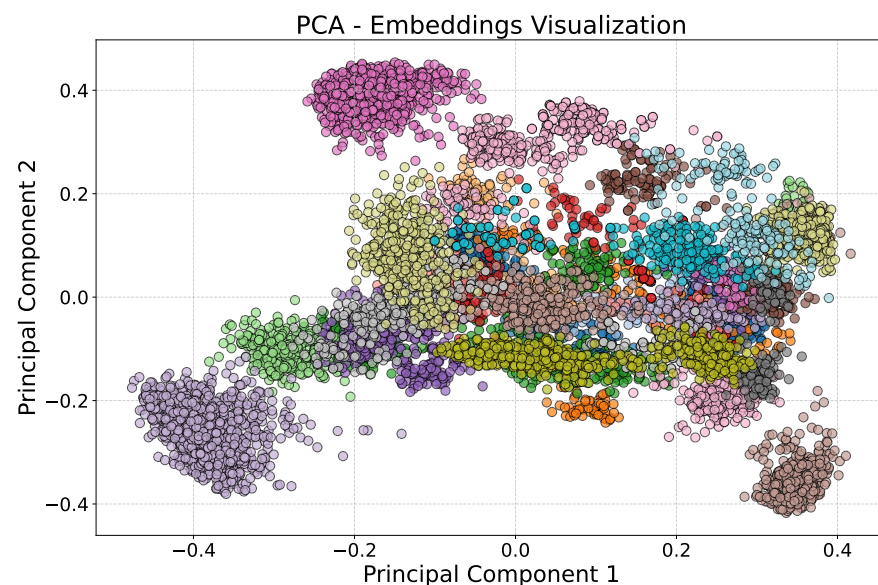


Figure 6. Realistic depiction of identity clusters and their overlap in a two-dimensional PCA projection, with different colors depicting different clusters.

While the PCA visualization in Figure 6 provides an intuitive low-dimensional illustration of the embedding distribution, it does not reflect the actual recognition process. The proposed system operates exclusively in the original 512-dimensional CLIP embedding space, where all similarity computations and classification decisions are performed. Dimensionality reduction is used solely for visualization purposes and has no impact on recognition accuracy. At the core of the recognition pipeline, FAISS indexes all enrolled facial embeddings using the IndexFlatL2 structure and performs exact nearest neighbor search based on the L2 distance metric. Although this approach computes exact distances, the use of FAISS provides highly optimized vectorized implementations that significantly accelerate similarity search compared to naive exhaustive comparisons. This ensures deterministic behavior and reproducible results, which is particularly important for scientific evaluation and controlled experimental analysis. For each query embedding, FAISS retrieves the top- k nearest neighbors, effectively constraining the decision process to a compact local region of the embedding space. By limiting classification to the most relevant neighbors, the system reduces unnecessary distance evaluations at the decision level and improves robustness against global embedding density variations. A distance-based threshold is subsequently applied to remove weak or ambiguous matches, ensuring that only sufficiently similar embeddings influence the final classification outcome. To

address class overlap and improve robustness in visually challenging scenarios, the system employs a deterministic majority voting scheme over the remaining nearest neighbors. Even in cases where embeddings from different identities partially overlap in the high-dimensional space, consistent local clustering within the k -nearest neighborhood enables reliable discrimination. Compared to single-nearest-neighbor classification, this strategy reduces sensitivity to noise, outliers, and borderline samples, resulting in more stable and interpretable recognition decisions. From a rationality and optimization perspective, the use of IndexFlatL2 represents a deliberate design choice that prioritizes accuracy, transparency, and experimental reproducibility over approximate acceleration techniques. By decoupling feature extraction from classification and restricting similarity evaluation to a localized neighborhood defined by top- k retrieval and distance thresholding, the proposed approach effectively mitigates embedding overlap, accelerates inference at the decision stage, and preserves recognition accuracy in realistic, high-variability environments.

3.2. Evaluation Process

After defining the dataset, an extensive evaluation procedure was conducted to test different combinations of the parameters k_1 , k_2 , and the similarity threshold. A grid search strategy [46–48] was employed to systematically explore the parameter space in a transparent and reproducible manner. The following parameter ranges were defined: $k_1 \in [1, 9]$, $k_2 \in [2, 10]$, and threshold values in the range $[0.5, 1.0]$ with a step size of 0.01.

The selected ranges for k_1 and k_2 were motivated by the nature of nearest-neighbor-based face recognition in high-dimensional embedding spaces. Since the classification decision relies on the closest embeddings in the FAISS index, restricting the search to a relatively small number of nearest neighbors ensures that only semantically relevant and spatially close embeddings are considered. Larger values of k would include increasingly distant neighbors, which are less informative and may introduce noise into the voting process, potentially degrading classification performance. Furthermore, limiting the ranges of k_1 and k_2 significantly reduces the computational cost of the grid search, making the evaluation feasible under constrained computational resources. This design choice enables a cost-effective and scalable evaluation strategy while preserving the interpretability and relevance of the tested configurations. Combinations where $k_1 = k_2$ or $k_2 < k_1$ were excluded, as such configurations do not provide meaningful additional information for the primary and secondary voting mechanism.

During evaluation, each image from the validation set was forwarded through the CLIP model to extract embeddings, which were subsequently normalized using the ℓ_2 norm (`linalg.norm`) [49] to facilitate consistent distance-based comparisons.

For each embedding from the validation folder, a class prediction, i.e., face recognition, was performed. The true label and predicted label values were also saved for each image, which defines the actual class to which the image belongs and the class predicted by the model. If they are the same, it is a correct classification. If they differ, it indicates an error in prediction. The prediction is made by taking each validation embedding and comparing it with the known embeddings extracted from the training images, finding the top- k most similar embedding (with the smallest distance). Once the system determines which class the most of top- k similar embeddings belong to, the validation embedding is assigned to that same class and recorded as a prediction, with an automatic accuracy check.

The threshold parameter was tested in the range of 0.5 to 1.0. Given the use of L2 distance, a lower threshold imposes a stricter criterion for identification, demanding closer proximity between the query and known face embeddings. The minimum value of 0.5 was selected to optimize the balance between minimizing false negatives and maintaining a viable acceptance rate during the initial classification phase. Conversely, increasing the

threshold subsequently enlarges the acceptable search volume, leading to a corresponding increase in true positive identifications.

For rigorous and reproducible evaluation, the results of each grid search iteration were systematically recorded in the file `grid_search_results`. This logging captured the configuration parameters and the full set of classification metrics, specifically: k_1 , k_2 , `threshold`, `precision`, `recall`, `f1`, `tp`, `fp`, `tn`, `fn`. [41]. The complete definitions for all parameters and metrics employed are detailed in Table 1.

The grid search process generated a distinct row of data for each iteration, documenting the k_1 , k_2 , and `threshold` parameters alongside the resulting evaluation metrics. A typical output row is shown in Table 3.

Table 3. Sample grid search output for one parameter combination.

k_1	k_2	Threshold	Precision	Recall	F1	TP	FP	TN	FN
1	2	1.0	0.7305	0.7305	0.677	2702	991	46,577	991

Each parameter combination produced a comprehensive set of evaluation metrics - precision, recall, F1-score, and the absolute counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). This tabular data provided a detailed understanding of how changes in hyperparameters affected classification performance, thereby enabling the selection of the optimal configuration that achieved the desired balance across output metrics. For example, the configuration $k_1 = 1$, $k_2 = 2$, and `threshold` = 1.0 resulted in `precision` = 0.7305 and `recall` = 0.7305, demonstrating the corresponding number of correct and incorrect classifications ($TP = 2702$, $FP = 991$, $TN = 46577$, $FN = 991$). Analyzing numerous results of this type was essential for assessing the ideal trade-offs necessary for optimal classification efficiency.

3.3. Effects of Augmentation

During the experimental phase, data augmentation [50] was applied to the training set using techniques such as rotation, mirroring, cropping, and blurring. However, these methods significantly degraded the model's performance by introducing unnatural distortions that even humans found difficult to recognize. Excessive geometric augmentation alters the fundamental biometric structure, effectively transforming the image into a representation of a person the system has not been "trained" on. Such augmentations do not reflect real-world conditions, where only slight variations in lighting or angle occur—changes the system can handle reliably. More realistic variations, such as hairstyle changes, facial hair, or makeup, have minimal impact on embeddings and do not impair recognition.

4. Results

Monitoring the system's performance across different dataset sizes revealed a critical observation: as anticipated, the metrics generally decreased as the number of classes and images increased. The performance metrics employed in this study are detailed in Table 4. For datasets containing fewer than 25 classes, performance was near-perfect because the limited number of embeddings ensured ample separation in the embedding space, thus preventing misclassifications. However, with an increased quantity of images, some minor misclassifications emerged due to rising embedding density. Table 5 presents the detailed evaluation metrics for 25 classes, including precision, recall, F1-score, and support parameters for each class.

Table 4. Explanation of Classification Metrics.

Term	Explanation
TP	True positive predictions, i.e., correctly predicted positive samples
FP	False positive predictions, i.e., incorrectly predicted positive samples
TN	True negative predictions, i.e., correctly predicted negative samples
FN	False negative predictions, i.e., incorrectly predicted negative samples
Precision	How many of all predicted positive samples are actually positive
Recall	How many positive samples are correctly predicted as positive
F1-score	The harmonic mean of precision and recall (their balance)

Table 5. Class-wise evaluation metrics reported across 25 classes.

Class	Precision	Recall	F1-Score	Support
1	1.0000	1.0000	1.0000	57
2	1.0000	1.0000	1.0000	67
3	0.9662	1.0000	0.9828	257
4	1.0000	1.0000	1.0000	249
5	0.7037	1.0000	0.8261	57
6	1.0000	1.0000	1.0000	102
7	1.0000	1.0000	1.0000	41
8	1.0000	0.6712	0.8033	73
9	0.7273	1.0000	0.8421	56
10	1.0000	1.0000	1.0000	16
11	1.0000	1.0000	1.0000	38
12	0.9494	0.9740	0.9615	77
13	1.0000	1.0000	1.0000	7
14	0.9608	1.0000	0.9800	49
15	0.8333	1.0000	0.9091	25
16	1.0000	1.0000	1.0000	13
17	1.0000	1.0000	1.0000	18
18	0.8788	1.0000	0.9355	58
19	0.8621	1.0000	0.9259	25
20	1.0000	1.0000	1.0000	98
21	1.0000	1.0000	1.0000	199
22	1.0000	1.0000	1.0000	17
23	0.0000	0.0000	0.0000	49
24	1.0000	0.9649	0.9821	114
25	1.0000	0.9592	0.9792	49
Unknown	0.0000	0.0000	0.0000	0

4.1. Class Results and Handling of Problematic Classes

During the grid search process, the best metrics were achieved at a certain configuration. Detailed class evaluation for the best grid search point with 50 classes yielded the following results: (1, 3, 0.82, 0.7305, 0.7305, 0.6770, 2702, 991, 46,577, 991) – indicating 991 misclassified images in total. A closer inspection of these results reveals that some classes perform significantly worse than others, which contributes to the overall misclassification rate as depicted in table Table 6.

The evaluation shows that some classes achieve excellent results, while others are significantly weaker, thus affecting the overall system performance.

By analyzing the performance metrics, we identified 17 problematic classes where faces were consistently difficult to classify due to extremely low-quality images. These images exhibited severe issues such as excessive blurring, very low resolution, visible

distortions, poor lighting, and partial or complete occlusion of facial features. As a result, embeddings extracted from these images were highly dispersed or overlapping, making accurate identification challenging even for human observers.

Table 6. Class-wise evaluation metrics reported across 50 classes.

Class	Precision	Recall	F1-Score	Support
1	0.98	1.00	0.99	57
2	0.92	0.99	0.95	67
3	1.00	1.00	1.00	257
4	1.00	0.98	0.99	249
5	0.65	0.96	0.78	57
6	0.92	1.00	0.96	102
7	1.00	1.00	1.00	41
8	0.91	0.67	0.77	73
9	0.98	1.00	0.99	56
10	0.31	1.00	0.48	16
11	0.20	1.00	0.33	38
12	0.99	0.97	0.98	77
13	0.88	1.00	0.93	7
14	0.21	0.14	0.17	49
15	0.52	1.00	0.68	25
16	1.00	1.00	1.00	12
17	1.00	1.00	1.00	18
18	0.94	1.00	0.97	58
19	1.00	1.00	1.00	25
20	1.00	1.00	1.00	98
21	1.00	0.91	0.95	199
22	1.00	1.00	1.00	17
23	0.00	0.00	0.00	49
24	0.40	0.69	0.51	114
25	1.00	0.96	0.98	49
26	0.87	1.00	0.93	13
27	0.95	1.00	0.98	61
28	1.00	0.05	0.10	57
29	0.00	0.00	0.00	57
30	0.12	0.05	0.07	185
31	0.48	0.69	0.56	58
32	0.99	1.00	0.99	87
33	1.00	1.00	1.00	71
34	1.00	0.09	0.17	173
35	0.51	1.00	0.68	53
36	0.75	1.00	0.85	156
37	0.61	1.00	0.76	35
38	0.62	1.00	0.76	73
39	0.53	0.98	0.69	61
40	1.00	1.00	1.00	13
41	0.00	0.00	0.00	0
42	1.00	1.00	1.00	45
43	0.00	0.00	0.00	53
44	0.00	0.00	0.00	57
45	0.42	0.93	0.58	83
46	0.62	1.00	0.76	69
47	1.00	0.10	0.18	153
48	0.42	0.95	0.58	77
49	1.00	0.91	0.95	56
50	1.00	0.04	0.08	142
Unknown	0.00	0.00	0.00	0

Notably, these 17 classes failed to produce reliable embeddings regardless of the parameter settings in our grid search, indicating that the difficulty was inherent to the data rather than the model configuration. To assess the system's performance on viable data, we conducted a secondary evaluation excluding these low-quality classes. This allowed us to determine the model's effectiveness when trained and tested only on images meeting minimum quality standards.

The criteria used for inclusion or exclusion of classes are summarized in Table 7, which captures the key factors that contributed to the low performance, such as general image blur, unrecognizability of the face to the naked eye, and insufficient resolution for feature extraction. After removing the 17 extremely low-quality classes, the best evaluation combination of parameters for the remaining dataset was tested, and the achieved results are presented in Table 8.

Table 7. Inclusion/Exclusion Criteria for Image Quality Assessment.

Criterion	Description
Unclear facial features	Faces not clearly visible due to occlusion, extreme pose, or poor framing.
General image blur	Motion blur, defocus, or camera shake causing smearing of facial details.
Unrecognizable to the naked eye	Face identity cannot be reliably determined by human inspection.
Low resolution	Images too small or pixelated to extract meaningful embeddings.
Lighting issues	Extreme overexposure or underexposure preventing clear visibility of facial landmarks.
Distortions	Warping, noise, or compression artifacts that affect face structure.

Table 8. Overall performance summary without low-quality classes.

Accuracy	Recall	F1 Score	TP	FP	TN	FN
0.8786	0.8786	0.8628	2295	317	9827	317

A detailed analysis of the confusion matrix [51] identifies a small subset of classes with persistent mutual confusion (e.g., 11–34, 39–50, 36–50, 29–49, 48–47, 35–23, 28–30). Visual inspection indicates that this behavior is primarily caused by poor data quality and strong intra-class variability, rather than inherent model limitations.

Several affected identities contain training and validation samples that differ significantly in appearance and, in some cases, do not consistently represent the same individual. Typical issues include severe blur, low resolution, occlusions, illumination variations, and geometric distortions, which are known to negatively affect embedding consistency and class separability [52].

To evaluate the impact of such data inconsistencies, the lowest-performing classes were excluded from the evaluation. This procedure was not applied to artificially improve performance, but to quantify the sensitivity of the system to corrupted or unreliable identity data. Following their removal, the number of dominant confusions decreased (e.g., 36–50, 39–50, 47–6, 34–13, 34–3), and overall performance metrics improved consistently as shown in Table 9.

Although iterative removal of confusing classes could theoretically approach 100% accuracy, such a strategy would not reflect realistic deployment scenarios. Instead, this analysis demonstrates that overall system performance is strongly constrained by data quality, and that eliminating identities with unreliable samples has a justified and measurable positive effect.

Table 9. Class-wise evaluation metrics reported without problematic classes.

Class	Precision	Recall	F1-Score	Support
1	0.9828	1.0000	0.9913	57
2	1.0000	1.0000	1.0000	67
3	0.7718	1.0000	0.8712	257
4	1.0000	1.0000	1.0000	249
5	0.6552	1.0000	0.7917	57
6	0.5862	1.0000	0.7391	102
7	1.0000	1.0000	1.0000	41
8	1.0000	0.6712	0.8033	73
9	0.9655	1.0000	0.9825	56
12	0.7353	0.9740	0.8380	77
13	0.1750	1.0000	0.2979	7
15	0.8621	1.0000	0.9259	25
16	1.0000	1.0000	1.0000	13
17	1.0000	1.0000	1.0000	18
18	1.0000	1.0000	1.0000	58
19	1.0000	1.0000	1.0000	25
20	0.9899	1.0000	0.9949	98
21	1.0000	1.0000	1.0000	199
22	1.0000	1.0000	1.0000	17
25	0.9800	1.0000	0.9899	49
26	1.0000	1.0000	1.0000	13
27	0.9683	1.0000	0.9839	61
28	1.0000	0.9825	0.9912	57
33	1.0000	1.0000	1.0000	71
34	1.0000	0.2948	0.4554	173
35	0.9464	1.0000	0.9725	53
36	0.6783	1.0000	0.8083	156
39	0.5259	1.0000	0.6893	61
40	0.8667	1.0000	0.9286	13
42	1.0000	1.0000	1.0000	45
46	1.0000	1.0000	1.0000	69
47	1.0000	0.3464	0.5146	153
50	1.0000	0.0493	0.0940	142
Unknown	0.0000	0.0000	0.0000	0

By removing problematic classes from the dataset, inter-class confusion is significantly reduced, leading to a measurable improvement in overall system performance. This further suggests that the primary sources of error were not rooted in the system's implementation, but rather in the inherent flaws of specific classes whose training and validation images hindered correct classification. Furthermore, due to the nature of this multi-class classification setup, metrics such as the ROC curve (compressed to the top-left) and the P-R curve (diagonal) exhibit a symmetrical behavior. Specifically, because the system is forced to assign each image to exactly one of the N classes, every misclassification inherently generates a false positive (FP) for the predicted class and a false negative (FN) for the ground-truth class. This 1:1 correspondence ensures that the total number of FPs and FNs remains equal across the dataset, causing precision and recall to converge to the same value. Schema of described classification system can be seen on Figure 7.

Face Recognition Error Logic (Single-Label Classification)

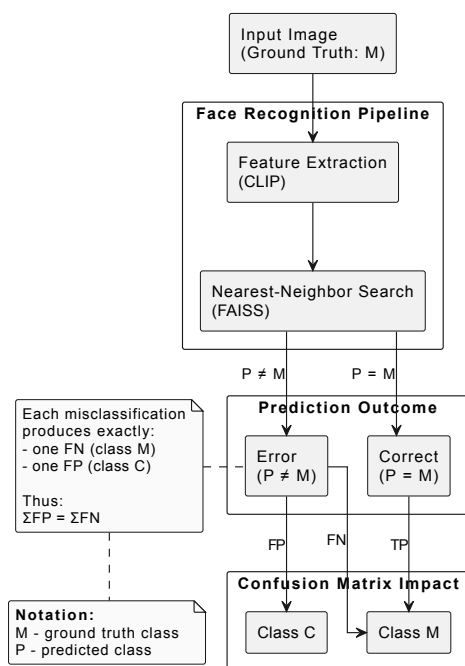


Figure 7. Schematic overview of described classification error logic.

A set of representative train-validation data pairs is shown in Figure 8.

Class	Images	Description
20		Class 20 achieves perfect metrics with a precision of 1.00, meaning that all samples predicted as class 20 are indeed correct.
17		Class 17 also records excellent results with a perfect precision of 100%, making it flawless in predictions.
46		Class 46 achieves a precision of 0.62, which is decent, but there is room for improvement.
39		Class 39 has a precision of 0.53, which means there is a significant number of false positive predictions.

The "Unknown" class achieves a precision of 0.00, which is actually a good sign because the "Unknown" class is defined for cases where the model doesn't know how to classify a particular image. This indicates that the system will provide an output for each input that, with sufficient data quality, will be accurate.

Figure 8. Sample training-validation image pairs from a subset of classes.

4.2. Comparative Analysis with Baseline Models

To evaluate the specific advantages of CLIP-based embeddings, we conducted a comparative analysis using a ResNet50 architecture as a baseline, following the benchmarking approach suggested in literature regarding vision-language model evaluation [53]. Both models were integrated into the same FAISS-based retrieval pipeline to ensure an apples-to-apples comparison of their feature extraction capabilities. The results, summarized in Table 10, demonstrate a significant performance gap. The CLIP-based pipeline achieved an accuracy of 87.75%, whereas the ResNet50 baseline reached only 36.29%. This disparity highlights the limitations of traditional CNNs trained on general object datasets (e.g., ImageNet) when applied to face recognition without extensive fine-tuning. In contrast, CLIP's contrastive pre-training provides a more robust and semantically rich representation space, allowing for superior class separability even in unconstrained environments.

Table 10. Performance comparison between the proposed CLIP-based pipeline and the ResNet50 baseline.

Model	Embedding Dimensions	Accuracy	F1-Score (Weighted)
CLIP (Proposed)	512	87.75%	0.8628
ResNet50 (Baseline)	2048	36.29%	0.2766

4.3. Time-Based Scalability Test

In testing the system's performance, the effect of increasing the number of classes on the loading time was observed, with a specific focus on the phases of converting images into embeddings and loading them into FAISS [45,48]. As expected, the loading time did not grow proportionally with the number of classes. As seen on Figure 9, instead of a linear relationship, the system showed disproportionate growth in time (the increase in the number of classes was greater than the increase in execution time), indicating optimized loading processes and scalability. This result suggests that the system can efficiently handle larger numbers of classes without a significant increase in loading time. Expected increase would follow the given formula:

$$\text{Total loading time} = \text{Number of classes} \times \text{Loading time per class}$$

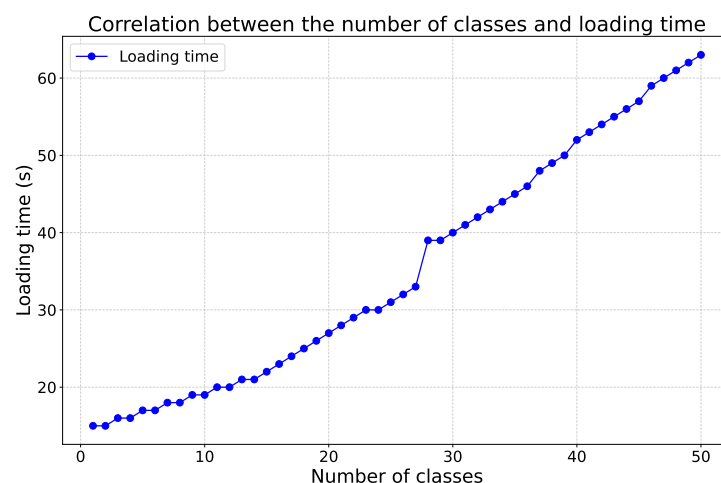


Figure 9. A visualization of time-based evaluation with number of classes on *x*-axis and loading time (in seconds) on *y*-axis.

However, the increase was noticeably smaller. For loading the system with 1 class, it took 14 s, while for loading the system with 50 classes, it took 64 s. Therefore, with an increase in the number of classes by 4900%, the loading time increased by 357.14%.

4.4. Metrics-Based Scalability Test

The graph on Figure 10 shows how the model's performance depends on the number of classes, with Precision, Recall, and F1-score initially remaining at a high level, and a significant drop occurring only after approximately 30 classes. However, the decline in performance is not necessarily only related to the increase in the number of classes, but also to the quality and diversity of the images in the dataset. Also, since precision and recall values remain consistently aligned, the precision-recall curve would closely resemble a diagonal line from (0,0) to (1,1), providing minimal additional information and thus not being particularly informative in this context. Although all three metrics decrease with the increase in the number of classes, data quality plays a crucial role in maintaining good results. With a well-prepared dataset, the model can maintain high performance even with a larger number of classes. Moreover, by adding high-quality classes, it is possible to improve the metrics, suggesting that the structure, quality, and informativeness [54] of the data has a greater impact on system performance than the number of classes alone.

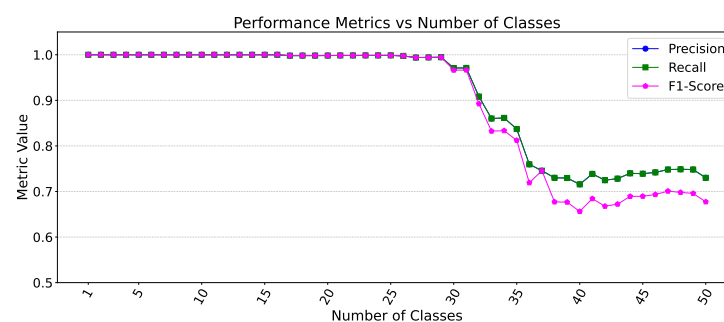


Figure 10. A visualization of performance-based evaluation with the number of classes on the x -axis and the metric values (scaled between 0.0 and 1.0) on the y -axis.

Furthermore, separated plots for each metric are shown on Figure 11.

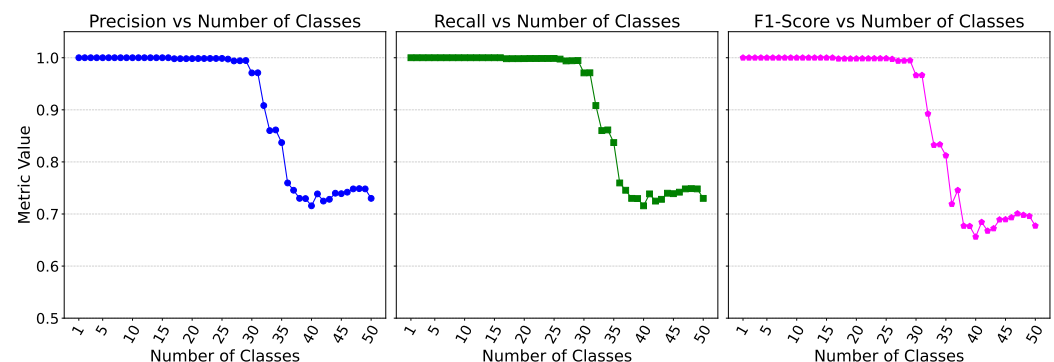


Figure 11. Visualization of performance evaluation results for precision, recall, and F1-score using separate plots.

4.5. Reaching Peak Performance

After a certain threshold value, the model reaches its maximum classification potential, where further increases in the threshold no longer affect the metrics. This indicates a high confidence in its decisions and the ability to clearly separate positive and negative examples. All positive classifications are above a certain confidence threshold, while negative ones are below it, meaning the model consistently and accurately makes decisions without uncertain predictions in the middle probability range. This effect confirms the robustness of the model and its resistance to threshold changes, achieving optimal accuracy in real-world scenarios. This is especially useful in systems where classification reliability

is crucial [55,56], and further adjustment can only be used for fine-tuning the model in specific cases. Precision and recall remain the same because FP and FN are always equal within each parameter combination. This means the model consistently confuses certain classes, regardless of the threshold or k-values it considers.

4.6. Comparative Evaluation of Accuracy and Computational Overhead

Compared to similar solutions listed in Table 11, described CLIP + FAISS system has certain advantages. Firstly, it does not need to be trained in the traditional sense; new classes/new faces can be added very quickly and easily. Furthermore, the system is extremely easy to implement in applications that involve various forms of face detection without requiring excessive modification, as the operating principle remains unchanged. Additionally, thanks to CLIP's strong embedding extraction capabilities, the system can achieve significant performance improvements when a high-quality dataset is collected under controlled conditions. This also enables successful recognition of a large number of different faces, particularly when the conditions for capturing reference images are predefined. It is particularly interesting that successful recognition does not require the live detection input to capture the entire face. The system is not limited by head coverings, masks, etc. If a face is still sufficiently recognizable even with such coverings, the system will classify it correctly as seen on Figure 12. The key is that the system successfully identifies specific parts of the face that contribute most to distinguishing an individual from others.

As illustrated in Table 11, the proposed CLIP+FAISS system achieves an accuracy of 87.75% on the YTF dataset. While this is lower than the 95.12% and 98.00% achieved by FaceNet [5] and ArcFace [10] respectively, it is crucial to note that these baseline models are heavily fine-tuned on millions of identity-specific facial images. In contrast, our solution utilizes a zero-shot CLIP transformer, which prioritizes architectural flexibility and significantly lower resource consumption, as detailed in Table 12.

Table 11. Comparative performance analysis standardized on the YouTube Faces (YTF) dataset.

Model	Architecture	Dataset	Accuracy	Inference (ms)	Approach
FaceNet [5]	CNN + Triplet	YTF	95.12%	400	Supervised
DeepFace [6]	9-layer CNN	YTF	91.40%	1500	Supervised
VGG-Face [57]	VGG-16	YTF	91.60%	1200	Supervised
ArcFace [10]	ResNet + Arc	YTF	98.00%	600	Supervised
Proposed System	CLIP-ViT-B/32	YTF	87.75%	420	Zero-Shot

Table 12. Scalability and hardware resource utilization comparison.

Metric	Proposed	DeepFace	VGG-Face	ArcFace
Avg. CPU Load (%)	43.17%	~85.0%	~90.0%	~65.0%
Avg. RAM Usage (MB)	1364.80	~1800.0	~2200.0	~1500.0
Inference Time (ms)	420.66	~1500.0	~1200.0	~600.0
Peak GPU Memory (MB)	1165.32	~2500.0	~3000.0	~2000.0

Note: Competitor hardware metrics are estimated based on standard CPU/GPU implementation benchmarks in existing literature.



Figure 12. Live detection with covered face.

5. Discussion

5.1. Testing Limitations

With 25 classes, there are minor misclassifications on certain similar images, but they are almost negligible, and each class is mostly correctly predicted overall. At 50 classes, the classification drops to 73%. A way to improve the results would definitely be to manually record 50 or more people in 2 videos under different conditions, which would provide higher quality test data and better embeddings. This part will be further explored during the scalability test evaluation. Testing and validation were carried out by comparing predictions on the validation set with images from the training set, rather than real-time live input. Although this evaluation method is necessary, it does not fully reflect the challenges of real-time face recognition from a live feed, where factors such as lighting changes, different facial angles, facial expressions, and resolution variations can further impact model performance, but it also excludes some advantages that live testing provides. Unfortunately, due to limitations in standard experimental protocols, simulating these real-world conditions through the validation set currently represents the closest possible method of testing before implementation in real-time face recognition systems. Using a live feed allows for the analysis of many more consecutive frames in a shorter time, which enables the aggregation of predictions and reduces the impact of certain anomalies that may occur on individual images due to poor lighting, shooting angles, blurriness, etc. Instead of relying on static images, the system can collect more facial samples from the user over time, improving recognition reliability. This approach is particularly useful in dynamic systems where the goal is to achieve stable and precise recognition, such as security systems or contactless authentication [58,59].

Furthermore, live input enables adaptive techniques such as face tracking, where the model can continuously correct and improve detection by analyzing micro-movements and different facial expressions. This flexibility often leads to better overall accuracy in real-world environments than would be expected based on validation results obtained from traditional static image testing methods.

5.2. Real-Life Applications

The described system can be implemented for real-time face detection and recognition using a computer camera by passing camera frames as input, creating a bounding box [60,61] around each detected person, extracting embeddings, and passing them into the described system. The predicted most likely class (with the most votes) is returned as the function's output and added as a label on the box. Additionally, the system can handle multiple known faces simultaneously in the same frame as seen on Figure 13, which adds an extra dimension to the system and enables more advanced recognition in complex situations. It is also important to note that faces can be classified in challenging conditions such as low-light environments, as seen on Figure 14. This implementation provides numerous

possibilities, such as creating systems for student attendance tracking, payroll calculation based on hours worked, security systems, various interactive interfaces, and more.

The system achieves exceptional real-time performance, with a low query latency of under 50 ms and a high processing throughput of approximately 200 frames per second (FPS). These metrics were measured using a standard webcam on a Lenovo IdeaPad Slim 3i laptop at a resolution of 1920×1080 pixels. During testing, subjects were required to remain relatively stationary in front of the camera for approximately 2 s to allow accurate embedding extraction. Experiments were conducted in typical indoor environments with varying lighting conditions, ranging from darker to brighter settings, demonstrating that the system can maintain high performance even under realistic illumination changes. The low latency and high FPS are a direct result of the optimized FAISS vector search implementation, confirming the system's suitability for fast, dynamic real-world applications. The proposed architecture demonstrates a significant advantage in resource efficiency by utilizing the relatively lightweight CLIP-ViT-B/32 transformer model as a feature extractor, in contrast to the computationally heavy deep convolutional neural networks (CNNs) typically found in state-of-the-art face recognition systems. Beyond classification accuracy, the system's scalability was rigorously evaluated through resource profiling. Performances were tested while executing a full end-to-end inference pipeline. During these operations, the system maintains a highly efficient resource footprint with an average RAM usage of 1364.80 MB and a CPU load of 43.17%, ensuring that the host machine remains responsive for concurrent tasks. Peak GPU memory consumption was measured at 1165.32 MB, further confirming that the architecture is well-suited for deployment on standard consumer-grade hardware or edge devices without requiring high-end industrial servers.



Figure 13. Live detection of multiple faces.

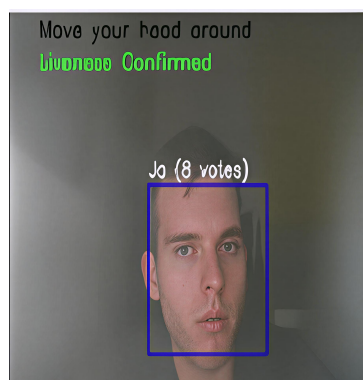


Figure 14. Live detection in dark surroundings.

6. Conclusions

Face recognition analysis has shown that image quality and background significantly impact the model's accuracy. When backgrounds are too varied, embeddings become less precise, making it harder to correctly recognize individuals. Additional challenges arise from text, objects in the frame, and large lighting changes, as these introduce variations that are not related to the identity of the person but to the environment. This problem is particularly evident when the system works with a large number of different faces - while the model achieves high accuracy under controlled conditions, its performance may decline if the data varies in lighting, capture angle, or image quality. Factors such as blurring, low resolution, and contrast can further reduce the reliability of recognition. To improve results, it would be beneficial to standardize the conditions for capturing reference images or apply methods that ensure greater consistency in input data. For example, in a system for automated student recognition at a university, it would be advisable for all reference images to be captured under the same conditions and with a uniform background, which would neutralize recognition errors. In addition to technical limitations, ethical considerations play a role in the responsible deployment of described facial recognition system [62,63].

The approach presented in this work ensures that privacy is preserved by design: users provide input images voluntarily, and these images are immediately transformed into non-reversible embeddings used exclusively for recognition. Original images are neither stored nor accessible outside the system, which eliminates the risk of misuse or external leakage. Furthermore, since the system operates entirely on local infrastructure without transmitting biometric data externally, it aligns with modern data protection principles. These safeguards make the proposed solution suitable for privacy-sensitive applications such as education and access control, where both efficiency and ethical standards are essential.

In conclusion, the results show that the system does not have issues with face recognition, but its accuracy is most affected by external factors that can degrade the quality of the embeddings.

Unlike traditional methods [64] that rely on deep convolutional networks [65–67] specialized for faces, our system uses multimodal [68] embeddings, offering greater flexibility and greater application in various recognition scenarios. Furthermore, the integration of FAISS indexing allows for extremely fast searches, significantly improving the efficiency and responsiveness of the model in real-time. Thus, this work contributes to the field of computer vision, presenting a new methodology that combines the power of transformer embeddings and optimized vector search, opening opportunities for future research and improvements in fast and scalable face recognition.

Author Contributions: Conceptualization, A.L. and N.T.; Methodology, A.L.; Software, A.L.; Validation, N.T., S.B.Š. and I.L.; Formal analysis, A.L.; Investigation, A.L.; Data curation, A.L.; Visualization, A.L.; Writing—original draft preparation, A.L.; Writing—review and editing, N.T., S.B.Š. and I.L.; Supervision, N.T.; Project administration N.T. and I.L.; Funding acquisition, S.B.Š. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the following projects: SPIN projects IP.1.1.03.0120, IP.1.1.03.0158 and IP.1.1.03.0039; the EU NextGeneration under the Juraj Dobrila University of Pula institutional research project number IIP_010144 and IIP_010136; and EC Digital Europe Programme EDIH Adria 2.0 101256325.

Data Availability Statement: The data supporting the findings of this study are publicly available. The evaluation and experimental code is provided as a Jupyter Notebook 7.5 hosted on Google Colaboratory at https://colab.research.google.com/drive/1hufVhOrntyKX8XniSwup0tdhER01eD_1?usp=sharing (accessed on 16 December 2024) (V1, Antonio Labinjan, Pula, Croatia). The experiments were conducted using the YouTube Faces (YTF) dataset, which can be accessed at <https://www.cs.tau.ac.il/~wolf/ytfaces/> (accessed on 16 December 2024).

Acknowledgments: The authors acknowledge the publicly available YouTube Faces (YTF) dataset used in this study. During the preparation of this manuscript, ChatGPT 3.5 (OpenAI) was used to assist with language editing and text polishing. All generated content was carefully reviewed and revised by the authors, who take full responsibility for the final manuscript.

Conflicts of Interest: The authors declare no conflicts of interest

Abbreviations

The following abbreviations are used in this manuscript:

CLIP	Contrastive Language–Image Pre-Training
FAISS	Facebook AI Similarity Search
CNN	Convolutional Neural Network
YOLO	You Only Look Once
IoU	Intersection over Union
L2	Euclidean distance
ANN	Approximate Nearest Neighbors
PCA	Principal Component Analysis
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
F1	F1-score
FPS	Frames Per Second
YTF	YouTube Faces dataset
P–R	Precision–Recall
ROC	Receiver Operating Characteristic

References

1. Zhao, W.; Chellappa, R.; Phillips, P.J.; Rosenfeld, A. Face recognition: A literature survey. *ACM Comput. Surv. (CSUR)* **2003**, *35*, 399–458. <https://doi.org/10.1145/954339.954342>.
2. Li, L.; Mu, X.; Li, S.; Peng, H. A Review of Face Recognition Technology. *IEEE Access* **2020**, *8*, 139110–139120. <https://doi.org/10.1109/ACCESS.2020.3011028>.
3. Wang, M.; Deng, W. Deep Face Recognition: A Survey. *Neurocomputing* **2021**, *429*, 215–244. <https://doi.org/10.1016/j.neucom.2020.10.081>.
4. Zhu, S. Enhancing Facial Recognition: A Comprehensive Review of Deep Learning Approaches and Future Perspectives. *Appl. Comput. Eng.* **2024**, *110*, 137–145. <https://doi.org/10.54254/2755-2721/110/2024MELB0107>.
5. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A Unified Embedding for Face Recognition and Clustering. *arXiv* **2015**, arXiv:1503.03832.
6. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2014; pp. 1701–1708. <https://doi.org/10.1109/CVPR.2014.220>.
7. Thinking Neuron. Face Recognition Using Deep Learning (CNN) in Python. 2020. Available online: <https://thinkingneuron.com/face-recognition-using-deep-learning-cnn-in-python/> (accessed on 25 April 2025).
8. Parmar, D.N.; Mehta, B.B. Face Recognition Methods & Applications. *arXiv* **2014**. Available online: <http://arxiv.org/abs/1403.0485> (accessed on 19 April 2025).
9. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep Face Recognition. In *Proceedings of the British Machine Vision Conference*; BMVA Press: Glasgow, UK, 2015; pp. 1–12. <https://doi.org/10.5244/C.29.41>.

10. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. ArcFace: Additive angular margin loss for deep face recognition. *arXiv* **2018**, arXiv:1801.07698.
11. Qi, D.; Tan, W.; Yao, Q.; Liu, J. YOLO5Face: Why Reinventing a Face Detector. *arXiv* **2021**, arXiv:2105.12931.
12. Wang, Y.; Zheng, J. Real-time face detection based on YOLO. In Proceedings of the 2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), Jeju, Republic of Korea, 23–27 July 2018. <https://doi.org/10.1109/ICKII.2018.8569109>.
13. Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. *arXiv* **2018**, arXiv:1801.06146.
14. Facebook AI Research. Faiss: A Library for Efficient Similarity Search and Clustering of Dense Vectors. 2025. Available online: <https://faiss.ai/> (accessed on 16 April 2025).
15. DataCamp Community. What Is Faiss (Facebook AI Similarity Search)? 2024. Available online: <https://www.datacamp.com/blog/faiss-facebook-ai-similarity-search> (accessed on 22 May 2025).
16. Needham, M. FAISS and Approximate Nearest Neighbors: Cell Probe. *Mark Needham's Blog* 2023. Available online: <https://www.markneedham.com/blog/2023/09/14/faiss-approximate-nearest-neighbors-cell-probe/> (accessed on 20 May 2025).
17. Johnson, J.; Douze, M.; Jégou, H. Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* **2019**, *7*, 53–547.
18. Shafiq, A. How to Use FAISS to Build Your First Similarity Search. 2022. Available online: <https://medium.com/loopio-tech/how-to-use-faiss-to-build-your-first-similarity-search-bf0f708aa772/> (accessed on 17 April 2025).
19. Sokolova, M.; Lapalme, G. Systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437.
20. Luu, N.T. CLIP Unreasonable Potential in Single-Shot Face Recognition. *arXiv* **2024**, arXiv:2411.12319.
21. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models From Natural Language Supervision. *arXiv* **2021**, arXiv:2103.00020.
22. Bhat, A.; Jain, S. Face Recognition in the Age of CLIP & Billion Image Datasets. *arXiv* **2023**, arXiv:2301.07315.
23. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer: Berlin/Heidelberg, Germany, 2010. <https://doi.org/10.1007/978-1-84882-935-0>.
24. Jurafsky, D.; Martin, J.H. *Speech and Language Processing*, 3rd ed.; Prentice Hall: Hoboken, NJ, USA, 2023. Available online: <https://web.stanford.edu/~jurafsky/slp3/> (accessed on 25 February 2026).
25. Harsh, T.; Kumar, T.; Mohanty, A. and Pandey, A. Predictive Maintenance of Industrial Equipment Using ML and IoT. In Proceedings of the 2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIoT), Vellore, India, 3–4 May 2024.
26. Keogh, E.; Kasetty, S. Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowl. Inf. Syst.* **2001**, *3*, 263–286.
27. Lampert, C.H.; Nickisch, H.; Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2009; pp. 951–958.
28. Palatucci, M.; Pomerleau, D.; Hinton, G.E.; Mitchell, T.M. Zero-shot learning with semantic output codes. In *Proceedings of the Advances in Neural Information Processing Systems*; NEURIPS: Sydney, Australia, 2009; pp. 1410–1418.
29. Frome, A.; Corrado, G.S.; Shlens, J.; Bengio, S.; Dean, J.; Ranzato, M.; Mikolov, T. DeViSE: A deep visual-semantic embedding model. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2013; pp. 2121–2129.
30. Ramesh, A.; Dhariwal, P.; Leike, Y.; Chen, M.; Murthy, K.; Radford, A.; Sutskever, I.; Zaremba, W. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv* **2022**, arXiv:2204.06125.
31. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-Resolution Image Synthesis with Latent Diffusion Models. *arXiv* **2021**, arXiv:2112.10752.
32. Lorencin, I.; Frank, D.; Vušić, D. Zero-Shot Learning in Maritime Domain: Classification of Marine Objects Using CLIP. *Sci. J. Marit. Res.* **2024**, *38*, 239–249.
33. Tan, J.; Yang, J.; Wu, S.; Chen, G.; Zhao, J.J. A critical look at the current train/test split in machine learning. *arXiv* **2021**, arXiv:2106.04525.
34. Brownlee, J. Distance Measures for Machine Learning. 2019. Available online: <https://machinelearningmastery.com/distance-measures-for-machine-learning/> (accessed on 24 April 2025).
35. Hadsell, R.; Chopra, S.; LeCun, Y. Dimensionality Reduction by Learning an Invariant Mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2006; pp. 1735–1742. <https://doi.org/10.1109/CVPR.2006.100>.
36. Bentley, J.L. Multidimensional binary search trees used for associative searching. *Commun. ACM* **1975**, *18*, 509–517.
37. Indyk, P.; Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*; Association for Computing Machinery: New York, NY, USA, 1998; pp. 604–613.

38. Malkov, Y.; Yashunin, D. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 824–834.
39. Kundu, R. Precision vs. Recall: Differences, Use Cases & Evaluation. 2022. Available online: <https://www.v7labs.com/blog/precision-vs-recall-guide> (accessed on 24 April 2025).
40. Berman, M.; Triki, A.R.; Blaschko, M.B. Multi-Class Classification Evaluation with Precision-Based Metrics. In *Proceedings of the International Conference on Machine Learning (ICML)*; PMLR: Cambridge, MA, USA, 2020; pp. 759–768.
41. Powers, D.M.W. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Int. J. Mach. Learn. Technol.* **2011**, *2*, 37–63.
42. Wolf, L.; Hassner, T.; Maoz, I. YouTube Faces Database. 2011. Available online: <https://www.cs.tau.ac.il/~wolf/ytfaces/> (accessed on 24 April 2025).
43. Lever, J.; Krzywinski, M.; Altman, N. Principal Component Analysis. *Nat. Methods* **2017**, *14*, 641–642. <https://doi.org/10.1038/nmeth.4346>.
44. Jolliffe, I.T. Principal Component Analysis. In *Springer Series in Statistics*; Springer: New York, NY, USA, 2002. <https://doi.org/10.1007/b98835>.
45. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128792–128822. <https://doi.org/10.1109/ACCESS.2019.2939201>.
46. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
47. GeeksforGeeks. Precision-Recall Curve in Machine Learning. Available online: <https://www.geeksforgeeks.org/precision-recall-curve-ml/> (accessed on 22 April 2025).
48. Scepanski, E.; Zillner, S. AI Systems and their Scalability—A Systematic Literature Review. In *Proceedings of the 35th Australasian Conference on Information Systems (ACIS 2024)*, Canberra, Australia, 4–6 December 2024.
49. Developers, N. numpy.linalg.norm—NumPy v1.26 Manual. 2024. Available online: <https://numpy.org/doc/2.1/reference/generated/numpy.linalg.norm.html> (accessed on 22 April 2025).
50. Yang, S.; Xiao, W.; Zhang, M.; Guo, S.; Zhao, J.; Shen, F. Image Data Augmentation for Deep Learning: A Survey. *arXiv* **2022**, arXiv:2204.08610.
51. GeeksforGeeks. Understanding the Confusion Matrix in Machine Learning. Available online: <https://www.geeksforgeeks.org/confusion-matrix-machine-learning/> (accessed on 22 April 2025).
52. Cai, J.; Chen, J.; Liang, X. Single-Sample Face Recognition Based on Intra-Class Differences in a Variation Model. *Sensors* **2015**, *15*, 1071–1087.
53. Chen, J.; Duan, H.; Zhang, X.; Gao, B.; Grau, V.; Han, J. From Gaze to Insight: Bridging Human Visual Attention and Vision Language Model Explanation for Weakly-Supervised Medical Image Segmentation. *arXiv* **2025**. Available online: <http://arxiv.org/abs/2504.11368> (accessed on 25 February 2026).
54. MacKay, D.J.C. *Information Theory, Inference, and Learning Algorithms*; Cambridge University Press: Cambridge, UK, 2003.
55. Grother, P.; Ngan, M.; Hanaoka, K. Face Recognition Vendor Test (FRVT) Part 3: Demographic Effects. 2019. Available online: <https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8280.pdf> (accessed on 21 May 2025).
56. Vishwanatha, C.R.; Asha, V.; Sajju, B.; Suma, N.; Reddy, T.R.M.; Sumanth, K.M. Face Recognition and Identification Using Deep Learning. In *Proceedings of the 2023 Third International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Bhilai, India, 5–6 January 2023.
57. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep Face Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, Swansea, UK, 7–10 September 2015.
58. Dayal, A.; Paluru, N.; Cenkeramaddi, L.R.; Yalavarthy, P.K. Design and Implementation of Deep Learning Based Contactless Authentication System Using Hand Gestures. *Electronics* **2021**, *10*, 182.
59. CIE-Group. Access Control Contactless Authentication Methods. Available online: <https://cie-group.com/how-to-av/videos-and-blogs/contactless-authentication> (accessed on 21 May 2025).
60. Viola, P.; Jones, M. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; IEEE: Piscataway, NJ, USA, 2001; pp. 511–518. <https://doi.org/10.1109/CVPR.2001.990517>.
61. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. <https://doi.org/10.1109/LSP.2016.2603342>.
62. Kavoliūnaitė-Ragauskienė, E. Right to Privacy and Data Protection Concerns Raised by the Development and Usage of Face Recognition Technologies in the European Union. *J. Hum. Rights Pract.* **2024**, *16*, 658–674.
63. Selinger, E.; Leong, B. The Ethics of Facial Recognition Technology. In *The Oxford Handbook of Digital Ethics*; Ess, C., Hagengruber, R., Eds.; Oxford University Press: Oxford, UK, 2022; pp. 590–610. <https://doi.org/10.1093/oxfordhb/9780198857815.013.32>.

64. Barcic, E.; Grd, P.; Tomicic, I. Convolutional Neural Networks for Face Recognition: A Systematic Literature Review. 2023. Available online: <https://www.researchsquare.com/article/rs-3145839/v1> (accessed on 25 February 2026).
65. Fukushima, K. Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biol. Cybern.* **1980**, *36*, 193–202.
66. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.
67. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551.
68. Manmothe, S.; Jadhav, J. Integrating Multimodal Data For Enhanced Analysis And Understanding: Techniques For Sentiment Analysis And Cross-Modal Retrieval. *J. Adv. Zool.* **2024**, *45*, 22–28. <https://doi.org/10.53555/jaz.v45iS4.4144>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.