Key Topics in Artificial Intelligence: Evaluating the Effectiveness of K-Means Clustering for Face Recognition

Jorrit Adema (852773086) Sander Brouwer (851807537) Ning Wei Zhou (852757814)

Open Universiteit

1. Introduction

Face recognition is a central topic in computer vision, with diverse applications ranging from security authentication to social media tagging. Traditional face recognition methods typically rely on supervised learning, requiring large labeled datasets. However, unsupervised learning techniques, such as K-Means clustering, offer a viable alternative by identifying patterns in facial features without the need for labeled data (MacQueen, 1967).

K-Means is a centroid-based clustering algorithm that groups data into clusters based on feature similarities. In the context of face recognition, K-Means can be used to categorize facial images into similar groups using vector embeddings. This approach is particularly useful when labeled data is limited or when the goal is to discover natural groupings within facial datasets.

This study investigates the effectiveness of K-Means in face recognition and addresses the following research questions:

1. **RQ1** How does the choice of k clusters influence the performance of K-Means in face recognition?
2. **RQ2** How does a recognition distance threshold impacts the K-Means model, and how can it be optimally determined?
3. **RQ3** How does the K-Means model perform on augmented images?

2. Methods

For the experiments, several videos from open-access platforms were used.

The face recognition system was implemented using high-level Python libraries, including OpenCV (OpenCV, 2025), Pandas (Pandas, 2025), Seaborn (Seaborn, 2025), MediaPipe (Google, 2025b), and PyTorch (Pytorch, 2025). Initially, video frames were extracted from the source footage using MediaPipe, forming the foundation for subsequent face detection and recognition.

Facial regions were identified using BlazeFace, a pre-trained convolutional neural network (CNN), and then converted into vector embeddings using FaceNet PyTorch with the InceptionResNetV1 architecture. These embeddings captured distinctive facial features and served as input for clustering via the K-Means algorithm (Sandberg, 2025)(davidsandberg/facenet, 2024)(Google, 2025a). The number of clusters (k) was determined empirically based on preliminary evaluation.

The K-Means model was trained on the extracted embeddings, with each cluster corresponding to a distinct identity. The model was then evaluated on previously unseen video footage, utilizing the learned embeddings for face recognition. To further assess the model's robustness, the testing video was augmented in various ways, and performance was re-evaluated.

3. Experimental Results

3.1. Datasets

The experiment was conducted using videos from the Dutch series *New Kids*, chosen for their short duration, which made them suitable for efficient processing and downloading (Wikipedia, 2024). Three videos were used for training, while a separate video was reserved for testing the model's performance. The brevity of these videos simplified the face detection and recognition tasks, making them ideal for this study. The following videos were used:

Notable is the visual resemblance of the characters in the videos. Several characters have concealing features like mustaches.

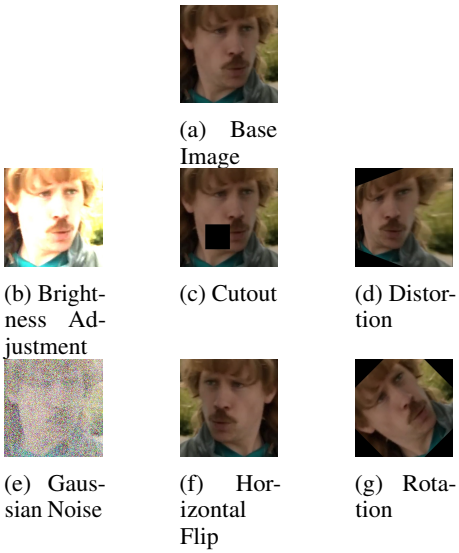
The training and test movies, as well as the intermediate images, are not included with this paper. However, all numerical data is provided. The full dataset is available upon request by contacting the authors. The training and test movies can also be found in the references or the README.md files

3.2. Implementation Details

Video frames were extracted using the OpenCV library at a sampling rate of 10 frames per second, assuming the training videos were originally recorded at 25 frames per second. This sampling rate balanced efficient frame extraction with sufficient data for model training.

Face detection was performed using a pre-trained CNN (TensorFlow's MTCNN (Zhang et al., 2016)), which is optimized for detecting facial regions, which gets saved in a separate folder named "face_folder" and "face_folder_test". The detected faces were then transformed into numerical embeddings and saved as a csv file called "results" and "test_results", which capture the most relevant facial features. These embeddings were clustered using the K-Means algorithm, with the optimal number of clusters k determined using evaluation metrics such as the Silhouette Score (Rousseeuw, 1987), Calinski-Harabasz (Caliński & Harabasz, 1974) Score, supported by visual inspection, tested for values of k ranging from 1 to 15. A visual inspection was also conducted to finalize the number of clusters, optimizing for recall and precision.

The trained K-Means model was applied to recognize faces in the test video by comparing the detected face embeddings to the centroids of the learned clusters, using various distance thresholds. To assess the model's robustness, the frames of the test video were augmented in several ways. The types of augmentations applied and their corresponding parameters are detailed in Figure 1 and Table 1, respectively. Finally, the performance of the model on the augmented test faces was evaluated and compared to its performance on the original, unaugmented images.



(a) Base Image

(b) Brightness Adjustment

(c) Cutout

(d) Distortion

(e) Gaussian Noise

(f) Horizontal Flip

(g) Rotation

Figure 1. Base Image with Applied Augmentations

Table 1. Parameter Settings for Each Augmentation Technique

Augmentation	Parameter(s)	Value(s)
Original Image	–	–
Brightness/Contrast	alpha, beta	3, 20
Cutout	size	80 px
Distortion	distortion_level	60
Gaussian Noise	Mean, Std Dev	0, 25
Horizontal Flip	flipCode	1
Rotation	angle	45°

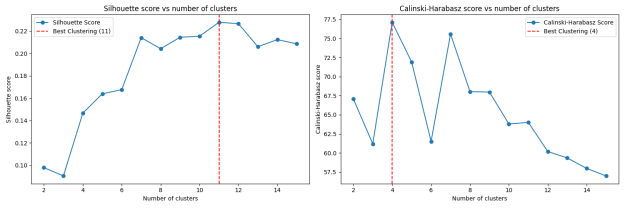


Figure 2. Silhouette and Calinski-Harabasz scores on *New Kids* videos

3.3. Results

Analyze and discuss the results.

This section presents the experimental results based on the methodology described earlier. The experiments are structured to address the research questions outlined in the introduction. Each experiment is analyzed to provide insights into the effectiveness of the K-Means model for face recognition, as well as its performance under augmented conditions.

3.3.1. Choice of k

Figure 2 illustrates the Silhouette and Calinski-Harabasz scores for the K-Means model trained on *New Kids* faces. Notably, the two evaluation metrics provide different optimal cluster numbers: the Silhouette Score suggests $k = 11$, while the Calinski-Harabasz Score recommends $k = 4$.

When visually inspecting the clustering results for $k = 4$, it was observed that clusters were dominated by characters with the highest number of embeddings. This suggests that the distribution of embeddings significantly influenced the clustering outcome at this k-value. A side effect of this dominance was reduced precision, as many non-corresponding characters were assigned to the dominant cluster containing the most frequently represented characters as shown in table 2.

In contrast, increasing the numbers of clusters to $k = 11$, more characters were assigned their own distinct clusters, leading to an increase in recognition precision as shown in table 2. However, this finer granularity also introduced a degree of overfitting: some characters are divided over multiple clusters corresponding to different videos. The model's robustness can be enhanced by merging clusters that correspond to the same character, which has been shown

to improve both precision and recall.

For the sake of simplicity, it was decided that the subsequent experiments would be performed with $k = 4$, as this value focuses on the main characters of the training set.

Table 2. Summary of Clustering Results for Different K Values on Training Data

Configuration	Average Precision	Average Recall	Average F1-Score
K = 4	0.76606	0.952211	0.849052
K = 11 (Detailed Clusters)	0.795046	0.710995	0.750675
K = 11 (Merged Option One)	0.872303	0.932871	0.901571
K = 11 (Merged Option Two)	0.86694	0.941617	0.902736

3.3.2. Recognition Distance Threshold

Since the K-Means clustering algorithm assigns every embedding to a cluster, it becomes necessary to implement a distance-based threshold to handle outlier embeddings—those that do not truly correspond to any specific character's cluster. This helps to avoid misclassifications by filtering embeddings that are too distant from cluster centroids. Two main strategies can be used to apply such a threshold:

1. Recognition Distance Threshold During Training To ensure clean and well-defined clusters, a distance threshold can be applied during the training phase. The process involves:

1. Fitting the K-Means model on the training embeddings.
2. Calculating the distance of each embedding to its assigned cluster centroid.
3. Removing embeddings that are deemed outliers based on a chosen distance threshold.
4. Refitting the K-Means model on the cleaned data.

This approach enhances the precision of the model by eliminating noisy samples. However, it may lead to a drop in recall if valid embeddings—especially those belonging to characters with more variability—are excluded due to high distances from their respective centroids.

Several statistical techniques can be used to determine the distance threshold for detecting outliers. These include:

- **Interquartile Range (IQR)** (Hoaglin et al., 1986): Embeddings with distances falling outside the range:

$$[Q_1 - 1.5 \times IQR,\ Q_3 + 1.5 \times IQR]$$

where $IQR = Q_3 - Q_1$, are treated as outliers.

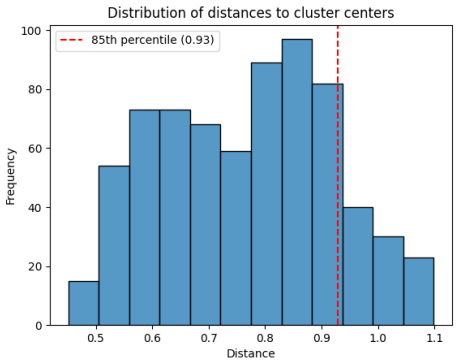


Figure 3. Clustering Distances on Trainingsdata for $k = 4$

- **Z-Score Method** (Kreyszig et al., 2011): Embedding Distance with a z-score greater than a threshold t, calculated as:

$$z_i = \frac{x_i - \mu}{\sigma}$$

where μ and σ are the mean and standard deviation of the distances, respectively.

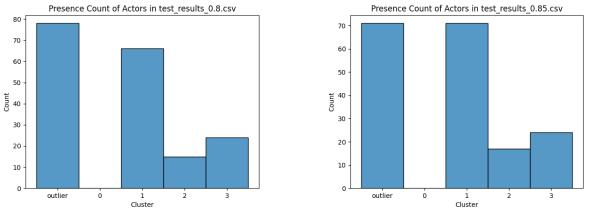
- **Percentile Method**: Embeddings with distances above a selected percentile of the training distances (e.g., 85th percentile) are marked as outliers.

2. Recognition Distance Threshold During Inference Alternatively, the threshold can be applied at inference time, after the K-Means model has been trained. In this method, if a test embedding lies beyond a certain distance from all cluster centroids, it is labeled as an outlier and left unclassified. This method does not affect model training but helps filter out false positives during recognition.

In preliminary experiments using the Interquartile Range (IQR) and Z-Score methods for $k = 4$, no outliers were detected in either the training or test datasets. However, based on the clustering results (as summarized in Table 2), outliers clearly exist. Therefore, the Percentile Method was chosen as the most suitable approach.

Figure 4 illustrates the detection of outliers for different distance thresholds using the Percentile Method. Threshold values ranging from 0.80 to 0.95 correspond to filtering out embeddings whose distances exceed the 80th to 95th percentile of training distances. As shown, increasing the threshold reduces the number of detected outliers. Interestingly, applying a higher threshold allowed the model to detect an additional cluster in the test data.

Based on thorough visual inspection and quantitative analysis, a threshold value of 0.85 was selected as the optimal trade-off between outlier filtering and classification sensitivity. This threshold was applied throughout the remaining experiments and directly contributed to the results illustrated in Figure 5.



(a) Distance Threshold 0.8 (b) Distance Threshold 0.85

(c) Distance Threshold 0.9 (d) Distance Threshold 0.95

Figure 4. Cluster detection results for various distance thresholds

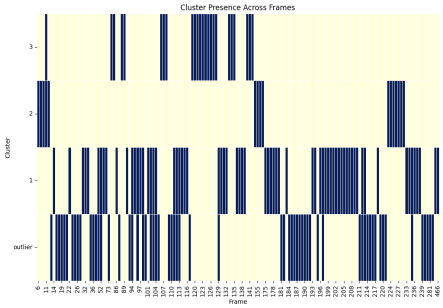


Figure 5. Result of Test Data for $k = 4$ and $threshold = 0.85$

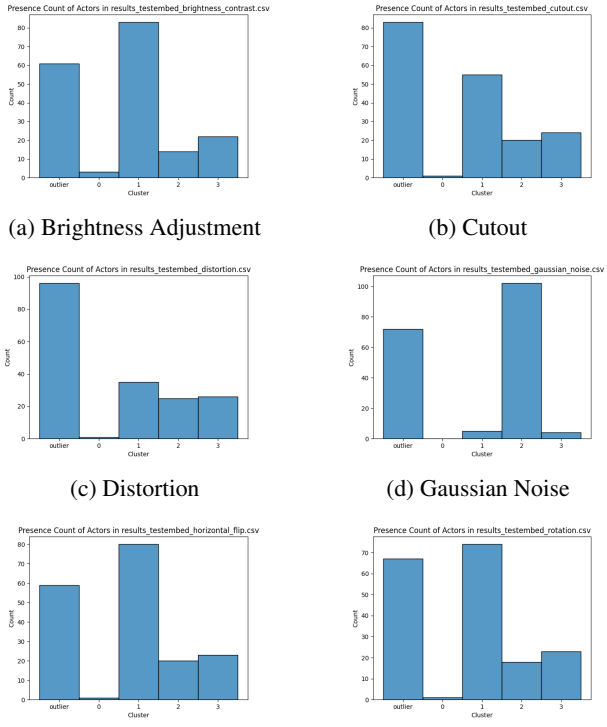
3.3.3. Performance on Augmentation

Figure 6 presents the performance of the K-Means model under various augmentation techniques, compared to its baseline performance on the original dataset shown in Figures 4b and 5. While the base image demonstrates relatively well-separated clusters with high counts in clusters 1, 2 and 3, the introduction of augmentations results in varying degrees of performance degradation.

Augmentations such as brightness adjustment and cutout (Figures 6a and 6b) show a moderate shift in the distribution of embeddings across clusters, indicating some impact on feature stability.

More aggressive transformations, such as distortion and Gaussian noise (Figures 6c and 6d), lead to a notable dispersion of embeddings and higher counts in the outlier cluster suggesting reduced model robustness. Similarly, spatial augmentations like horizontal flip and rotation (Figures 6e and 6f) also show mild to moderate performance drops.

Overall, the K-Means model exhibits some resilience to basic augmentations, but struggles with more disruptive transformations, indicating a need for enhanced generalization strategies or augmentation-aware clustering approaches.



(a) Brightness Adjustment (b) Cutout

(c) Distortion (d) Gaussian Noise

(e) Horizontal Flip (f) Rotation

Figure 6. Cluster detection results for various augmentations for $k = 4$ and $threshold = 0.85$

3.4. Discussion

In finalizing this study, we must first acknowledge several limitations before revisiting the research questions stated in the introduction.

Time constraints were a significant factor—the research was conducted over approximately two weeks, which limited the depth of exploration. Some obvious and important questions were not fully addressed. For example, while unsupervised learning was investigated, supervised learning should also have been considered to allow for a meaningful comparison between the two approaches. Additionally, the algorithms used for face detection and recognition were not analyzed in detail; with further study, these could have been optimized. Lastly, only a single dataset was used, consisting of the same set and type of characters, which may have limited the generalizability of the findings.

Despite these limitations, some observations can still be made regarding each of the three research questions.

The first question concerned the influence of the number of clusters on the performance of the classifier. It was found that the clusters were dominated by the main characters in the training set. When the number of clusters was low, this resulted in lower precision. Increasing the number of clusters improved precision but also led to characters being assigned to multiple clusters.

For the second research question, the influence and optimization of a recognition threshold were investigated. Applying a threshold to the training set helped produce purer clusters and improved precision. When applied to the test set, the threshold was effective at excluding previously unseen characters.

Lastly, the effect of image augmentation was examined. K-Means appeared to be resilient to basic augmentations but was negatively affected by operations such as distortion and Gaussian noise. In the end, some visual inspection was required for optimal tuning, which is contradictory to unsupervised learning.

3.5. Citing references

References

Alkohol. New kids: Fußballspiel, Feb. 2021. URL https://www.youtube.com/watch?v=DxXC9s3AROU. Used as Training Video [Accessed: 2025-04-13].

Batsjongu. New kids alfabet, Apr. 2021. URL https://www.youtube.com/watch?v=n2hZ2Y5dRHg. Used as Training Video [Accessed: 2025-04-13].

Caliński, T. and Harabasz, J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27, 1974. [Accessed: 2025-04-13].

davidsandberg/facenet. incepresnet, 2024. https://github.com/timesler/facenet-pytorch [Accessed: 2025-04-13].

Google. Blazeface, 2025a. https://ai.google.dev/edge/mediapipe/solutions/vision/face_detector [Accessed: 2025-04-15].

Google. Mediapipe solutions guide, 2025b. https://ai.google.dev/edge/mediapipe/solutions/guide [Accessed: 2025-04-13].

Hoaglin, D. C., Iglewicz, B., and and, J. W. T. Performance of some resistant rules for outlier labeling. *Journal of the American Statistical Association*, 81(396): 991–999, 1986. doi: 10.1080/01621459.1986.10478363. https://www.tandfonline.com/doi/abs/10.1080/01621459.1986.10478363 [Accessed 2025-04-14].

Jonatan. New kids nitro, "peter lemonade!", May. 2012. URL https://www.youtube.com/watch?v=VYgKTaAZi7Y. Used as Training Video [Accessed: 2025-04-13].

Klink, R. New kids turbo: Tankstation, Mrt. 2022. URL https://www.youtube.com/watch?v=PGUi6uoacEE. Used as Training Video [Accessed: 2025-04-13].

Kreyszig, E., Kreyszig, H., and Norminton, E. J. *Advanced Engineering Mathematics*. Wiley, Hoboken, NJ, tenth edition, 2011. ISBN 0470458364. [Accessed: 2025-04-12].

MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pp. 281–298. University of California press, 1967. [Accessed: 2025-04-8].

Nederland, P. V. New kids, maar ze zeggen alleen k*t — new kids turbo — prime video nl, Mrt. 2021. URL https://www.youtube.com/watch?v=3kBXtOqQoEM&t=1s. Used as Test Video [Accessed: 2025-04-13].

OpenCV. Opencv, 2025. https://opencv.org/ [Accessed: 2025-04-13].

Pandas. Pandas, 2025. https://pandas.pydata.org/ [Accessed: 2025-04-15].

Pytorch. Pytorch, 2025. https://pytorch.org// [Accessed: 2025-04-15].

Rousseeuw, P. J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. [Accessed: 2025-04-12].

Sandberg, D. Facenet, 2025. https://github.com/davidsandberg/facenet [Accessed: 2025-04-15].

Seaborn. Seaborn, 2025. https://seaborn.pydata.org/ [Accessed: 2025-04-15].

Wikipedia. New kids, 2024. https://en.wikipedia.org/wiki/New_Kids [Accessed: 2025-04-13].

Zhang, K., Zhang, Z., Li, Z., and Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23 (10):1499–1503, October 2016. ISSN 1558-2361. doi: 10.1109/lsp.2016.2603342. URL http://dx.doi.org/10.1109/LSP.2016.2603342. [Accessed: 2025-04-14].