CCC Publications

Identification of Facial Features in Face Templates Using Deep Neural Network Models

M. Malickas, P. Stefanovič, S. Ramanauskaitė

Matas Malickas
Neurotechnology
Laisvės pr. 125A, LT-06118, Vilnius, Lithuania
matas@neurotechnology.com

Pavel Stefanovič*
Department of Information Systems, Vilnius Gediminas Technical University
Saulėtekio al. 11, LT-10223 Vilnius, Lithuania
pavel.stefanovic@vilniustech.lt
*Corresponding author: pavel.stefanovic@vilniustech.lt

Simona Ramanauskaitė
Department of Information Technology, Vilnius Gediminas Technical University
Saulėtekio al. 11, LT-10223 Vilnius, Lithuania
simona.ramanauskaite@stud.vilniustech.lt

Abstract

In this research, facial features are identified from face templates using deep neural network models. Facial templates are widely used in modern biometric systems, enabling efficient and reliable individual recognition. Such templates are compact, easy to process, and are often used in real-time systems. During the research, a dataset with 115,023 items (face templates), used by the Neurotechnology company to identify persons, was prepared as a new dataset. The dataset has been prepared to study three facial features from face templates: gender, race, and age. The facial templates are lighter-weight than real photos and could be reused to estimate the facial features. However, it is not yet known how effective facial feature identification will be, as some data are lost when converting a photo into a face template. Several methods have been proposed for classifying original image attributes solely from face templates. Both deep neural networks and classical machine learning algorithms were used in the experiments. The experiments revealed that gender is the most accurately predictable attribute, with the best model achieving 93 % accuracy. Classification of people's race and age from face templates is more challenging, likely because face template generation models are designed to eliminate information unrelated to identity. A comparison of machine learning methods showed that deep neural networks are better suited to this task than classical classification algorithms.

Keywords: face templates, face recognition, deep learning, machine learning, hyperparameter optimization.

1   Introduction

With the rapid growth of artificial intelligence and data processing technologies over the past few decades, computer vision has emerged as one of the most significant research and application areas in information technology. Image recognition systems are widely used in both academic and industrial environments, from medical diagnostics [5, 19] and autonomous vehicles [4, 9] to security systems [11], industrial manufacturing [13], agriculture [2], social media analysis [15, 24], and even the arts [25]. The goal of these systems is to automatically interpret and classify image content, typically using deep learning (DL) methods. DL, especially convolutional neural networks (CNNs), has revolutionized image analysis [6]. CNNs allow computers to learn to recognize visual features in raw data. During model training, the system learns to identify which information is relevant to solving a specific task. This eliminates the need to manually create descriptive features. The versatility of these methods has enabled their application to a range of problems in computer vision, including object detection, classification, segmentation, and facial analysis. One of the most significant applications of computer vision is face recognition, along with the analysis of facial features related to it. Face recognition is the process by which a system identifies or authenticates a person based on their facial features. Such systems are widely used in security, access control, biometric authentication, and law enforcement. Facial attribute recognition involves determining additional semantic features, such as gender, age, emotion, race, and smile [12]. Such features can help improve understanding of the context of a given situation, support user analysis, and enable the personalization of various services. Most often, facial attributes are determined directly from images using convolutional neural networks trained to classify specific features in full-face photos. However, in biometric systems, it is common to transform a face into a lower-dimensional face template, which is a fixed-length vector representation – the so-called embedding, feature vector – generated by a neural network. These templates are commonly used for identity verification. Identity verification is performed by comparing facial feature vectors; therefore, the aim is to make them unbiased and free of redundant information about the person, except for identity. Recent research suggests that facial feature vectors may contain additional information, including gender or race. This means that facial features can be extracted from face templates, raising concerns about security and confidentiality. Kotwal and Marcel [3] examine the extent to which facial feature information is retained in these templates and whether it can be extracted using specially trained deep learning models.

This work investigates whether it is possible to classify human facial attributes, such as gender, age, and race, from a face template alone, without the original face image. The use of face templates for facial attribute estimation can enhance user privacy in a multi-level system architecture. Using face templates can improve facial attribute estimation performance by reducing the dimensionality of input data. Such studies not only reveal the properties of templates but also help assess the information that can be recovered from them without access to the original data. An experimental investigation into the efficiency of various deep learning architectures that accept a template as input and output facial attributes has been conducted. The experiments were conducted using Neurotechnology's private dataset.

The main contributions of this research are as follows:

1. During the experiments, the influence of hyperparameters has been analyzed. It enables us to identify which parameters have the most significant impact on classification performance. The results can help other researchers save time by identifying better parameter settings in different cases. Also, different deep learning architectures have been used.

2. Additionally, the results have been compared with traditional machine learning models, such as random forests, support vector machines (SVM), and decision trees. These two algorithms have been selected for their high performance.

3. The main contribution of this work is to establish empirical evidence and analytical benchmarks for demographic inference from facial templates, with implications for both biometric system design and privacy risk assessment.

The structure of this paper is as follows. In Section 2, related work on facial recognition models and facial feature extraction is reviewed. The experimental investigation is given in Section 3. Additionally, the dataset characteristics and descriptive statistics are presented. Section 4 presents the discussion and limitations of the research. Section 5 concludes the paper.

2   Related work

One of the most significant breakthroughs in face recognition was achieved with the DeepFace model [18]. This model is often cited as one of the first deep learning methods to achieve face recognition performance comparable to that of humans. It is considered a foundational work that laid the foundations for the architecture and methodology of modern face recognition systems. Since the principles of this model are also applied in current face recognition systems [7], it can also be used to study various parts of the system and methodologies, such as face template representations. DeepFace proposed a system that integrates several key components: face alignment, a seven-layer neural network, and a two-dimensional vector space for image comparison. The system's uniqueness lies in the fact that, before transmitting the image to the neural network, the face is aligned with a 3-dimensional model, thereby ensuring that the network views the face in the most similar canonical position. This significantly reduced variations between photos due to head rotation, lighting, and other factors, thereby improving classification accuracy and, consequently, face recognition. Another significant work in the field of face recognition, marking the transition from classification systems to direct extraction of facial feature vectors, is FaceNet [14]. The essence of this method is that, instead of classifying into specific identity classes, the network is trained to immediately project faces into Euclidean space so that their distances reflect their similarity. This vector representation enables easy face identification and authentication via vector distance calculations. For these reasons, this algorithm is also applied in modern face recognition systems [16]. Unlike DeepFace, which relied on a classification approach, FaceNet uses a triplet loss function that forces the network to learn projections in which images of the same identity are close together. Pictures of different identities are far apart. Training is based on three examples: an anchor, a positive, and a negative. The network is trained so that the distance between the anchor and the positive example is smaller than that between the anchor and the negative example. This allows the system to be more flexible and easier to adapt to real-world scenarios where the number of users is not fixed. This model originally used the GoogLeNet network [17], but later methods used simplified or more efficient architectures. One of the most important aspects is that the model's output layer is directly used as a face template (a 128- or 512-dimensional vector) for comparison with other face vectors. In this way, the network becomes not only a classifier but also a map of faces, with faces existing in a unified space.

One of the most advanced methods for solving the face recognition problem has significantly improved the quality of face vectors: ArcFace [1]. Unlike previous methods that primarily focused on generating vectors in Euclidean space (e.g., FaceNet), ArcFace introduced a new principle: additive angular margin loss. This means that the model was trained to classify faces so that class solutions are separated not only by distance but also by the angular distance between their vectors, thereby improving the sharpness of class boundaries. Traditional Softmax-based loss functions have certain shortcomings - they do not always ensure a sufficiently high separation between classes in vector representations, especially when faces are visually similar. ArcFace addresses this issue by transforming the traditional Softmax formula into an angular space, thereby introducing an additional margin between different classes. In this way, the network is forced to push the other people's vectors further apart in angular space. This principle is beneficial for separating face templates – vector representations of different person identities become better separated. ArcFace is one of the most modern and successful methods for generating face templates [23]. Due to the angular-loss method used, the vectors it generates have extremely high resolution but may also contain latent information about facial attributes. Another critical development in facial template generation over the past decade has been the emergence of alternative loss functions. This has significantly improved neural networks' ability to distinguish between latent representations of different individuals. One of the most significant works in this area is the CosFace method [20]. This method introduces the large-margin cosine loss to improve the clarity

of classification boundaries in face recognition tasks. This method is considered a natural predecessor of ArcFace and is widely used in modern face recognition systems. The idea behind this method is that traditional Softmax functions do not provide sufficient inter-class separation in the embedding space. Instead, the authors proposed a modified loss function that introduces an additional angular deviation (cosine margin) between the correct class representation and those of other classes. In this method, each vector is normalized to unit length, so its similarity is evaluated solely by the angular measure, rather than by the Euclidean distance. This ensures that identical individuals in the embedding space are significantly closer to each other than different ones, even with minimal differences in data or various disturbances such as lighting, pose, image resolution, and so on. It is essential to note that CosFace not only enhanced identification accuracy but also significantly influenced the development of subsequent facial template generation methods. The angular deviation, as an additional component of the Softmax function, has become standard and has been further developed in other works [21]. This method also contributes to the understanding of how facial templates or facial feature vectors are formed, which will be examined in this work. One of the early but significant steps in the development of facial recognition representations was the SphereFace method proposed in 2017 [8]. This method fundamentally changed the paradigm for training facial feature templates by introducing geometric structure into the vector space in which these representations are formed. Sphere-Face introduces an angular Softmax function, which is a modification of the traditional Softmax function. This function helps increase the interclass separation of vectors during training, not only in amplitude but also in angle. This allows projections in the face representation space to be organized on the surface of a hypersphere, which, as the authors demonstrate, facilitates better face discrimination in that space. This method is characterized by restricting representations to the surface of a single sphere, thereby eliminating the dependence on vector length. Instead, identity is encoded only through the angular distance between vectors, making the system more resistant to lighting, pose, and other extraneous image factors. The authors empirically demonstrated that this method significantly enhances the model's ability to generalize and recognize faces unseen during training, which is of great importance for real-world biometric identification systems. The SphereFace model has been influential not only for its direct application but also for its role in developing later models, such as CosFace and ArcFace, that rely on angular-based loss functions and introduce different normalization and other strategies. SphereFace can be considered the first method to provide a mathematically rigorous justification for the utility of hyperspherical embedding spaces in face recognition. This method is relevant because it demonstrates how facial representations can be modeled to be distinguishable in the embedding space, even with limited data. It is also still used in modern systems [22]. Since the representations of this model are based on angular topology, they can be assumed to retain essential facial features, such as gender, ethnicity, and even facial structure. Like the other models reviewed, the purpose of SphereFace was not to form feature vectors of faces to extract facial features from them. However, because the embedding vectors preserve information, it can be assumed that these methods partially encode not only identity but also other semantically meaningful information.

3 Experimental investigation and results

3.1 Descriptive statistics of the analyzed dataset

Neurotechnology provided the dataset used in the experimental investigation. The dataset comprises face photos with accompanying metadata and annotated facial attributes, including age, gender, and race. To achieve the best possible results, the dataset was processed, and suitable data for training models was selected. This dataset stores facial feature vectors extracted using the MegaMatcher SDK [10]. Each of these vectors is 512-dimensional. The dataset was balanced to create the most comprehensive set possible, featuring a diverse range of attributes. The data was evenly distributed across the categories and classes predicted from the facial feature vectors. The resulting dataset consisted of 115,023 records, all of which were correct, with no invalid or empty data. Most of the records are anonymized, containing unique information about individual people; that is, most of the data consists of unique identities.

In Figure 1a, the distribution of gender categories is presented. The category labels consist of two

discrete values: "Male" and "Female", with 57,512 images assigned to "Male" and 57,511 to "Female". The labels for people's race are assigned by individuals who provide the pictures and declare their race. Declared race was treated as an original category of the class. In this research, the five races of people were analyzed from images. The distribution by race is presented in Figure 1b. Approximately 23,000 images have been selected from each race category. As shown in Figure 1, the dataset is balanced.
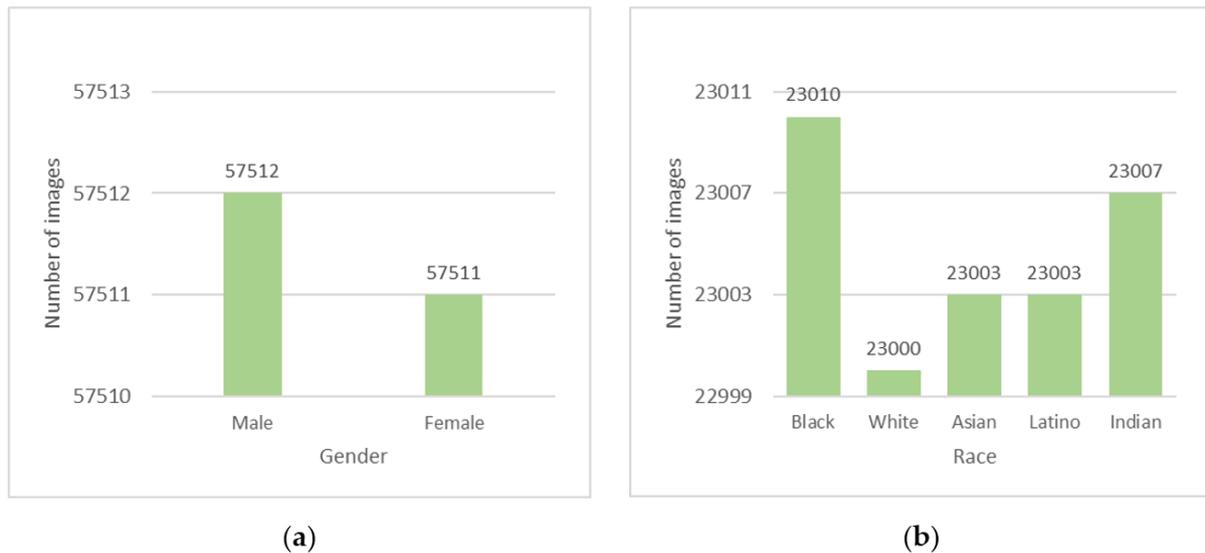

(a) **(b)**

Figure 1: Descriptive statistics of the dataset analyzed: (a) People's images distribution by gender; (b) People's images distribution by race.

Compared with the dataset's gender and race distributions, the age distribution cannot be assumed to be balanced. The primary reason was that during dataset creation, most images provided by people depicted individuals aged 20 to 40. The class distribution by age group is illustrated in Figure 2. The class "20-29" receives the most images (57,893), and only 645 are assigned to the class "60-69".
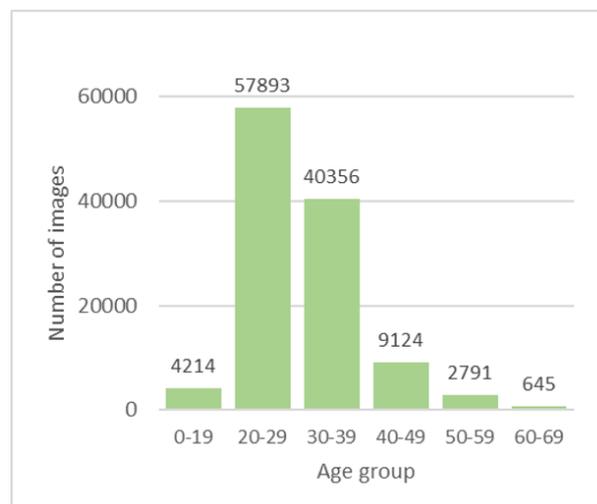

Figure 2: Descriptive statistics of the dataset analyzed: (a) People's images distribution by gender; (b) People's images distribution by race.

3.2 Experimental research environment and technologies

An environment with an Intel Core i7-14700 K processor and 64 GB of DDR5 RAM was used for the experiments. The models were developed and tested in the Windows 11 environment using Python 3.9.18. Several Python frameworks and libraries were used to implement the models and process the

data. The deep learning framework PyTorch was used to create, train, and test neural networks. The Pandas and NumPy libraries were utilized to process and prepare the data, while the scikit-learn library was employed to develop supervised deep learning algorithms.

3.3 The results of the experimental investigation

This section presents the results of an experimental study assessing whether it is possible to accurately identify human facial features, including gender, race, and age, using face templates generated by a neural network within a face recognition algorithm. The study employed both deep learning and traditional machine learning algorithms, including decision trees, SVM, and random forests. The main reason for choosing these two machine learning algorithms is their speed. The dataset is quite huge for traditional machine learning algorithms. Additionally, during the research, hyperparameter optimization was performed, which requires considerable time and resources. For deep learning algorithms, the dataset has been split into training (80%) and testing (20%) subsets, using stratified sampling to maintain class balance. To ensure reliable evaluation results for traditional machine learning algorithms, a 5-fold cross-validation method was used to train the models. To achieve the best possible results, hyperparameter optimization was performed, and various models and network architectures were tested. Grid search was used to train machine learning-based models. For neural networks, the learning rate, number of epochs, and batch size were varied, and various model architectures were also tested.

3.3.1 Results using the deep learning models

One of the primary methods used in this work to classify facial features from facial feature vectors was deep neural networks. In the experiments, a multi-layer perceptron (MLP) was employed, comprising an input layer (with dimensions equal to the feature vector), several hidden layers, and an output layer corresponding to the number of predicted classes. The Adam optimizer was selected to optimize the hyperparameters. During the experiments, learning rates (LR) of 0.01, 0.001, 0.0001, and 0.00001 were used. The batch size (BS) was also explored, varying during the experiments as follows: 32, 64, 256, and 512. The models were trained for up to 100 epochs. Still, an early stopping mechanism was employed that terminated training if no significant improvement in the validation loss was observed within 10 epochs. The patience parameter, set to 10, was used for early stopping, and the minimum loss value was set to 0.0001. These parameters indicate that if the learning loss does not decrease by 0.0001 within 10 training epochs, training is stopped and the model with the best result is saved. This method avoids overtraining in models and accelerates the experimental process, as there is no need to wait for training to complete 100 epochs. Separate neural networks with different architectures and applied methods were created for each facial feature to be determined, to find the best models. Different regularization techniques were employed, including dropout layers and batch normalization layers. A total of 144 neural network model trainings were performed using hyperparameter optimization.

During the research, three neural network architectures (TemplateGenderNet1, TemplateGenderNet2, and TemplateGenderNet3) were developed to classify people by gender. The TemplateGenderNet1 neural networks architecture is a fully connected (FC) neural network that consists of following layers: an input layer, which corresponds to the number of ele-ments in the face template vector and is equal to 512; the first dense layer reduces the di-mension to 64 neurons and applies the ReLU activation function; the second dense layer reduces the dimension to 16 neurons and applies the activation function; the output layer has one neuron and a sigmoid activation function is applied. The model was trained using the binary cross-entropy (BCE) loss function, since the classification task is binary – male or female. In Figure 3a, the architecture of the TemplateGenderNet1 neural network is presented.

The medium-sized neural network architecture is named TemplateGenderNet2. This network is larger than the first network tested and incorporates a new type of method, batch normalization layers. This is one of the significant improvements to the architecture. This normalization method was tested to stabilize and improve neural network training. It reduces internal covariate shift and helps
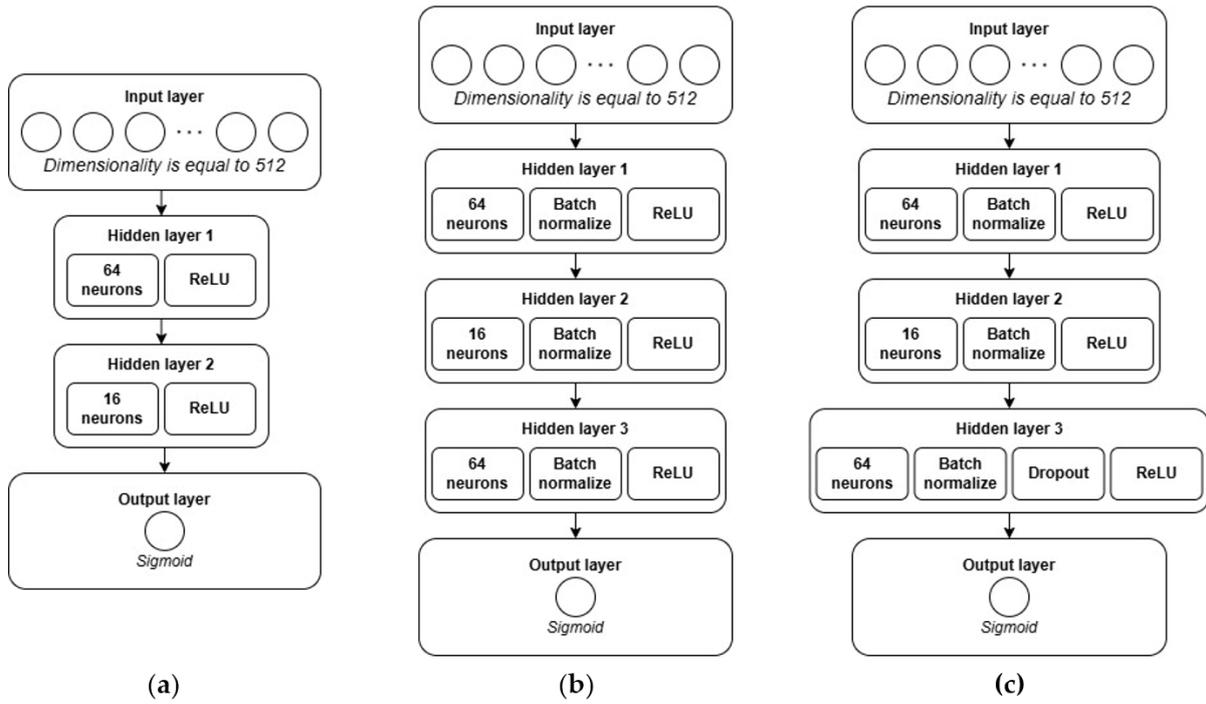
Figure 3: Architecture of neural network models for gender determination: (a) TemplateGenderNet1; (b) TemplateGenderNet2; (c) TemplateGenderNet3.

maintain more appropriate gradient descent. This technique enables larger learning steps, reduces the risk of gradient explosion or vanishing, and often enhances the model's overall generality. The architecture of this model is shown in Figure 3b. The third model has been named TemplateGenderNet3. This architecture has more layers and additional regularization mechanisms to enhance the model's generalization. This model maintains the same general structure as previous versions. Still, the main difference lies in the inclusion of a dropout layer, applied after the third hidden layer and before the activation function. This technique is fundamental in neural networks, as it helps reduce the risk of overfitting. This involves randomly shutting down a specific subset of neurons during training, helping prevent the model from becoming overly dependent on particular combinations of neurons. During training, each neuron has a probability of being turned off, i.e., its value is set to zero. This forces the network to learn resilient, less correlated connections. In this model architecture, the dropout probability was equal to 0.5. This means that each neuron in that layer has a 50% chance of being turned off at each batch. This decision was based on practice, where a value of 0.5 often serves as a practical compromise between regularization and learning efficiency in medium-sized networks. The architecture of this network is shown in Figure 3c.

A comparative analysis of the models and hyperparameter search revealed that a more complex model does not necessarily yield better results, as shown in Table 1. The TemplateGenderNet1 usually achieves 91.1+-1.2 percent accuracy, which is the lowest and has the largest variation. Meanwhile, TemplateGenderNet2 achieves 91.8+-0.9 percent, and TemplateGenderNet3 achieves 91.7+-1.1 percent accuracy. This indicates the last two models have basically the same accuracy metrics, and comparing them with the first model, the difference between them is not statistically significant (using the t distribution, the p-values are 0.07 for the 1st and 2nd models, and 0.15 for the 1st and 3rd model, 0.78 for the 2nd and 3rd model comparison).

To assess the quality of the classification models in more detail, a confusion matrix (Figure 4) was compiled for the model that achieved the best result (93.04% accuracy, with an F1 score of 0.93) – TemplateSexNet3 (epochs is equal to 11; learning rate is equal to 0,0001; batch size is equal to 512). These data allow us to state that the model is not biased towards any class. These results indicate that the training data was sufficiently balanced and that the model architecture is suitable for this binary classification task.

Table 1: The summary of the results of all trained models in determining gender.

Model Name	Epochs Number	Learning Rate	Batch Size	Accuracy(%)	F1 Score
TemplateGenderNet1	4	0.01	32	90.71	0.90
TemplateGenderNet1	10	0.01	64	91.09	0.91
TemplateGenderNet1	6	0.01	256	90.53	0.90
TemplateGenderNet1	5	0.01	512	91.39	0.92
TemplateGenderNet1	2	0.001	32	91.65	0.92
TemplateGenderNet1	3	0.001	64	91.75	0.92
TemplateGenderNet1	2	0.001	256	91.75	0.92
TemplateGenderNet1	3	0.001	512	91.70	0.92
TemplateGenderNet1	5	0.0001	32	91.45	0.92
TemplateGenderNet1	7	0.0001	64	91.45	0.92
TemplateGenderNet1	11	0.0001	256	91.06	0.91
TemplateGenderNet1	13	0.0001	512	92.85	0.93
TemplateGenderNet1	24	0.00001	32	91.21	0.91
TemplateGenderNet1	36	0.00001	64	91.17	0.91
TemplateGenderNet1	24	0.00001	256	90.31	0.90
TemplateGenderNet1	18	0.00001	512	87.13	0.87
TemplateGenderNet2	8	0.01	32	92.49	0.93
TemplateGenderNet2	6	0.01	64	92.33	0.92
TemplateGenderNet2	2	0.01	256	92.06	0.92
TemplateGenderNet2	4	0.01	512	92.12	0.92
TemplateGenderNet2	7	0.001	32	92.18	0.92
TemplateGenderNet2	4	0.001	64	92.11	0.92
TemplateGenderNet2	2	0.001	256	92.12	0.92
TemplateGenderNet2	3	0.001	512	91.87	0.92
TemplateGenderNet2	8	0.0001	32	92.20	0.92
TemplateGenderNet2	11	0.0001	64	91.98	0.92
TemplateGenderNet2	8	0.0001	256	91.29	0.91
TemplateGenderNet2	9	0.0001	512	92.66	0.93
TemplateGenderNet2	35	0.00001	32	91.77	0.92
TemplateGenderNet2	32	0.00001	64	91.54	0.92
TemplateGenderNet2	32	0.00001	256	90.73	0.91
TemplateGenderNet2	26	0.00001	512	88.91	0.89
TemplateGenderNet3	9	0.01	32	92.35	0.92
TemplateGenderNet3	5	0.01	64	92.24	0.92
TemplateGenderNet3	6	0.01	256	92.12	0.92
TemplateGenderNet3	4	0.01	512	92.27	0.92
TemplateGenderNet3	4	0.001	32	92.15	0.92
TemplateGenderNet3	6	0.001	64	91.95	0.92
TemplateGenderNet3	3	0.001	256	91.79	0.92
TemplateGenderNet3	3	0.001	512	91.71	0.92
TemplateGenderNet3	7	0.0001	32	91.72	0.92
TemplateGenderNet3	5	0.0001	64	91.66	0.92
TemplateGenderNet3	10	0.0001	256	91.39	0.91
TemplateGenderNet3	11	0.0001	512	93.04	0.93
TemplateGenderNet3	43	0.00001	32	91.89	0.92
TemplateGenderNet3	43	0.00001	64	91.75	0.92
TemplateGenderNet3	31	0.00001	256	89.85	0.90
TemplateGenderNet3	28	0.00001	512	88.20	0.88

Predicted label

	Male	Female
Male	10500	1140
Female	915	10725

Figure 4: The confusion matrix of the model achieving the highest accuracy in the gender classification task.

In addition to the gender classification task, three variations of a deep neural network were created for the people's race classification task (TemplateRaceNet1, TemplateRaceNet2, TemplateRaceNet3). The structure of these models was similar to the models already used for gender classification. These neural networks were adapted for multiclass classification. One of the first models studied was TemplateRaceNet1. The main difference lies in the output layer, which uses the Softmax activation function to compute class probabilities. The TemplateRaceNet1 network consists of five hidden layers, where the output of each layer is normalized using batch normalization. Additionally, based on previous data, the gender prediction model performed best when normalization and dropout layers were incorporated. The architecture of this model is presented in Figure 5a.
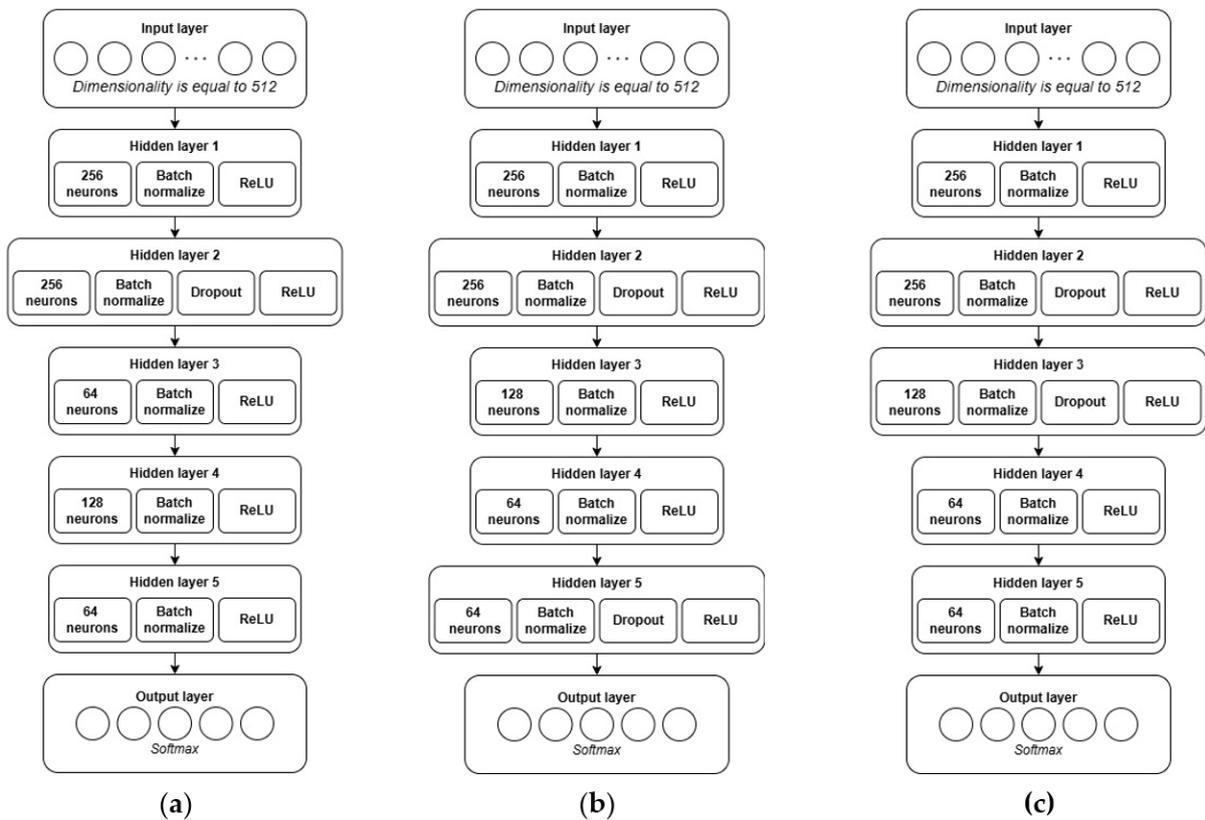
Figure 5: Architecture of neural network models for people's race determination: (a) TemplateRaceNet1; (b) TemplateRaceNet2; (c) TemplateRaceNet3.

The second neural network created for people race classification was TemplateRaceNet2. The architecture of this model is very similar to that of the first network, but it has several fundamental differences. One of the main changes is the increased depth of the network, along with the additional

inclusion of a dropout layer in one of the last hidden layers (Figure 5b). The third network architecture (TemplateRaceNet3) tested the dropout technique in the middle of the network. Its architecture is analogous to that of the second network, but the location of the dropout layer has been changed, and one dropout layer has been removed (Figure 5c). The classification and hyperparameter combination results are summarized in Table 2.

The model TemplateRaceNet1 performed best in terms of overall classification accuracy (70.3%, with 2.1% standard deviation) and F1 score (0.70 and 0.02, respectively). Its architecture was characterized by a balanced level of complexity, which matched the structure of the available data. Meanwhile, the second model (TemplateRaceNet2) was deeper, with an additional fully connected and dropout layer; however, this did not improve accuracy (mean value of 69.9% and standard deviation is 2.2%). The third model, TemplateRaceNet3, performed the worst on both validation accuracy (69.9% with a)standard deviation of 2.2%) and F1 score (0.70 and 0.02, respectively). This suggests that although the network architecture was sufficiently deep, it is not optimal for predicting people's race, and there is no significant difference between the three models (for t-distribution p-values, all cases are>0.5). To determine whether the best race-determination model, TemplateRaceNet1, is not biased towards any class, the confusion matrices for each class were reviewed. Since this model performs multiclass classification, the confusion matrices are presented for each class. This allows for a visual review of the matrix to decide whether there is a strong bias towards any class.

The confusion matrix representing the prediction of the white race (Figure 6a) shows that the model performed exceptionally well in distinguishing this class. After calculating the precision and sensitivity (recall) for this class, the results are 79.57% and 76.98%, respectively. Figure 6b illustrates the prediction results for determining whether a person belongs to the black race. The model's accuracy for this class was 75.23%, and its sensitivity was 77.03%. These results are close to the first-class prediction results and the model's overall weighted F1 score, indicating that the model can recognize this class with a relatively high level of confidence. The confusion matrix for class "Asian" is shown in Figure 6c. The model's accuracy for this class was 74.89%, and its sensitivity was 72.34%. These results do not deviate too much from the overall accuracy of the model and the previously analyzed classes. No anomalies in the model's accuracy are visible in this confusion matrix. The accuracy and sensitivity metrics of the best TemplateRaceNet1 model for the "Latino" class were calculated, yielding the following results: accuracy, 60.45%; and sensitivity, 60.66% (Figure 6d). This is the worst result of all the previously reviewed classes. This means that this model has a harder time predicting the class of faces belonging to the same race as the person. It is possible that the face template does not encode enough information to distinguish faces of this class reliably. The results of the model validation prediction (class of "Indian") in confusion matrix format are shown in Figure 6e. Based on the confusion matrix shown, the following model results were obtained for this class: accuracy, 66.28%; sensitivity, 69.74%. These results indicate that the model distinguished instances of this class more accurately than the class "Latino", but its accuracy and sensitivity lagged behind those of the classes "White", "Black", and "Asian". Given these results, it can be inferred that the face template contains insufficient information to distinguish this class from others.

Similar to gender- and race-detection models, three deep neural network architectures have been developed to analyze age (TemplateAgeNet1, TemplateAgeNet2, TemplateAgeNet3). All network architectures are presented in Figure 7. The first neural network created is TemplateAgeNet1, which consisted of six fully connected layers. In this network, each layer performs batch normalization, and a dropout layer is inserted midway through the network. This architecture is shown in Figure 7a. The second architecture tested for age attribute detection was TemplateAgeNet2. This architecture has a gradually decreasing number of neurons. It uses the normalization and dropout layers from previous networks, but the output layer differs. It consists of 6 output neurons, whose activations are computed using the Softmax function (Figure 7b). For the third age prediction architecture (TemplateAgeNet3), the network that performed best on the people race detection task was selected. To adapt it for age group detection, the output layer was modified so that the results returned by the model corresponded to the number of age categories (Figure 7c).

After training neural networks for age attribute detection and searching for the best hyperparameters, the worst results were obtained from all previously studied models. None of the models achieved

Table 2: The summary of the results of all trained models in determining people's race.

| Model Name	Epochs Number	Learning Rate	Batch Size	Accuracy(%)	F1 Score
TemplateRaceNet1	34	0.01	32	71.15	0.71
TemplateRaceNet1	31	0.01	64	71.69	0.72
TemplateRaceNet1	17	0.01	256	70.36	0.70
TemplateRaceNet1	13	0.01	512	70.46	0.70
TemplateRaceNet1	31	0.001	32	71.81	0.72
TemplateRaceNet1	27	0.001	64	71.47	0.71
TemplateRaceNet1	12	0.001	256	70.70	0.71
TemplateRaceNet1	9	0.001	512	70.92	0.71
TemplateRaceNet1	29	0.0001	32	72.47	0.73
TemplateRaceNet1	21	0.0001	64	71.76	0.72
TemplateRaceNet1	13	0.0001	256	69.81	0.70
TemplateRaceNet1	20	0.0001	512	70.84	0.71
TemplateRaceNet1	58	0.00001	32	71.26	0.71
TemplateRaceNet1	49	0.00001	64	70.04	0.70
TemplateRaceNet1	32	0.00001	256	66.06	0.66
TemplateRaceNet1	31	0.00001	512	64.40	0.64
TemplateRaceNet2	24	0.01	32	68.08	0.68
TemplateRaceNet2	25	0.01	64	69.61	0.70
TemplateRaceNet2	20	0.01	256	70.27	0.70
TemplateRaceNet2	12	0.01	512	70.11	0.70
TemplateRaceNet2	25	0.001	32	71.27	0.71
TemplateRaceNet2	24	0.001	64	71.11	0.71
TemplateRaceNet2	17	0.001	256	70.95	0.71
TemplateRaceNet2	12	0.001	512	70.90	0.71
TemplateRaceNet2	34	0.0001	32	72.38	0.72
TemplateRaceNet2	21	0.0001	64	71.81	0.72
TemplateRaceNet2	18	0.0001	256	71.22	0.71
TemplateRaceNet2	14	0.0001	512	69.81	0.70
TemplateRaceNet2	59	0.00001	32	71.42	0.71
TemplateRaceNet2	47	0.00001	64	69.64	0.70
TemplateRaceNet2	34	0.00001	256	66.50	0.67
TemplateRaceNet2	30	0.00001	512	63.87	0.64
TemplateRaceNet3	44	0.01	32	71.33	0.71
TemplateRaceNet3	33	0.01	64	71.48	0.71
TemplateRaceNet3	18	0.01	256	70.62	0.71
TemplateRaceNet3	11	0.01	512	69.71	0.70
TemplateRaceNet3	37	0.001	32	71.68	0.72
TemplateRaceNet3	21	0.001	64	71.07	0.71
TemplateRaceNet3	13	0.001	256	70.57	0.71
TemplateRaceNet3	21	0.001	512	71.07	0.71
TemplateRaceNet3	27	0.0001	32	71.95	0.72
TemplateRaceNet3	25	0.0001	64	71.25	0.71
TemplateRaceNet3	14	0.0001	256	69.55	0.70
TemplateRaceNet3	14	0.0001	512	69.33	0.69
TemplateRaceNet3	55	0.00001	32	69.92	0.70
TemplateRaceNet3	43	0.00001	64	68.86	0.69
TemplateRaceNet3	29	0.00001	256	66.18	0.66
TemplateRaceNet3	28	0.00001	512	64.50	0.64
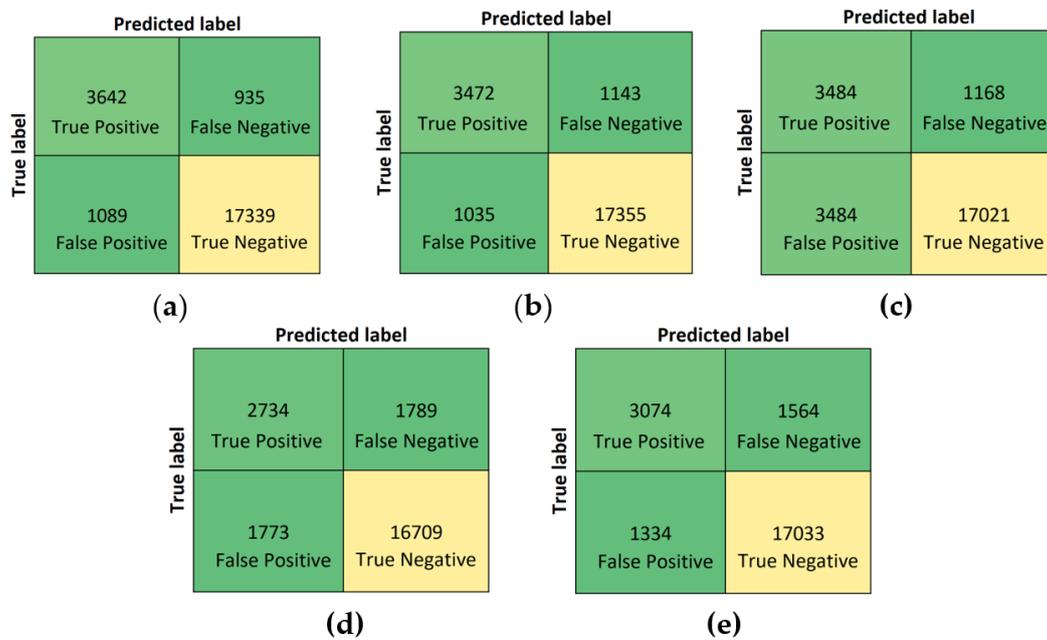
Figure 6: The confusion matrix of each class: (a) "White"; (b) "Black"; (c) "Asian"; (d) "Latino"; (e) "Indian".

an accuracy of 70%, and all had F1 scores below the model validation accuracy estimates (Table 3). Among the reviewed models, the TemplateAgeNet1 achieved the best performance, with 65.52% accuracy (average 63.8% and standard deviation 0.9%) and a maximum F1 score of 0.63 (average F1 score 0.60 and standard deviation 0.03). The results of other age-determination models were not far from those of the best model, and their accuracy and F1 scores were similar (for the t distribution, the smallest p value was 0.07, between the 1st and 2nd models, which does not indicate a significant difference).

To assess the effectiveness of the age determination models, a confusion matrix analysis was conducted for each age group (Figure 8). Additionally, the age attribute was less well-balanced than the people's race or gender attributes. During the validation of the TemplateAgeNet1 model, which achieved the highest accuracy, it was observed that the model did not make any predictions for the class "0-19". This means that the model was unable to identify a single case belonging to this class. Based on the confusion matrix for this class (Figure 8a), the model made no positive predictions, indicating that it is entirely unsuitable for identifying this age group. The class "20-29" had the highest number of dataset items for model training and validation. This may lead to the model blindly and positively predicting this class, since the probability of guessing correctly by random methods is relatively high. After calculating the accuracy and sensitivity for this class from the confusion matrix results (Figure 8b), the following were obtained: accuracy, 82.77%; sensitivity, 70.32%. These are relatively high indicators, which typically suggest that the model can recognize face templates belonging to this class. However, such results may arise because records from this age group dominate the dataset. This is likely why the model learned to predict this class, as it was statistically the best choice. The third class predicted by the model was "30-39". This group had slightly fewer data than class "20-29", but significantly more than the remaining classes. The confusion matrix of the model predictions for this class is shown in Figure 8c. The model accuracy for this class was 60.29%, and the sensitivity was 58.05%. These indicators show that, as with the second class, the model tended to make predictions for this class. The accuracy and sensitivity were slightly lower than those of the second class, but not significantly so. Comparing these results, one can observe a specific dependence between the class's data size and the model's predictions, suggesting that the model tends to choose the option with the highest probability. The fourth class in the dataset was the third largest. This class, along with the second and third, dominated the dataset. The impact of this fact on the model's predictions for this class is evident in Figure 8d. The model accuracy for this class of data was 35.31%, and the sensitivity was 45.29%.

Table 3: The summary of the results of all trained models in determining people's age.

| Model Name	Epochs Number	Learning Rate	Batch Size	Accuracy(%)	F1 Score
TemplateAgeNet1	17	0.01	32	63.57	0.58
TemplateAgeNet1	10	0.01	64	63.28	0.58
TemplateAgeNet1	9	0.01	256	63.43	0.58
TemplateAgeNet1	9	0.01	512	63.45	0.58
TemplateAgeNet1	16	0.001	32	64.46	0.58
TemplateAgeNet1	12	0.001	64	63.95	0.59
TemplateAgeNet1	13	0.001	256	65.52	0.63
TemplateAgeNet1	7	0.001	512	64.96	0.62
TemplateAgeNet1	16	0.0001	32	64.96	0.67
TemplateAgeNet1	10	0.0001	64	63.36	0.58
TemplateAgeNet1	11	0.0001	256	64.29	0.61
TemplateAgeNet1	13	0.0001	512	64.67	0.62
TemplateAgeNet1	23	0.00001	32	63.44	0.58
TemplateAgeNet1	20	0.00001	64	62.97	0.58
TemplateAgeNet1	24	0.00001	256	62.38	0.58
TemplateAgeNet1	30	0.00001	512	62.50	0.58
TemplateAgeNet2	11	0.01	32	63.54	0.58
TemplateAgeNet2	13	0.01	64	63.74	0.59
TemplateAgeNet2	9	0.01	256	63.20	0.58
TemplateAgeNet2	6	0.01	512	63.27	0.58
TemplateAgeNet2	12	0.001	32	63.48	0.58
TemplateAgeNet2	10	0.001	64	63.19	0.58
TemplateAgeNet2	15	0.001	256	64.90	0.62
TemplateAgeNet2	12	0.001	512	63.48	0.58
TemplateAgeNet2	17	0.0001	32	60.62	0.58
TemplateAgeNet2	20	0.0001	64	64.06	0.62
TemplateAgeNet2	7	0.0001	256	63.01	0.60
TemplateAgeNet2	10	0.0001	512	63.88	0.61
TemplateAgeNet2	21	0.00001	32	62.97	0.58
TemplateAgeNet2	20	0.00001	64	62.56	0.58
TemplateAgeNet2	26	0.00001	256	62.74	0.58
TemplateAgeNet2	29	0.00001	512	62.20	0.60
TemplateAgeNet3	16	0.01	32	63.83	0.59
TemplateAgeNet3	13	0.01	64	63.62	0.58
TemplateAgeNet3	9	0.01	256	63.24	0.58
TemplateAgeNet3	8	0.01	512	63.33	0.58
TemplateAgeNet3	18	0.001	32	64.44	0.58
TemplateAgeNet3	10	0.001	64	63.24	0.58
TemplateAgeNet3	6	0.001	256	65.05	0.63
TemplateAgeNet3	9	0.001	512	64.86	0.63
TemplateAgeNet3	22	0.0001	32	63.19	0.58
TemplateAgeNet3	18	0.0001	64	64.74	0.62
TemplateAgeNet3	11	0.0001	256	64.62	0.62
TemplateAgeNet3	11	0.0001	512	64.16	0.62
TemplateAgeNet3	21	0.00001	32	63.22	0.58
TemplateAgeNet3	21	0.00001	64	62.86	0.58
TemplateAgeNet3	26	0.00001	256	63.23	0.60
TemplateAgeNet3	26	0.00001	512	60.86	0.57
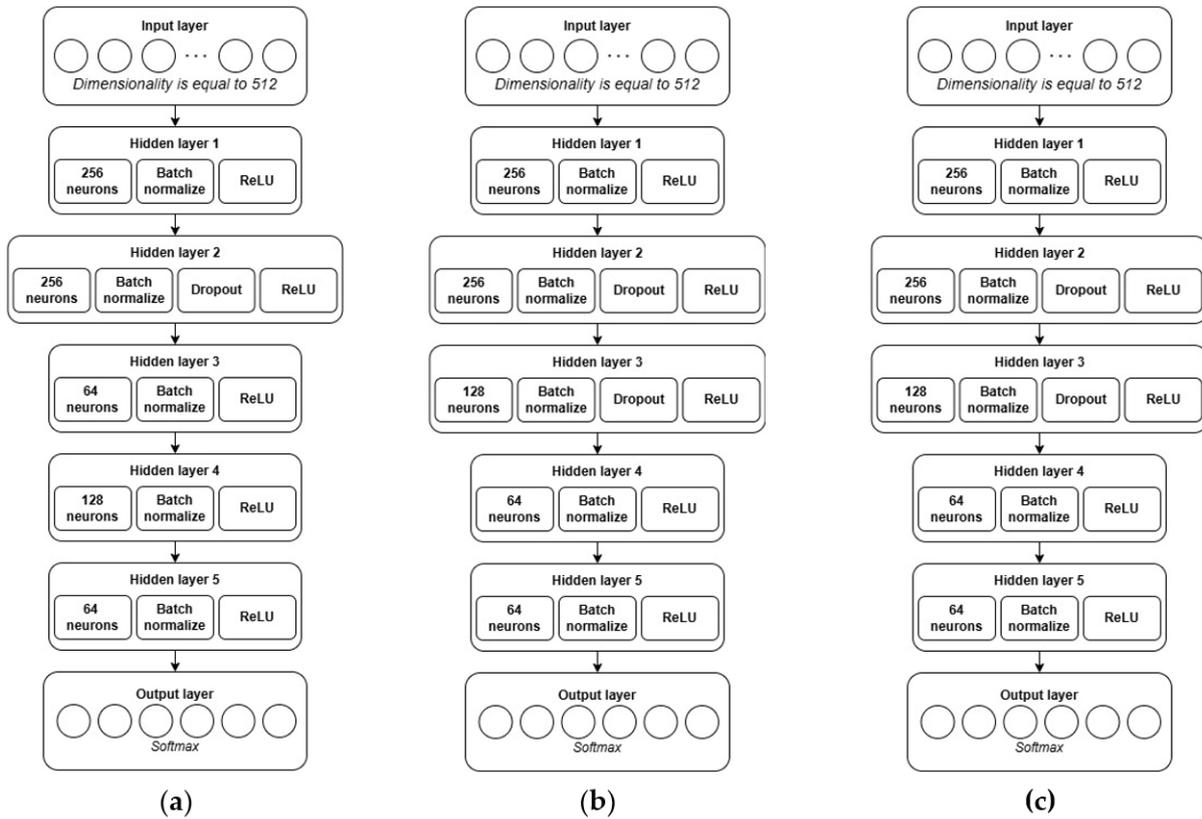
Figure 7: Architecture of neural network models for people's age determination: (a) TemplateAgeNet1; (b) TemplateAgeNet2; (c) TemplateAgeNet3.

As with the previously analyzed classes, a relationship can be observed between the size of the dataset items and the model's accuracy in determining a particular class. The confusion matrix of the model's predictions for the fifth class, "50-59", is shown in Figure 8e. Based on the confusion matrix, the model's accuracy for this class was 30.29%, and its sensitivity was 49.72%. This is a poor result, indicating that the model is unable to recognize face templates belonging to this class reliably. The last class ("60-69") had the least training data, which may have affected the results shown in the confusion matrix (Figure 8f). The confusion matrix for this class is very similar to that of the first class. The accuracy for this class was 0, indicating that the model made no positive predictions. Based on these results, it can be concluded that this model is not suitable for identifying this age group.

Table 4: The values of hyperparameters during the training of machine learning algorithms.

Algorithm	Parameter	Value of Parameter
Decision Tree	Maximum tree depth	3, 5, 10, not limited
	Minimum number of samples	2, 5, 10, 20, 50, 100
	Maximum number of features	5, 20, 50, 100, 200, 500, not limited
Random Forest	Number of trees	50, 100, 200, 500
	Maximum tree depth	10, 20, 30, 50, not limited
	Minimum number of samples	2, 5, 10, 20, 50
	Minimum number of nodes	1, 2, 4, 8, 16
	Bootstrap	True, False
SVM	Regularization strength	0.1, 1.0, 10.0
	Class weight	None, Balanced
	Iteration	2000, 3000, 4000, 5000

3.3.2 Results using the machine learning models

In this research, supervised learning methods were used as alternatives to neural networks to evaluate how classical machine learning algorithms perform on face templates. Such a study allows not only the comparison of different methods' effectiveness but also the assessment of whether complex
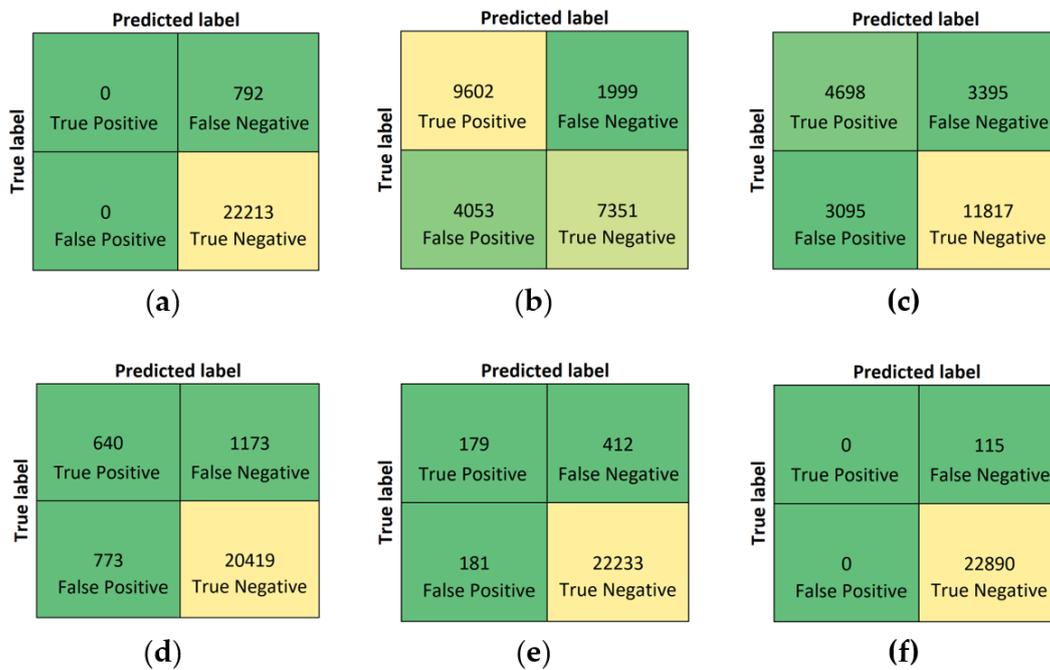
Figure 8: The confusion matrix of each class for the age attribute: (a) "0-19"; (b) "20-29"; (c) "30-39"; (d) "50-59"; (e) "60-69".

deep neural networks are necessary for solving these tasks. In this research, decision trees, SVM, and random forests have been employed to detect age, race, and gender from face vector templates. To find the most effective model, hyperparameter optimization was performed using a grid search. A total of 1000 hyperparameter combinations were tested during the study. The grid search values for both models are presented in Table 4.

The best random forest model achieved 89.28% accuracy in gender classification, with an overall F1 score of 0.89. Examining the results for each class, we observed a fairly symmetrical classification quality – the accuracy, sensitivity, and F1 score of both classes remained nearly the same. This indicates that the model handles data imbalance well and is not biased towards either class. These results suggest that the random forest classifier can be an effective alternative method for feature classification using face templates. Close to the best results were achieved using the SVM, where the accuracy is equal to 86.57%, and the F1 score is equal to 0.86. The best decision-tree model achieved 65.59% accuracy, indicating that this algorithm is not suitable for gender classification, as it lags significantly behind neural networks and other advanced machine learning algorithms.

For people's race detection, the random forest achieved an overall classification accuracy of 69.23%. This suggests that this method is not significantly behind neural networks trained for predicting people's race. A decision tree model for race classification is attractive because its training time is considerably shorter; however, it is often prone to overfitting when the tree becomes too deep. Despite the hyperparameter search, the model's accuracy was only 30.65%. The SVM model is again close to the random forest results by achieving an accuracy of 61.52% and an F1 score of 0.61. These results indicate that the model was unable to effectively learn to generalize facial feature vectors and associate them with the corresponding racial classes of the people. This indicates that decision trees are unsuitable.

A random forest classifier was used to predict the age attribute in the task. After the hyperparameter search, the best results were obtained with the parameters listed in Table 5. More details are presented in Table 6, where the mean and standard deviation for each fold are shown. With these parameters, the model achieved its best accuracy of 58.03% and an F1 score of 0.50. The results for the best model indicate that this classifier is unsuitable for the age attribute, as the F1 score is close to 0.5, suggesting that the model cannot accurately predict age classes. In case of age detection, the overall best results has been obtained using the SVM algorithm, where accuracy is equal to 62.63%

Table 5: The summary of the results of all trained models in determining people's gender, race, and age using machine learning algorithms.

Algorithm	Task	Best Parameter	Accuracy (%)	F1 Score
Decision Tree	Gender detection	Maximum tree depth: 5 Minimum number of samples: 10 Maximum number of features: 500	65.59	0.71
	Race detection	Maximum tree depth: 5 Minimum number of samples: 2 Maximum number of features: 5	30.65	0.30
	Age detection	Maximum tree depth: 5 Minimum number of samples: 2 Maximum number of features: not limited	51.27	0.43
Random Forest	Gender detection	Number of trees: 500 Maximum tree depth: not limited Minimum number of samples: 5 Minimum number of nodes: 1 Bootstrap: False	89.28	0.89
	Race detection	Number of trees: 1000 Maximum tree depth: 30 Minimum number of samples: 2 Minimum number of nodes: 1 Bootstrap: False	69.23	0.69
	Age detection	Number of trees: 200 Maximum tree depth: 30 Minimum number of samples: 2 Minimum number of nodes: 1 Bootstrap: False	58.03	0.50
SVM	Gender detection	Regularization strength: 10.0 Class weight: Balanced Iteration: 5000	86.57	0.86
	Race detection	Regularization strength: 0.1 Class weight: Balanced Iteration: 5000	61.52	0.61
	Age detection	Regularization strength: 0.1 Class weight: None Iteration: 5000	62.63	0.59

and F1 score - 0.59. However, it remains from the results obtained by deep learning algorithms. After training and optimizing hyperparameters using a decision tree algorithm, the classifier achieved an overall accuracy of 51.27% and an F1 score of 0.43. Like the random forest algorithm, this model yielded poor results and was unable to determine the age attribute from facial feature vectors. As mentioned earlier, these results may be due to the principle of facial template formation, since the goal of face detection algorithms is to store only identity-related data in the facial feature vector.

4 Discussion

After a more detailed analysis of the results from the most accurate age-group determination model using deep neural networks, it was observed that neural networks are not a suitable solution for age classification from face templates. Based on the confusion matrices, it can be seen that the model's predictions and accuracy for each class correlate with the number of exemplars in the dataset for specific age groups. This means that the model learns to make predictions based solely on statistical properties, rather than exact data values. One possible reason age recognition models performed poorly is that the principles and goals underlying the acquisition of a face template differ from those underlying the acquisition of a face template. The goal of face recognition algorithms that generate such templates is to encode only those features relevant to determining a person's identity. Since a person's age is not a defining feature of their identity, it is likely that the algorithms used to form face templates disregard this information and do not include it in the face attribute vector.

The research performed has shown that deep neural networks are characterized by greater flexibility and the ability to learn more complex data structures. Still, they require more testing when designing network architectures. Meanwhile, traditional machine learning algorithms offer higher speed and greater ease of application, but may not be sufficiently accurate when classifying more complex attributes. In the gender attribute classification task, differences in model performance emerged during

Table 6: Performance (mean ± standard deviation) over 5-fold cross-validation.

Model	Task	Metric	Fold1	Fold2	Fold3	Fold4	Fold5	Mean	Std
Decision Tree	Age detection	Accuracy	0.5092	0.5153	0.5152	0.5149	0.5091	0.5127	0.0032
		F1 score	0.4341	0.4334	0.4202	0.4180	0.4366	0.4285	0.0087
	Race detection	Accuracy	0.3047	0.3110	0.3062	0.3013	0.3095	0.3065	0.0039
		F1 score	0.2959	0.2972	0.2946	0.2944	0.2960	0.2956	0.0011
	Gender detection	Accuracy	0.6571	0.6533	0.6562	0.6610	0.6519	0.6559	0.0035
		F1 score	0.7093	0.7073	0.7037	0.7130	0.6995	0.7066	0.0052
Random Forest	Age detection	Accuracy	0.5822	0.5813	0.5840	0.5802	0.5740	0.5803	0.0038
		F1 score	0.5030	0.5000	0.5027	0.4983	0.4929	0.4994	0.0041
	Race detection	Accuracy	0.6864	0.6937	0.6941	0.6954	0.6920	0.6923	0.0035
		F1 score	0.6858	0.6931	0.6933	0.6947	0.6911	0.6916	0.0035
	Gender detection	Accuracy	0.8944	0.8927	0.8938	0.8919	0.8914	0.8928	0.0013
		F1 score	0.8930	0.8911	0.8926	0.8903	0.8895	0.8913	0.0015
SVM	Age detection	Accuracy	0.6227	0.6292	0.6327	0.6256	0.6212	0.6263	0.0042
		F1 score	0.5908	0.5969	0.6015	0.592	0.5878	0.5937	0.0048
	Race detection	Accuracy	0.6135	0.6179	0.6137	0.6165	0.6145	0.6152	0.0017
		F1 score	0.6125	0.6167	0.612	0.6157	0.6129	0.6139	0.0019
	Gender detection	Accuracy	0.8642	0.8665	0.867	0.8645	0.8665	0.8657	0.0012
		F1 score	0.8625	0.8644	0.8654	0.8628	0.8644	0.8639	0.0011

the experiments. Deep neural networks yielded the best results, particularly those with architectures that incorporated batch normalization and dropout layers. The best neural network achieved 93.04% accuracy, while the results of supervised learning algorithms were somewhat worse, with the best random forest model achieving 89.28% accuracy (Figure 9a). In the people's race classification task, the models' overall accuracy was lower than in the gender classification task. The best result was achieved by the deep neural network TemplateRaceNet1, whose architecture incorporated normalization and dropout layers, yielding an accuracy of 72.47%. Other deep learning algorithms achieved results that were very close to those of the best models, but their accuracy was slightly lower. The results of the machine learning algorithms were less favorable – the random forest achieved 69.23% accuracy, whereas the decision tree performed poorly, with its accuracy not exceeding 40% (Figure 9b). The results of the people's race classification models indicate that this task is significantly more complex than gender classification, requiring larger neural networks and more extensive parameter optimization. Several models were tested in the age classification task, yielding unsatisfactory results. The best result was achieved by the TemplateAgeNet1 model, which achieved 65.52% accuracy and an F1 score of 0.63. In the experiments on age attribute prediction, the same trend was observed as with the previously studied facial features. The results of machine learning algorithms were inferior to those of deep neural networks. The best result was achieved with the random forest algorithm, with the best model achieving an accuracy of 58.03% (Figure 9c). In summary, none of the tested models yielded satisfactory results in determining age from face templates. This can be explained by the fact that face template generation algorithms are specifically designed to reduce the amount of information in the facial feature vector that is unrelated to the person's identity.

The generalizability of the presented results is influenced by the choice of face template extraction methodology. All experiments in this study rely on face templates generated using the proprietary Neurotechnology face recognition SDK. As different face recognition SDKs employ distinct feature representations, embedding spaces, and preprocessing strategies, the downstream performance of attribute detection models may vary accordingly. Therefore, while the observed performance trends provide insight into the proposed approach, the reported quantitative results should be considered specific to the employed SDK and may differ when alternative template extraction frameworks are used. Additionally, the deep learning models considered in this study were intentionally kept relatively shallow in order to focus on methodological feasibility. A systematic evaluation of deeper and more complex architectures lies outside the scope of this work and will be addressed in future research.
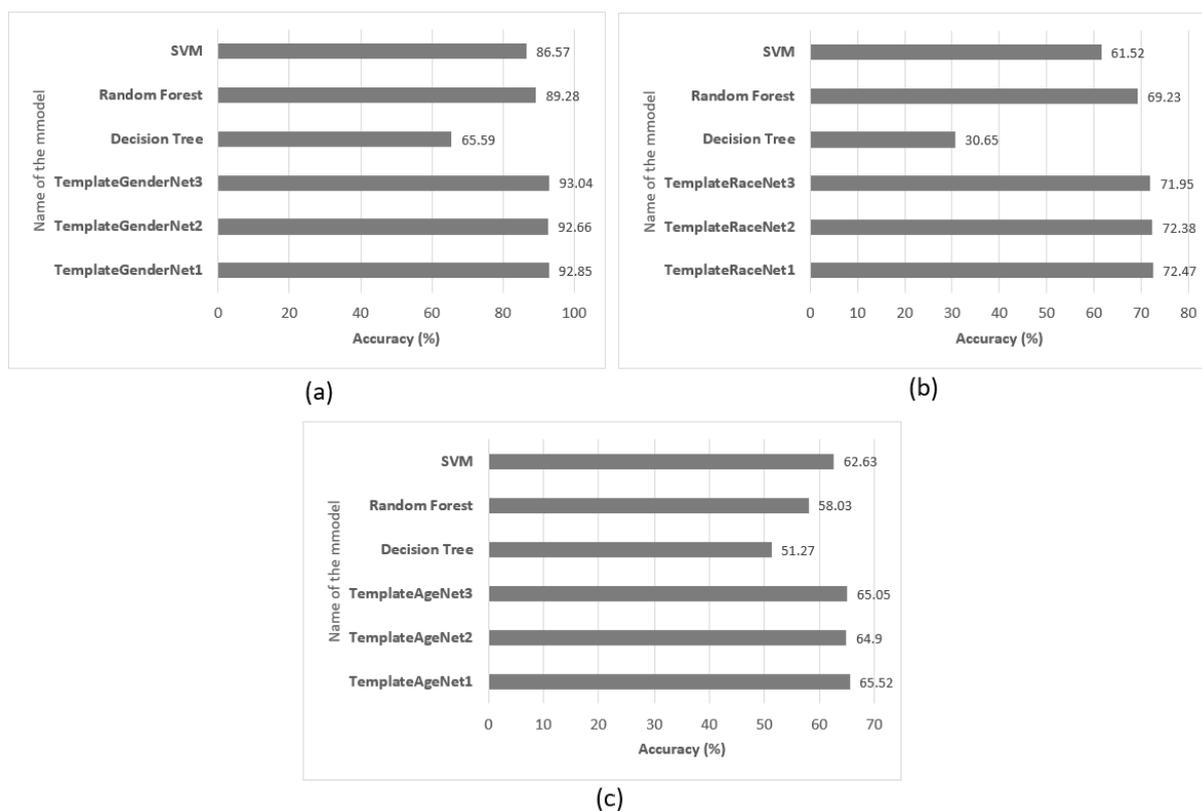
Figure 9: Comparison of results between deep learning and machine learning algorithms: (a) gender detection; (b) people's race detection; (c) age detection.Comparison of results between deep learning and machine learning algorithms: (a) gender detection; (b) people's race detection; (c) age detection.

5   Conclusion

This work provides the first systematic empirical evaluation of demographic attribute inference from a previously unexamined facial template dataset, thereby establishing analytical benchmarks for what can and cannot be inferred from template-based biometric representations. The results demonstrate that facial templates preserve varying degrees of demographic information, with gender being the most reliably inferred attribute and age the least, highlighting clear differences in the separability of demographic features within template space.

For the experimental analysis, a real dataset of face templates was prepared, associated with certain facial features – gender, age, and race. When preparing this data, data cleaning, balancing, and merging were performed to ensure that the distribution of facial feature classes was as similar as possible and that the data were suitable for training and testing. The prepared data allowed for experiments with both traditional machine learning algorithms and deep neural networks.

Experimentally, nine deep learning neural network architectures were created, and two traditional supervised learning methods were evaluated: decision tree and random forest classifiers. A hyperparameter search was performed to determine the optimal models. The easiest attribute to recognize was gender – the deep neural network achieved the best result, determining gender from the face template with an accuracy of 93.04%. Experiments on people's race classification demonstrated that this facial feature is more challenging to decide on; the deep neural network achieved the best result, with an accuracy of 72.47%. The age classification results were the worst, indicating that the amount of facial age data remaining in the face template was minimal. The accuracy of the best age prediction model was 65.52%. However, upon closer examination of the results, it was observed that the model adapted to the dataset's statistical distribution to make predictions. These results show that facial templates retain some information about the properties of the original image; however, this information must be significant in face recognition, as the goal of facial recognition algorithms is to accommodate only data related to identity in the facial template.

Author contributions

The authors contributed equally to this work.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition, *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, IEEE, 2019*, pp. 4690-4699, 2019.

[2] Fu, X.; Ma, Q.; Yang, F.; Zhang, C.; Zhao, X.; Chang, F.; Han, L. (2024). Crop pest image recognition based on the improved ViT method, *Information Processing in Agriculture*, 11(2), pp. 249-259, 2024.

[3] Kotwal, K.; Marcel, S. (2025). Review of demographic fairness in face recognition, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2025.

[4] Kozłowski, M.; Racewicz, S.; Wierzbicki, S. (2024). Image Analysis in Autonomous Vehicles: A Review of the Latest AI Solutions and Their Comparison, *Applied Sciences*, 14(18), p. 8150, 2024.

[5] Kumar, Y.; Shrivastav, S.; Garg, K.; Modi, N.; Wiltos, K.; Woźniak, M.; Ijaz, M. F. (2024). Automating cancer diagnosis using advanced deep learning techniques for multi-cancer image classification, *Scientific Reports*, 14(1), p. 25006, 2024.

[6] LeCun, Y.; Bengio, Y.; Hinton, G. (2015). Deep learning, *nature*, 521(7553), pp. 436-444, 2015.

[7] Li, M. (2025). Research and Analysis of Facial Recognition Based on FaceNet, DeepFace, and OpenFace, *In ITM Web of Conferences. EDP Sciences*, 70, p. 03009, 2025.

[8] Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. (2017). Sphereface: Deep hypersphere embedding for face recognition, *In Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE*, pp. 212-220, 2017.

[9] Meftah, I.; Hu, J.; Asham, M. A.; Meftah, A.; Zhen, L.; Wu, R. (2024). Visual detection of road cracks for autonomous vehicles based on deep learning, *Sensors*, 24(5), p. 1647, 2024.

[10] Neurotechnology (2025). MegaMatcher SDK (Version 2025.1) [Software], https://neurotechnology.com/megamatcher.html

[11] Praneeth, D.; Sree, T. R.; Viswanatha, V.; Sai, M. A. (2024). AI Enabled Home Security System Using Object Detection and Face Recognition with Haar Cascade Algorithm, *In 2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET), IEEE*, pp. 1-6, 2024.

[12] Ranjan, R.; Sankaranarayanan, S.; Castillo, C. D.; Chellappa, R. (2017). AI Enabled Home Security System Using Object Detection and Face Recognition with Haar Cascade Algorithm, *An all-in-one convolutional neural network for face analysis. In 2017 12th IEEE International Conference on automatic face & gesture recognition (FG 2017), IEEE*, pp. 17-24, 2017.

[13] Rashid, A. B.; Kausik, A. K. (2024). AI revolutionizing industries worldwide: A comprehensive overview of its diverse applications, *Hybrid Advances*, p. 100277, 2024.

[14] Schroff, F.; Kalenichenko, D.; Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering, *In Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE*, pp. 815-823, 2015.

[15] Stefanovič, P.; Ramanauskaitė, S. (2023). Travel direction recommendation˙ model based on photos of user social network profile, *IEEE Access*, 11, pp. 28252-28262, 2023.

[16] SV, S. (2021). Security Monitoring System Using FaceNet For Wireless Sensor Network, *arXiv preprint*, arXiv:2112.01305., 2021.

[17] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Rabinovich, A. (2015). Going deeper with convolutions, *In Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE*, pp. 1-9, 2015.

[18] Taigman, Y.; Yang, M.; Ranzato, M. A.; Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification, *In Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE*, pp. 1701-1708, 2014.

[19] Virbukaitė, S.; Bernatavičienė, J.; Imbrasienė, D. (2024). Glaucoma Identification Using Convolutional Neural Networks Ensemble for Optic Disc and Cup Segmentation, *IEEE Access*, 12, pp. 82720-82729. 2024.

[20] Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Liu, W. (2018). Cosface: Large margin cosine loss for deep face recognition, *In Proceedings of the IEEE conference on computer vision and pattern recognition, IEEE*, pp. 5265-5274, 2018.

[21] Wang, M.; Deng, W.; Hu, J.; Su, S. (2025). Marginal debiased network for fair visual recognition, *Pattern Recognition*, 158, p. 111046, 2025.

[22] Wu, S.; Lin, X.; Yao, T. (2022). Research on Face Recognition Technology of Subway Automatic Ticketing System based on Neural Network and Deep Learning, *International Journal of Advanced Computer Science and Applications*, 13(12), 2022.

[23] Xin, Y.; Zhou, Y.; Jiang, J. (2024). Self-Attentioned Arcface Towards Improved Efficiency for Deep Face Recognition, *SSRN*, Available at SSRN 4898101, 2024.

[24] Xu, Q. A.; Jayne, C.; Chang, V. (2024). An emoji feature-incorporated multi-view deep learning for explainable sentiment classification of social media reviews, *Technological Forecasting and Social Change*, 202, p. 123326, 2024.

[25] Yangxiaoxiao, Z. (2021). mage modal analysis in art design and image recognition using AI techniques, *Journal of Intelligent & Fuzzy Systems*, 40(4), pp. 6961-6971, 2021.

C | O | P | E

Member since 2012
JM08090

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Malickas, M.; Stefanovič, P.; Ramanauskaitė, S. (2026). Identification of Facial Features in Face Templates Using Deep Neural Network Models, *International Journal of Computers Communications & Control*, 21(2), 7294, 2026.
https://doi.org/10.15837/ijccc.2026.2.7294