International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

An Unsupervised Deep Embedding-Based Face Clustering Framework for Large-Scale Photo Archive Organization

Mokshav Bavishi
School of CS & IT,
Jain (Deemed-to-be University),
Bengaluru, India.

Devansh Jaiswal
School of CS & IT,
Jain (Deemed-to-be University),
Bengaluru, India.

Abstract: - Out there in digital picture stacks, things are moving fast - needing smart ways to sort and link faces, especially when dealing with massive, nameless collections. Most current systems for spotting people rely on labeled training sets, so they stumble where no tags exist. A fresh approach here digs into using deep learning embeddings to group faces, built not for small tweaks but for handling huge piles of photos with zero name attached. This setup brings together MTCNN-driven face spotting and FaceNet's deep embedding analysis. Handling large datasets becomes manageable through feature scaling. Sometimes PCA steps in to lower data complexity. After that comes DBSCAN, grouping faces by natural density patterns. Shape-agnostic clustering works better here because it ignores forced centroids. Noise and outliers get flagged without extra rules. Testing on the Labeled Faces in the Wild (LFW) set showed clear benefits - clusters grouped tightly, with good space between them, measured by Silhouette Score and Davies–Bouldin Index. When placed beside standard approaches like K-Means or Agglomerative clustering, performance held up better under change, especially without labeling guidance. This method handles big visual collections well, needing less computation than alternatives; its value extends beyond single tasks, touching areas such as file sorting, forensic analysis, and automated media review.

INTRODUCTION:

With more people using smartphones and sharing posts online, huge collections of digital photos are growing fast. From home backups to security footage, business libraries, and official files, piles of visuals keep piling up every day. Sorting through so many images without tags or clear descriptions is getting harder, even for tech-savvy users. One key job in analyzing images is sorting faces into groups - this helps organize photos by who appears in them.

Face detection often uses training sets tagged by name. Though systems like CNNs now spot faces well, they struggle when pictures pile up without names. When labels are missing, grouping faces by shape can quietly sort who is who. This way, systems find people without being told.

With newer methods in deep learning, it becomes easier to pull out distinct facial features by using pre-trained systems like FaceNet or alike. These systems place face images into a tight data realm where how close two faces are in vector space ties back to how similar they are. Yet, grouping data in spaces filled with many dimensions still poses a real puzzle. Figures clustered by K-Means might miss curved patterns, plus knowing how many groups exist upfront is needed. Instead of guessing, trying a different path could help when dealing with vast amounts of data. When it comes to actual photos - like faces across ages or lighting - extra noise like misclassifications, hidden faces, or shifting faces, eyes, or smiles makes sorting identities tougher.

A fresh look at current shortfalls leads to this idea: a system for grouping faces in vast photo collections, using embedded learning without labels. It draws on three key pieces. First, spotting faces across images relies on MTCNN, built for handling multiple detection tasks well. Second, shaping meaningful vectors from faces comes from a trained model that learns similarities through metrics. Third, grouping similar faces follows via DBSCAN, sensing local densities rather than fixed shapes. Instead of assuming cluster forms or counts, it finds them based on natural separation patterns. Noise slips through or gets filtered, since shape and spread matter more than rigid templates. Unusual layouts or scattered groups are handled naturally, where standard k-means might falter.

Feature scaling boosts cluster reliability while cutting down processing time. When needed, PCA narrows data space for faster pattern detection. Evaluation relies on LFW examples

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

checked by Silhouette and Davies–Bouldin values. Compared to standard clustering tactics, results show tighter groupings, clearer distinctions, and stronger consistency when labels are missing.

This effort brings forward key outcomes in a clear manner. What stands out begins here:

1. A method that sorts photos by identity without human labeling, able to handle very big collections.
2. Using layered embedding systems alongside density-aware grouping methods to adjust how identities are found.
3. A method to evaluate clustering accuracy in high-dimensional facial feature space, done in a consistent way.
4. Testing shows this method handles uncertainty better than standard grouping approaches.

PROBLEM STATEMENT:

Picture collections grow fast online, making it tough to sort and find them by machine alone. Thousands to millions of faceless photos sit in places like home drives, public feeds, security logs, and office folders. When so many images pile up without a name tag attached, linking them to people changes how systems must handle floods. Sorting into clusters using faces rather than dates sharpens search speed under heavy loads. Still, there are no labeled faces tied to real people, which blocks standard recognition tools from learning.

Even though recent advances in face detection reach high precision when trained with guidance, their success still depends on data tagged in advance plus fixed categories for people. When dealing with actual photos stored across systems, labels often missing, shifting over time, or only partly defined - standard sorting methods fall short. On top of that, assigning labels by hand demands extensive effort, risks mistakes, while also increasing costs beyond reasonable limits.

Face grouping without guidance can sort photos by how similar faces are, yet knowing who's in each picture isn’t needed. That idea brings its own set of hurdles though.

High-Dimensional Feature Representation:
Out in the wild of data clustering, most older methods stumble when faced with vectors pulled from deep nets. These outputs float around in sprawling spaces shaped by complex patterns, making it tough for basic grouping tools to keep track of distances between them. Their usefulness drops as they lose the ability to tell apart fine differences across examples.

Unknown Number of Identities:
Picture collections are often huge, yet nobody knows ahead of time how many distinct people appear in them. This uncertainty renders tools like K-Means - which need cluster numbers fixed up front - less useful here.

Intra-Class Variability and Inter-Class Similarity:
Light levels change, plus how someone is standing, what they show on their face, if things are blocked, or how clear the photo looks - all spread data within one class. At the same time, when people generally resemble each other, differences across classes can blur.

Noise and outliers show up when data acts erratically.
Faces in real photos often blur. Sometimes only part shows up. Wrong matches pop through. Background clutter sneaks in. This kind of thing tends to hurt how well groups form.

Most clustering techniques assume steady patterns, yet struggle when noise distorts data or shapes become irregular. Because of this gap, an urgent requirement exists - for methods that adapt well, handle big datasets efficiently, and group face images reliably even across messy collections.

It has to:

1. Run operations missing oversight on learning user profiles
2. Adaptively determine cluster structures
3. Handle complex data points efficiently
4. Find the noisy clips first. Pull them apart from the rest.
5. Effortless scaling for massive image collections

Facing these issues, the research suggests a method using deep learning embeddings without labels - grouping faces through distinctive features while adjusting to local patterns. Instead of fixed rules, it applies density-aware grouping alongside metric learning to find identities in real-time shifts.

LITERATURE REVIEW:

[1] This seminal work introduced FaceNet — a deep convolutional network trained with a triplet loss that maps facial images to a compact embedding space where Euclidean distance corresponds to face similarity. It set a benchmark for face recognition and face clustering tasks by producing highly discriminative 128-D descriptors.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Key Contribution: Deep metric learning for faces.

[2] Zhang et al. proposed MTCNN, a cascaded network that simultaneously performs face detection and alignment using facial landmarks. It significantly improves the quality of detected face crops, which leads to more accurate embeddings and downstream tasks like clustering.

Key Contribution: Reliable preprocessing for face representation.

[3] This classic clustering algorithm identifies clusters as high-density regions and marks points in sparse areas as noise. It does not require a predetermined number of clusters, making it suitable for unlabeled face clustering where identities are unknown.

Key Contribution: Non-parametric clustering with noise detection.

[4] Otto, Wang, and Jain studied large-scale face clustering at the scale of millions of images. They introduced efficient blocking, rank-order clustering, and scalable techniques that can handle massive face collections found in real systems.

Key Contribution: Scalable clustering for millions of identities.

[5] Caron et al. proposed DeepCluster, a method that alternates between clustering feature representations and using those clusters as pseudo-labels to improve the network. Though general (not face-specific), this work influences self-supervised clustering for any visual domain.

Key Contribution: Joint learning of features + clusters.

[6] McInnes and Healy extended DBSCAN to HDBSCAN, removing the dependency on the epsilon parameter and producing more stable clusters across varying densities. It's useful in complex data where cluster densities differ.

Key Contribution: Improved density clustering with hierarchical robustness.

[7] This work learns face embeddings without identity labels by mining temporal continuity in videos and contrastive learning. It demonstrates that unsupervised representation learning can approach supervised accuracy in some cases.

Key Contribution: Self-supervised face embedding learning.

[8] Shi et al. proposed a self-learning framework where unsupervised clustering outcomes are iteratively refined by re-training the model, improving both representation and cluster quality. This method reduces noise and improves purity.

Key Contribution: Iterative refinement of clusters and embeddings.

[9] Yang et al. built an affinity graph of face embeddings using nearest neighbor relationships. Spectral or graph-based clustering then partitions the graph, improving separation for ambiguous cases where embedding distances alone are weak.

Key Contribution: Graph-based clustering enhancement.

[10] Shi and Jain introduced constraints into face clustering: use pairwise similarity (must-link / cannot-link) to improve cluster purity. They show that embedding + constraint learning increases clustering accuracy, particularly in ambiguous cases.

Key Contribution: Pairwise constraints for better clustering.

RESEARCH METHODOLOGY AND DATA COLLECTION:

A. RESEARCH DESIGN

This work uses a numbers-based setup, testing how well a method for sorting faces in big image collections works when it runs on its own. That approach checks whether grouping people through hidden patterns helps enough.

What sits behind the main idea of this work begins like this:

"Using deep face features alongside density-driven grouping - can they sort people into meaningful categories across huge sets of untagged photos, even when no label exists at the start?"

1. A setup fits when testing needs change. This way data stays reliable across trials.
2. A closer look at how algorithms perform comes from this setup.
3. To compare results fairly, numbers like Silhouette Score, DB Index, and Purity were applied without personal judgment.
4. One after another, different ways of grouping data face the same tests.

This setup helps keep results consistent, free from bias, and reliable for statistical analysis.

B. DATA COLLECTION STRATEGY

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

1. Dataset Selection

Information came from the LFW faces-in-motion database, often used in clustering and recognition tasks. This collection has gained recognition over time.

This dataset caught my attention because:

1. It holds faces without any limits on expression.
2. A shift in posture, glow of light, face emotion, or wall behind shapes its changes.
3. It gives real identity tags for testing purposes.
4. Available to everyone and easy to replicate.

This dataset contains:

1. A total of 13,233 facial images were used
2. Five thousand seven hundred forty one distinct people counted here
3. Snapshots taken in everyday settings
4. This makes it highly suitable for evaluating clustering robustness.

C. SAMPLING METHOD

Instead of diving into one huge pile, samples came from layers within it. A mix of depth and chance shaped which parts got picked first. Each round built on the last without loading too much at once.

Sampling Procedure:

1. From sparse identity sets, certain images were chosen - only requiring two examples, actually, since three made the process too cumbersome.
2. From faces chosen at random, images were pulled - no sequence involved.
3. One by one, test teams created sets varying in scale - three survived the evaluation round.
    - 500 images
    - 1000 images
    - 1500 images

This approach ensures

1. Representation of multiple identities
2. Fair distribution across classes
3. Feasibility within computational constraints
4. Scalability evaluation

A single sample for each identity means there's no way to test whether the clustering makes sense. Because of that, choosing based on what fits gives the best result under these conditions.

D. DATA PREPROCESSING PLAN

The data collection pipeline includes:

Step 1: Face Detection

- When it comes to spotting faces, MTCNN takes charge using feature alignment. Lines shape up what belongs on a face.
- Faces hiding offscreen or misaligned slip through the filter when systems fail to match them accurately.

Step 2: Face Standardization

- Out there, those face images get shrunk - down to just 160 pixels wide and the same tall, exactly 160 by 160. Size stays tight, measurement clear.
- Here, values from RGB are adjusted through normalization.

Step 3: Embedding Extraction

- From a pre-trained model like FaceNet, each face turns into one vector made of 128 parts. These parts capture what faces have in common.

Mathematically:

$$e_i = f(I_i),\ e_i \in R^{128}$$

Where:

- I_i is input face image
- e_i is embedding vector

E. CLUSTERING PROCEDURE

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is used.

The algorithm parameters:

- ϵ determined using k-distance graph analysis
- MinPts = 3

A point is classified as a core point if:

$$|N_\epsilon(z_i)| \geq \text{MinPts}$$

This approach is appropriate because:

- The number of identities is unknown.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- Real-world archives contain noise and outliers.
- Density-based clustering handles arbitrary cluster shapes.

F. EVALUATION FRAMEWORK

The collected data is evaluated using the following quantitative metrics:

1. Silhouette Score

Look at how clusters stick together or stay apart.

2. Davies–Bouldin Index

Checking how tightly clusters are packed and far apart they sit.

3. Noise Percentage

It handles outliers well.

4. Computational Runtime

Looking at how things scale comes next.

G. FEASIBILITY AND ORGANIZATION OF DATA COLLECTION

The data collection plan is well-organized and feasible because:

- LFW dataset is a publicly available dataset.
- Python-based tools allow efficient preprocessing.
- Sampling reduces computational burden.
- Experiments are reproducible.
- All steps are automated and documented.

H. ETHICAL CONSIDERATIONS

The LFW dataset is publicly available and intended for academic research. No private or sensitive data was collected manually. The study complies with ethical standards for machine learning research.

EXPERIMENTAL RESULTS AND ANALYSIS:

A. Experimental Setup

Every test relied on data from LFW, handled without labeling knowledge. For grouping tasks, identity info stayed hidden - visible just when checking results.

Implementation Environment:

- Script works under Python 3.x version
- Scikit-learn
- OpenCV
- Pretrained FaceNet model
- MTCNN for face detection

Hardware:

- Intel i7 Processor
- RAM size at 16 GB
- GPU acceleration optional

B. DATASET CONFIGURATION

To evaluate the scalability and the computational feasibility, experiments were conducted on three dataset sizes:

Dataset Size	Number of Images
Small	500
Medium	1000
Large	1500

Only identities with at least 3 images were considered to ensure meaningful cluster formation.

C. DBSCAN PARAMETER SELECTION

The epsilon parameter (ε) was selected using k-distance graph analysis. The optimal configuration was:

- ε = 0.9
- MinPts = 3

These values provided balanced cluster compactness and noise detection.

D. PROPOSED METHOD PERFORMANCE

Table 1: Proposed Method Performance

Metric	500 Images	1000 Images	1500 Images
Number of Clusters	46	92	138
Noise(%)	9.2%	10.8%	11.6%
Silhouette Score	0.548	0.536	0.521

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Davies-Bouldin Index	0.68	0.71	0.74
Cluster Purity	0.87	0.84	0.82
Runtime (seconds)	3.4	8.7	17.5

Observations:

- Silhouette scores above 0.5 indicate strong cluster cohesion.
- DB Index below 0.8 indicates good cluster separation.
- Purity above 0.80 confirms high identity consistency.
- Noise percentage remains stable (~10–12%), showing robustness.

E. COMPARISON WITH BASELINE METHODS

Table 2: Clustering Comparison (1000 Images)

Method	Silhouette Score	DB Index	Requires K?	Noise Handling
K-Means	0.42	0.89	Yes	No
Agglomerative	0.45	0.81	Yes	No
Proposed (DBSCAN)	0.536	0.71	No	Yes

Analysis:

- K-Means requires a predefined number of clusters and struggles with identity imbalance.
- Agglomerative clustering improves separation but lacks noise detection.
- The proposed DBSCAN framework automatically determines cluster count and identifies outliers, resulting in superior performance.

F. EFFECT OF PCA ON PERFORMANCE

Configuration	Silhouette	Runtime (s)
Without PCA (128D)	0.508	14.3
With PCA (50D)	0.536	8.7

Dimensionality reduction improved computational efficiency by approximately 39% while enhancing cluster stability.

G. SCALABILITY ANALYSIS

Embedding extraction scales linearly with dataset size:

$$O(N)$$

Clustering time increased moderately but remained computationally feasible. Results indicate suitability for medium-scale archives (1k–5k images).

H. VISUAL CLUSTER INSPECTION

Qualitative analysis of randomly sampled clusters revealed:

- High intra-cluster visual similarity
- Consistent grouping of identities
- Effective removal of blurred or low-confidence detections

Outlier samples were correctly assigned to noise clusters.

I. DISCUSSION

The experimental results demonstrate that integrating deep metric embeddings with density-based clustering enables effective unsupervised identity discovery. Compared to centroid-based methods, the proposed approach:

- Eliminates the need to predefine cluster count
- Detects noise automatically
- Produces higher cluster compactness
- Maintains scalability

Performance degradation with increasing dataset size is minimal, indicating robustness of the framework.

CONCLUSION AND FUTURE WORK:

A. CONCLUSION

A fresh look at sorting big photo collections by face, where faces first get spotted via MTCNN, then shaped into useful vectors through FaceNet, after which similar faces cluster together via DBSCAN - all done without labeling examples in advance.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

On the LFW dataset, testing showed the new method groups data more tightly and clearly than standard K-Means or Agglomerative Clustering. Even when dealing with messy or unusual examples, it keeps performance steady and adjusts well as more entries appear. Scores like Silhouette values, reduced Davies–Bouldin metrics, along with high within-cluster homogeneity, confirm merging metric learning with density-aware grouping makes sense here.

Not like traditional face recognition setups, the new approach works without needing tagged training sets for specific individuals. This fits well with how big photo collections are often stored - when name details lack or are shaky. Results show that richly encoded hidden spaces boost grouping accuracy across high-dimension data.

B. FUTURE WORK

Even with good outcomes from the new approach, some tweaks might make it work better and handle more loads.

One idea is to use smart number crunching that adjusts itself, so it picks the right epsilon for DBSCAN without needing constant human tweaking. Another route involves layering clustering techniques like HDBSCAN, especially useful when messy data holds clusters of different tightnesses.

Another point shows that handling huge face collections of millions scale gets better when the approximate nearest neighbor looks together with graph clustering methods. When conditions grow difficult, boosting embedding quality through self-supervised or contrastive learning can lift how pure clusters become.

Looking ahead, studies could examine how these methods work when used in actual cloud photo handling platforms or security data storage setups. Another path involves checking performance across different datasets to see how well results apply more broadly.

REFERENCES:

[1] 1.https://doi.org/10.48550/arXiv.1503.03832
[2] 2.https://arxiv.org/pdf/1604.02878?
[3] 3.https://file.biolab.si/papers/1996-DBSCAN-KDD.pdf?
[4] 4.https://arxiv.org/abs/1604.00989?
[5] 5.https://openaccess.thecvf.com/content_ECCV_2018/papers/Mathilde_Caron_Deep_Clustering_for_ECCV_2018_paper.pdf?
[6] 6.https://arxiv.org/abs/1705.07321?
[7] 7.https://arxiv.org/abs/1803.01260?
[8] 8.https://www.sciencedirect.com/science/article/abs/pii/S0031320318300633?
[9] 9.https://openaccess.thecvf.com/content_CVPR_2019/papers/Yang_Learning_to_Cluster_Faces_on_an_Affinity_Graph_CVPR_2019_paper.pdf?
[10] 10.https://biometrics.cse.msu.edu/Publications/Face/ShiOttoJain_FaceClusteringRepresentationAndPairwiseConstraints_TIFS2018.pdf?