European Research Studies Journal
Volume XXVII, Special Issue 2, 2024

pp. 3-13

A Real-Time Autonomous Machine Learning System for Face Recognition Using Pre-Trained Convolutional Neural Networks

Submitted 18/02/24, 1st revision 16/03/24, 2nd revision 20/04/24, accepted 16/05/24

Michał Maj[1], Jacek Korzeniak[2], Józef Stokłosa[3], Paweł Barwiak[4], Bartłomiej Bartnik[5], Łukasz Maciura[6]

Abstract:

Purpose: *This paper aims to present a novel real-time, autonomous machine learning system for face recognition. This system employs pre-trained convolutional neural networks for encoding facial images and applies a Naive Multinomial Bayes model for autonomous learning and real-time classification.*
Design/Methodology/Approach: *The system leverages a pre-trained ResNet50 model to encode facial images from a camera, while cognitive tracking agents collaborate with machine learning models to monitor the faces of multiple people. A novelty detection algorithm based on a Support Vector Machine (SVM) classifier checks whether a detected face is new or already recognized. The system autonomously starts the learning process if an unrecognized face is identified. Real-time classification of individuals relies on a Naive Multinomial Bayes model, with special agents tracking each face.*
Findings: *Experiments demonstrated that the system can accurately learn new faces appearing within the camera frame in favorable conditions. The key determinant of successful recognition and learning is the novelty detection algorithm, which, if it fails, may assign multiple identities or group new individuals into existing clusters.*
Practical Implications: *This system offers a practical solution for real-time, autonomous face recognition, with potential applications in security, access control, and personalized services. Its ability to quickly learn new faces while maintaining classification accuracy ensures adaptability in dynamic environments.*
Originality/Value: *The research introduces an innovative approach by combining pre-trained neural networks with autonomous learning and a novelty detection algorithm to classify faces in real-time. This hybrid method ensures rapid and accurate face recognition while minimizing the need for extensive training data or prolonged training times.*

[1]*Corresponding Author: Netrix/ WSEI University , Lublin, Poland, e-mail: michal.maj@netrix.com.pl;*
[2]*WSEI University, Lublin, Poland, e-mail: jacek.korzeniak@wsei.lublin.pl;*
[3]*WSEI University, Lublin, Poland, e-mail: jozef.stokłosa@wsei.lublin.pl;*
[4]*WSEI University, Lublin, Poland, e-mail: paweł.barwiak@wsei.lublin.pl;*
[5]*Wyższa Szkoła Biznesu - National Louis University, e-mail: bbartnik@wsb-nlu.edu.pl;*
[6]*Netrix , Lublin, Poland, e-mail: lukasz.maciura@netrix.com.pl;*

Keywords: *Real-Time Face Recognition, Autonomous Systems, Machine Learning, Convolutional Neural Networks, Deep Learning, Computer Vision, Artificial Intelligence*.

JEL codes: *C45, C61, C84, L11, L86, O33.*.

Paper type: *Research article.*

1. Introduction

The system can learn unsupervised to recognize speakers based on microphone or audio file data, so it does not require any training data. The system checks whether a given signal belongs to speech for each time window. Then, after the previous condition is met, the system checks whether the signal comes from a new (unknown) speaker. If so, automatic learning occurs, and a new identifier is assigned; if not, the previously trained face is recognized.

The most important output of the system is the identifiers of newly recognized speakers, which can be integrated with other systems (e.g., super-system - robot or intelligent chat - the bot can ask the newly recognized user for the name and surname).

2. Literature Review

Object detection using feature-based cascade classifiers is a practical object detection method proposed by Paul Viola and Michael Jones in the article "Rapid Object Detection using a Boosted Cascade of Simple Features." (Viola and Jones, 2001). It is a machine learning-based approach where a cascading feature is trained from multiple positive and negative images. It is then used to detect objects in other images. So, to train the classifier, the algorithm needs many positive images (face images) and negative images (non-face images).

Then, you need to extract the features from it. So, each kernel's possible sizes and locations are used to compute multiple functions. For each feature calculation, find the sum of the pixels under the white and black rectangles. To solve this problem, integral imaging was introduced. Regardless of the image size, it reduces the computation for a given pixel to an operation involving only four pixels.

From a mathematical point of view, the description presented above may resemble wavelets in some way. In particular, it can be seen that they show some similarity to Haar basis functions and Haar wavelets (Pandey *et al.*, 2022). As mentioned earlier, to get the features for each of these five rectangular areas, you need to subtract the sum of the pixels in the white area from the sum of the pixels in the black area. It is worth noting that these features are of fundamental importance in

the context of face detection:

- The eye area is usually darker than the cheek area;
- The nose area is lighter than the eye area.

Therefore, given these five rectangular regions and their corresponding difference sums, a set of features can be created to classify facial parts. Then, for the entire dataset of features, you can use the AdaBoost algorithm (Ding *et al.*, 2022) to select which ones correspond to the facial regions of the image.

However, it should be noted that using a fixed-size kernel and moving it over each (x, y) coordinate of the image and then computing these features along with performing the actual classification can be computationally expensive.

Therefore, the concept of cascades or stages was introduced. At each stop along the sliding window's path, the window must undergo a series of tests, with each subsequent test being more computationally expensive than the previous one. If any test fails, the window is automatically dismissed. This involves using integral images (also called summary area Tables). Thanks to the AdaBoost algorithm, they are also very efficient in feature selection (Masnadi-Shirazi and Masnadi-Shirazi, 2018).

Detecting facial landmarks is one of the most essential topics in computer vision, and only in the last few years have fast and accurate algorithms been developed. The algorithms also include data that consists of thousands of images with manually marked landmarks. This data is divided into training (80%) and test (20%) sets. The training set is used to train the machine learning model, and the test set is used to test the model's accuracy.

A machine learning model is only good if it works well on new data, so you shouldn't mix training and test data. So we can assume that $n = \text{Total number of images}$ in the training set. For the training set $n = 3000$ $p = \text{Total number of landmarks}$. The standard Dlib model has 68 landmarks, so $p = 68$. The image in the training set is the $I_i = i\text{th}$ image in the set.

Whereas $S_i = [x_1, y_1, x_2, y_2, \ldots, x_n, y_n]^T$ is the shape associated with the image, therefore (x_k, y_k) is kth landmark. Transpose (T) transforms a row vector into a column vector. According to the above, we can say that the training set consists of n images and n corresponding shapes (Kim *et al.*, 2021).

In the traditional image classification application, an image is converted into a feature vector (or equivalent point) in a higher-dimensional space. This was done by computing a feature descriptor (e.g., HOG) for the image elements (Bakheet and Al-Hamadi, 2021). Once the image was represented as a point in a higher-dimensional space, a learning algorithm such as SVM could be used to partition the

space using hyperplanes that separated points representing different classes (Sharma *et al.,* 2022). Like most architectures built on the principle of convolutional neural networks, ResNet is composed of convolutional layers followed by a Fully Connected layer (Maj *et al.,* 2023). Convolutional layers build a feature vector in a higher-dimensional space, similar to the HOG descriptor.

Some notable algorithms such as Eigen Faces, Fischer Faces, and LBPH attempt to apply mathematical tricks (such as PCA, LDA, histograms, etc.) to represent the face in a more compact form, extracting the most useful information (features) from the face and stripping away unnecessary information (Wu and Ji, 2019).

With the advent of deep learning, these hand-crafted features took a back seat, and researchers turned these tasks over to convolutional neural networks (Pham *et al.,* 2022). Facial recognition is a category of biometric software that mathematically maps a person's facial features and stores the data as a facial print.

The software uses deep learning algorithms to compare live captures or digital images with a stored facial print to verify a person's identity (Kim *et al.,* 2021).

3. Research Methodology

The system's main element is the face detector (face-detection-retail-0004 from the depth library available on the OAK1 device). The encoder is a previously trained ResNet50 convolutional network with the classifier removed (uploaded to the OAK1 device).

Multinomial Naive Bayes classifier with the possibility of incremental learning. A cognitive agent with a state machine that autonomously controls the learning process (Rojas-Perez and Martinez-Carranza, 2023).

Face detection and encoding are implemented via a pipeline on the OAK1 device face-detection-retail-0004 from the depth library (one of the built-in models of the OAK1 device) was used as the face detector. The frames surrounding the faces are then transformed to replace the rectangular frames with square frames (the longer side is shortened on both sides).

A previously trained ResNet50 network on the VggFace2 set with the classifier removed, uploaded to the OAK1 device, is used as an encoder. For each frame, the OAK1 device returns an image, windows surrounding the face, and encoded vectors to the client on the computer (Capodiferro and Mazzei, 2023).

The Multinomial Naive Bayes Bayesian model (implemented in the sci-kit-learn library) was used as a face classifier, which is highly effective, fast in training and inference, and also allows for incremental learning thanks to the partial_fit function (Chebil *et al.,* 2023). The input to the classifier is code vectors for encoded faces.

To train the model using a new identifier, all possible person identifiers are set in the model at the beginning (e.g., from 0 to 999) - then the model can accommodate a maximum of 1000 people.

Before using the model in an autonomous system, the initial version of the model is established by training it with a certain number of classes (105 celebrities) so that it is then possible to detect novelties based on the classifier's responses.

The cognitive agent controlling the autonomous learning process is the main element of the system (Maciura *et al.*, 2023). In addition to tracking separate faces (it is activated when a face appears in a place where it was not), it has the following states (Figure 1):

0. Novelty detection state: in this state, the novelty detection algorithm works, which determines based on N_0=10 samples whether a given face is new. If newness is detected, the agent moves from state 1 to state 3.

1. Training initiation state: The vectors (encoded facial images) remembered since the agent started are used to train the model using the new class. In this state, a new class label is assigned. After completing the training initialization, the agent moves to state 2.

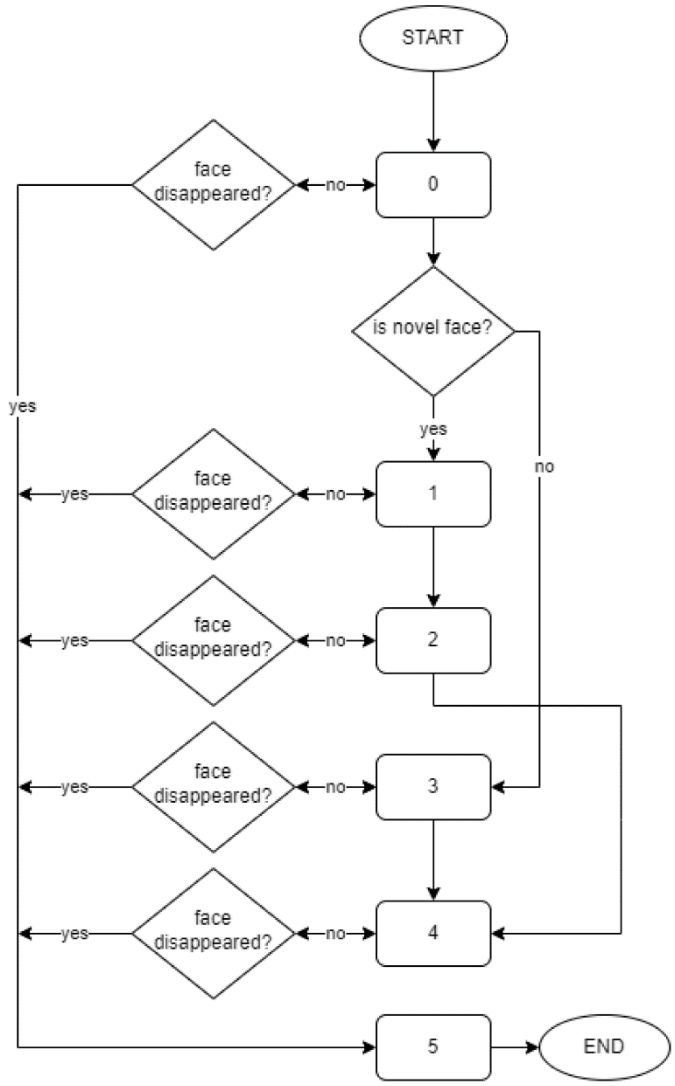
2. Continue training state: This state uses new samples to train the given class using the label obtained in the previous state. After training with N_1 = 10 samples, the agent moves to state 4 (since the label is now known, state three can be omitted).

3. Recognition initiation state: the agent predicts an existing (not new) class using the dominant label determined from the set of samples remembered in this state (at least N_2 = 5). If most predictions have the same label, then this label is saved as recognized, and the agent moves to state 4.

4. Analyzing state: the agent analyzes subsequent predictions of a known face, and when these predictions are different from the correct label determined in the previous state, the model is updated using this incorrectly recognized sample (thanks to which the model automatically improves in identifying a given person).

5. Delete state: The agent transitions to this state from any other state if the tracked face disappears after N_3=10 frames without a face being detected near the last position of the face the agent tracks. An agent in this state may be inactive and removed from the agent list.

Figure 1. *Agent state machine diagram*



Source: *Own creation.*

4. Research Results and Discussion

The system's main elements are Encoder in the form of a previously trained ResNet50 convolutional network with the classifier removed, the input for which is mel—a spectrogram from the signal time window. DTLN signal denoiser trained on data with a sampling frequency of 20050. Bernoulli Naive Bayes classifier with the possibility of incremental learning.

Novelty detector testing whether a given signal is speech or not. A cognitive agent

with a state machine that autonomously controls the learning process.

The DTLN denoiser was used in the system because various noises (e.g., the sound of computer fans) combined with voice cause disruptions to the novelty detection algorithm (e.g., a known voice combined with fan noise may be recognized as new) - the use of the denoiser eliminates this phenomenon.

The signal is first denoised using a DTLN denoiser trained on a signal with a sampling frequency of 22050 (this sampling frequency was used in all models used in this system) on its data created from several data sets. The DTLN model has been modified compared to the original version so that, among others, the number of LSTM cells in layers and the encoder size have been doubled.

The sound signal is divided into time windows from which mel - spectrograms (images) are calculated. These are the inputs to the ResNet50 network, and the classifier was removed to obtain a vector of codes for each time window. The ResNet50 (Maj *et al.*, 2022).

The network was previously trained on speech signals (processed in a similar way) so that appropriate structures were developed in its convolutional layers to encode speech signals properly. The input of the ResNet50 network is images of me-spectrograms with a size of 256 x 256 x 3, and the output is x-element codes of 8192.

The Bernoulli Naive Bayes classifier, which can incrementally train the model, was used as the primary classifier for recognizing existing speakers in the system and autonomously learning new speakers. Its input is 8192-element vectors of encoded time windows. The classifier is first trained with 138 base speakers (a certain number of base speakers is necessary for the correct operation of the novelty detection algorithm, which is based on the outputs from the current model).

When training the model with base speakers, the model is prepared for subsequent training with new speakers, up to a maximum of 1000 speakers (including the base ones) - the maximum number of classes determines the size of the model, which will no longer change when updating the model with new speakers. The model updating process is controlled by a cognitive agent, which will be described later.

The role of the novelty detector is played by the Gaussian Naive Bayes model, which takes as input the processed outputs in logarithmic form from the primary classifier (discussed on the previous slide) in response to a given sample: the 25 highest answers (after sorting them) are transferred to the model which determines whether a given answer comes from a new sample or not.

The process of training the novelty detection model is as follows: out of 277 speaker classes, 138 are used to train the BNB base model (discussed in the

previous slide), and the rest are used as examples of novelty data. The data is appropriately encoded to obtain inputs to the classifier. After training the base model of the BNB classifier, the inference was made on non-new and new samples, the answers were processed as in the point above, and the appropriate class labels were added (0 - not new, 1 - new) as a result of which a set of training data was created, which was used to train the model Gaussian Naive Bayes novelty detection.

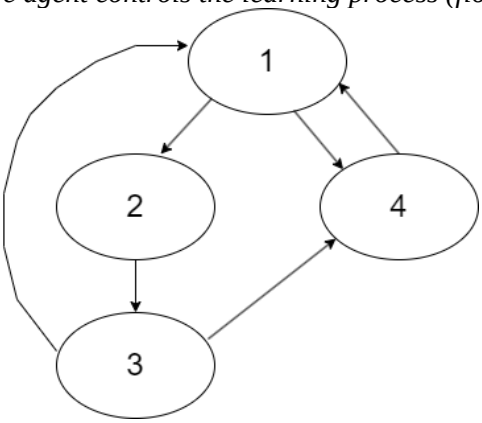
Since the novelty detector described a moment ago can only operate on a speech signal, it was necessary to check whether a given signal sample (time window) is a speech signal. The detector testing whether a given signal contains speech (classifier consisting of two classes) is based on the SGD model as input: time windows encoded into 8192 element vectors (as in most other system elements). The detector was trained on its own data set generated based on two publicly available data sets:

1. A dataset of Polish speakers.
2. A collection of various sounds (e.g., noise, musical instruments, equipment, etc.).

The main element of the system is a cognitive agent with a state machine controlling the learning process, which contains the following states:

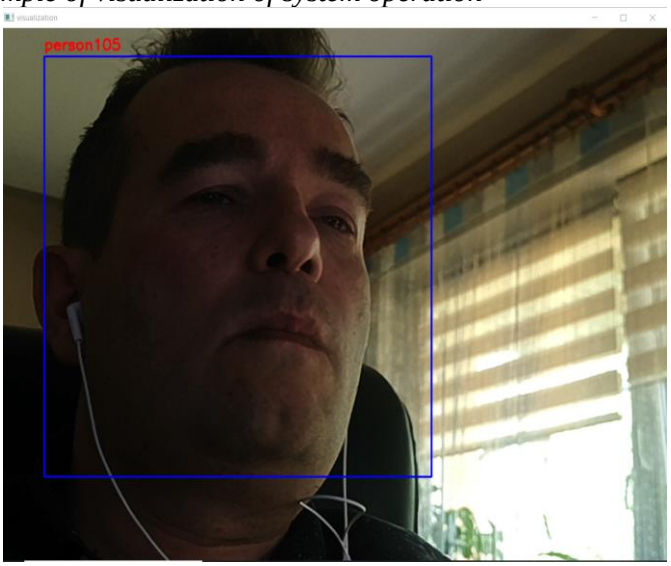
1. Detection state - based on the vector representing the time window, average results from detectors (speech and novelty) are calculated and checked in a queue of 10 consecutive samples (the queue is moved by 1 sample/time window). If the examined signal is speech, then after checking the detection of novelty, when the speaker is new, the system goes to state 2 (initial training), and when the speaker is old, to state 4 (recognition). When moving to state 2, the system also collects valid pre-training samples (from 1 speaker). If the signal is not speech, the agent remains in state 1 (detection).
2. Pre-training state: In this state, the system creates a new speaker ID and trains the model using the correct samples collected in the detection state. Then, the system moves to state 2 (continuation of training).
3. Training continuation state—in which the system continues training using continuously collected samples until a certain number of samples are collected (in this case, the agent moves to state 4—recognition) or when one of the samples turns out to be not from a speech signal or from a new speaker (in this case, the agent goes to state 1—detection).
4. Recognition state –the speaker in the queue is recognized based on several samples. When the sample turns out not to be speech or the novelty detector is activated, the agent goes to state 1.

Figure 2. *The cognitive agent controls the learning process (flow diagram)*



Source: *Own creation.*

Figure 3. *Example of visualization of system operation*



Source: *Own creation.*

Figure 5 shows the cognitive agent controls the learning process (flow diagram) and figure 3 presents the result of the visualization.

5. Conclusions, Proposals, Recommendations

A study of an independent speaker recognition system was collected using voice and image data, which has implications and can be used for applications. An approach that integrates information on speech and visual issues opens new perspectives in speaker recognition, especially in the context of the need to use biometric technologies.

The systems available for speaker recognition have a range of applications, including security, access authorization, monitoring, and personalization with electronic access—User-driven useability and an approach that opens the way to more transparent and secure user interfaces.

The consequences of applying the results continue to be the challenges that follow from research and development. In addition, the system's extension to variable environmental conditions must be taken into account, in addition to exemptions regarding data quality and against an attack intended to deceive the system.

An accessible, independent speaker recognition system based on voice and image data is an acceptance that has the potential to alternative how we identify and authorize users. Further research and innovation in this category may be available in more secure, accessible, and convenient systems.

References:

Bakheet, S., Al-Hamadi, A. 2021. A framework for instantaneous driver drowsiness detection based on improved HOG features and naïve Bayesian classification. Brain Sciences, 11(2). https://doi.org/10.3390/brainsci11020240.

Capodiferro, C., Mazzei, M. 2023. Applications of deep learning and artificial intelligence methods to smart edge devices and stereo cameras. 2023 8th International Conference on Smart and Sustainable Technologies, SpliTech 2023. https://doi.org/10.23919/SpliTech58164.2023.10193298.

Chebil, W., Wedyan, M., Alazab, M., Alturki, R., Elshaweesh, O. 2023. Improving Semantic Information Retrieval Using Multinomial Naive Bayes Classifier and Bayesian Networks. Information (Switzerland), 14(5). https://doi.org/10.3390/info14050272.

Ding, Y., Zhu, H., Chen, R., Li, R. 2022. An Efficient AdaBoost Algorithm with the Multiple Thresholds Classification. Applied Sciences (Switzerland), 12(12). https://doi.org/10.3390/app12125872.

Kim, T., Mok, J., Lee, E. 2021. Detecting facial regions and landmarks at once via a deep network. Sensors, 21(16). https://doi.org/10.3390/s21165360.

Maciura, Ł., Cieplak, T., Pliszczuk, D., Maj, M., Rymarczyk, T. 2023. Autonomous Face Classification Online Self-Training System Using Pretrained ResNet50 and Multinomial Naïve Bayes. Sensors, 23(12), 5554. https://doi.org/10.3390/s23125554.

Maj, M., Rymarczyk, T., Cieplak, T., Pliszczuk, D. 2022. Deep learning model optimization for faster inference using multi-task learning for embedded systems. Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM. https://doi.org/10.1145/3495243.3558274.

Maj, M., Rymarczyk, T., Maciura, Ł., Cieplak, T., Pliszczuk, D. 2023. Cross-Modal Perception for Customer Service. Proceedings of the 29th Annual International Conference on Mobile Computing and Networking. https://doi.org/10.1145/3570361.3615751.

Masnadi-shirazi, H., Masnadi-shirazi, H. 2018. AdaBoost Face Detection. Computer Engineering.

Pandey, A., Choudhary, D., Agarwal, R., Shrivastava, T.K. 2022. Face detection using Haar cascade classifier. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4157631.

Pham, A.H.T., Pham, D.X., Thalassinos, E.I., Le, A.H. 2022. The application of sem–neural network method to determine the factors affecting the intention to use online banking services in Vietnam. Sustainability, 14(10), 6021.

Rojas-Perez, L.O., Martinez-Carranza, J. 2023. DeepPilot4Pose: a fast pose localisation for MAV indoor flight using the OAK-D camera. Journal of Real-Time Image Processing, 20(1). https://doi.org/10.1007/s11554-023-01259-x.

Sharma, S., Raja, L., Bhatnagar, V., Sharma, D., Bhagirath, S.N., Poonia, R.C. 2022. Hybrid HOG-SVM encrypted face detection and recognition model. Journal of Discrete Mathematical Sciences and Cryptography, 25(1). https://doi.org/10.1080/09720529.2021.2014141.

Viola, P., Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1. https://doi.org/10.1109/cvpr.2001.990517.

Wu, Y., Ji, Q. 2019. Facial Landmark Detection: A Literature Survey. International Journal of Computer Vision, 127(2). https://doi.org/10.1007/s11263-018-1097-z.