Fırat Üniversitesi Müh. Bil. Dergisi
35(2), 561-567, 2023

Araştırma Makalesi
https://doi.org/10.35234/fumbd.1233505

Feature Selection From MagFace Face Recognition Model With Optimization Algorithms

Mehmet Fatih ÖZDEMIR[1,2*], Davut HANBAY[2]
[1] Havelsan A.Ş., Ankara, Türkiye
[2] Bilgisayar Mühendisliği, Mühendislik Fakültesi, Malatya, Türkiye
[*1] mfatih.ozdemir@inonu.edu.tr, [2] davut.hanbay@inonu.edu.tr

(Geliş/Received: 12/01/2023; Kabul/Accepted: 20/06/2023)

Abstract: In recent years, many studies have been carried out in the field of artificial intelligence in the literature with the development of equipment. Face recognition algorithms have an important place among these developments. Among the face recognition algorithms, the most successful ones are usually deep learning approaches. Models such as SphereFace, CosFace, ArcFace, and MagFace are important deep learning models in the literature. Despite their success, deep learning models are often computationally costly. Therefore, advanced methods are needed to reduce the computational load for these models. One of the most valid methods for this is to choose the most valuable one among embedding features for face recognition. Thus, cost can be reduced, and accuracy values can be increased even more. In this study, the most valuable of the 512 embedded features in the MagFace model was tried to be obtained by using PSO, GA, SCA, and DE optimization algorithms. As a result, accuracy values of 99.83%, 98.57%, and 98.65% were reached for 193, 252, and 280 features selected in the LFW, CFP, and AGEDB datasets, respectively.

Key words: Face Recognition, Feature Selection, Optimization.

Optimizasyon Algoritmaları ile MagFace Yüz Tanıma Modelinden Özellik Seçimi

Öz: Son yıllarda gelişen donanımlarla birlikte literatürde yapay zekâ alanında birçok çalışma yapılmaktadır. Bu gelişmeler arasında yüz tanıma algoritmaları önemli bir yere sahiptir. Yüz tanıma algoritmaları arasında ise en başarılı olanları genellikle derin öğrenme yaklaşımlarıdır. SphereFace, CosFace, ArcFace, MagFace gibi modeller literatürde yer alan önemli derin öğrenme modelleridir. Derin öğrenme modelleri başarılarının aksine genellikle hesaplama açısından maliyetlidir. Bu nedenle, bu modeller için hesaplama yükünü azaltacak gelişmiş yöntemlere ihtiyaç duyulmaktadır. Bunun için en geçerli yöntemlerden biri gömülü yüz öznitelikleri arasından en değerli olanı seçmektir. Böylece maliyet düşürülebilir hatta başarı değerleri daha da arttırılabilir. Bu çalışmada PSO, GA, SCA, DE optimizasyon algoritmaları kullanılarak MagFace 512 gömülü özelliklerinin en değerlileri elde edilmeye çalışılmıştır. Sonuç olarak LFW, CFP, AGEDB veri setlerinde seçilen değerli 193, 252, 280 öznitelikleri sırasıyla 99.83, 98.57, 98.65 doğruluk değerlerine ulaşılmıştır.

Anahtar kelimeler: Yüz Tanıma, Öznitelik Seçimi, Optimizasyon.

1. Introduction

Face recognition algorithms have been developing rapidly in recent years. Along with these developments, there are many problems that need to be dealt with, such as low resolution, blurriness, and lighting issues. In addition, it is necessary to correctly extract the real features of the face for successful face recognition.

Face recognition algorithms are examined in two categories in the literature: closed-set and open-set. In the open-set approach, the identities in the train dataset and test dataset are expected to be completely different. After the face features are extracted in the face recognition model, a comparison is made using the nearest neighbor metric. In the closed-set approach, the identities of the training dataset must be included in the test dataset. Therefore, it operates in a more limited space. On the other hand, the open-set approach is universally adaptable. However, obtaining learning characteristics in the open-set system is often difficult due to high inter-class similarity and large intra-class variation. SphereFace [1], CosFace [2], ArcFace [3], and MagFace [4] can be shown among the successful open-set models in the literature. SphereFace used A-Softmax loss function, which is characterized as angular Softmax, enabling convolutional neural networks to learn angular distinctive features. The A-Softmax loss function can be easily generalized for multiple classes, similar to the softmax loss in the classification process. To learn the task of the minimum distance between classes being greater than the maximum

* Sorumlu yazar: mfatih.ozdemir@inonu.edu.tr. Yazarların ORCID Numarası: [1] 0000-0003-3563-054X, [2] 0000-0003-2271-7865

distance within classes, the limit values were obtained. The model trained on the CASIA-WebFace [5] dataset achieved competitive results on various criteria in the LFW [6], YTF [7], and MegaFace [8] datasets.

Deep learning models in face recognition methods generally use large margin Softmax or angular Softmax loss functions instead of traditional Softmax. All these loss functions aim to maximize the variance between classes and minimize the variance within the class. In the CosFace model, a new loss function called Large Margin Cosine Loss (LMCL) is proposed with this approach. It uses a cosine margin to further maximize the decision margin in angular space. The model is trained on the CASIA-WebFace dataset using the large margin cosine loss function. Benchmarks were made on the CosFace model, MegaFace, YTF, and LFW face recognition datasets.

ArcFace, on the other hand, has proposed a new loss function with the same approach. ArcFace, the proposed face recognition model to obtain highly distinctive features, has a clear geometric interpretation as it fits the geodetic distance on a hypersphere precisely with its new loss function. The softmax loss function used in many standard face recognition approaches has been reinterpreted.

MagFace, on the other hand, emerged as a model that falls into the category of loss functions that learn a universal feature that can evaluate the quality of the face. The model attracted relevant candidates to class centers while removing irrelevant candidates, thus providing an adaptive mechanism to learn well-structured in-class feature distributions. As a result, it was emphasized that it prevents the models from overfitting to noisy, low-quality samples. In the study, extensive experiments on face recognition, quality evaluation, and clustering underscored its superiority over state-of-the-art technology.

In the study, it was aimed to select valuable features by using face-embedding features, which are outputs of MagFace face recognition algorithm. Valuable features were selected using 4 optimization algorithms for embedded features in the output. The success of the selected features in the datasets has been tested.

The organization of the article consists of four sections. Methods and algorithms used are described in the materials and methods section. Details of the application are given in the third section and the conclusions are given in the fourth section.

2. Material and Methods

2.1. MagFace

Face recognition systems perform poorly when there is a lot of variation in the faces they analyze. To tackle this issue, a new approach called MagFace [4] is proposed. MagFace uses a set of losses to learn a feature embedding that can measure the quality of a face based on its magnitude. This approach ensures that the magnitude of the feature embedding increases for faces that are more likely to be recognized. It also includes a mechanism to improve the structure of within-class feature distributions, preventing overfitting on low-quality samples and enhancing face recognition in real-world scenarios.

The another face recognition losses based on cosine similarity lacks a detailed constraint beyond a fixed margin, resulting in an unstable within-class structure, especially when faces have high variability. MagFace addresses this problem by encoding a quality measure into the face representation. Unlike previous methods that introduce uncertainty terms, MagFace optimizes the magnitude of each feature without normalization, allowing the use of the cosine-based metric commonly used in inference systems. By simultaneously considering the direction and magnitude, the learned face representation becomes more robust to variations in real-world faces. This is the first work to unify feature magnitude as a quality indicator in face recognition.

MagFace incorporates two auxiliary functions: the magnitude-aware angular margin and the regularizer. The angular margin concentrates high-quality face samples around the cluster center, penalizing samples with large magnitudes more. The regularizer rewards samples with large magnitudes and pushes them towards the boundary of the feasible region. This approach extends the ArcFace method by incorporating the magnitude-aware margin and regularizer, promoting diversity among different face samples and similarity among samples of the same class. MagFace's design is intuitive and also provides theoretical guarantees, ensuring a unique optimal solution for the magnitude and revealing the difficulties of recognition based on feature magnitudes. The new loss proposed by MagFace is shown in Eq. 1.

$$L_M ag = \frac{1}{N}\sum_{i=1}^{N} L_i \quad where \quad L_i = \frac{e^s cos\left(\theta_{y_i} + m(a_i)\right)}{e^s cos\left(\theta_{y_i} + m(a_i)\right) + \sum_{j \neq y_i}^{e^S} cos\theta_j} \tag{1}$$

where λg determines the balance between the classification loss and the regularization loss. a_i is magnitude. $m(a_i)$ is the magnitude of the angular edge, that is the ascending convex function. $g(a_i)$ is the regularizer that diminishing convex function. Let's consider a training batch consisting of N face samples $\{f_i, y_i\}_{i=1}^{N}$, where $(f_i \epsilon R)$ d represents the d −dimensional embedding obtained from the last fully connected layer of the neural networks. Each sample is associated with a class label y_i, ranging from 1 to n, representing different identities. The angle θ_j is between j-th class center and f_i. The parameter m, with a value greater than zero $(m > 0)$, represents the additive angular margin, while s refers to the scaling parameter.

2.2. Particle swarm optimizer (PSO)

Particle Swarm Optimizer (PSO) [9] is metaheuristic as it forms few or no suppositions for problems being optimized and is able to explore very large areas of candidate solutions. It also does not use the gradient of the optimized problem. That is, it does not require the optimization problem to be differentiable as required by classical optimization methods such as quasi-newton and gradient descent methods. However, metaheuristic methods such as PSO do not guarantee that an optimal solution will be found.

The PSO algorithm works for optimization using a population (called a swarm) of candidate solutions (called particles). These particles are displaced in the search space according to some formulas. Particles are guided by the swarm's best-known positions along with the best-known positions in the search space. Discovered improved locations guide the movements of the swarm. The optimization process cannot be guaranteed, but a satisfactory solution can be hoped. Cost-effect is quite successful as to other heuristic algorithms. The position and velocity of each particle are regenerated using Eq. 2.

$$v_{t+1}^{i} = v_t^i + c_1 r_1 (pbest_t^i - x_t^i) + c_2 r_2 (gbest - x_t^i) \quad (2)$$

where v_t^i indicates the ith particle's velocity at the iteration t, indicates x_t^i, ith particle at the iteration t, c_1 refers to cognition learning parameter, c_2 refers to social learning parameter, r_1 and r_2 are random numbers (0-1), $pbest_t^i$ indicates local best position for ith particle at the iteration t, $gbest$ indicates global best position for all particles. v_{t+1}^i indicates the ith particle's the velocity at next iteration $t + 1$.

Particles are re-positioned as to an objective function. Each particle compares the fitness values from the previous best position to the current position in order to find the new best position. $c_1 r_1 (pbest_t^i - x_t^i)$ in the formula is used for this. Also, $c_2 r_2 (gbest - x_t^i)$ in the formula is used to approximate particles to the global best position of all particles.

2.3. Genetic algorithm (GA)

For the first time in history, genetic algorithm emerged as a stochastic solution to optimization problems. Genetic algorithm (GA) [10] is an approach to solving constrained or unconstrained optimization problems. At each step, individuals are randomly selected from the current population. Selected individuals use as parents for the next generation. The population changes towards an optimal solution over successive generations. This algorithm is used to solve problems where the fitness function is stochastic, non-differentiable, discontinuous or not highly linear. At the same time, it can be used to solve problems that are not very suitable for standard optimization algorithms. The genetic algorithm summary is as follows:

- Initialize populations randomly and determine fitness of the population.
- While algorithm stop conditions are not satisfied:
    a. Calculate the score of the current population using raw fitness data for each member
    b. Calculate expectation values to convert raw fitness scores into a more useful range of values.
    c. Selects parents based on their expectation values.
    d. Some of the individuals with a lower fitness value are selected as elite and transferred to the next population.
    e. The offspring are obtained either by random changes (mutation) in a single parent or by combining vector inputs from a pair of parents (crossover).
    f. Replaces the current population with the offspring to create the next generation.

2.4. Sine cosine algorithm (SCA)

A new population-based stochastic optimization algorithm called Sine Cosine Algorithm (SCA) [11] was proposed in 2016. SCA initially generates multiple candidate solutions randomly. A mathematical model based on sine and cosine functions is used. This mathematical model makes it fluctuate out of the solution space or towards the best solution. Various adaptive and random variables are also integrated to explore the search space of the optimization.

The common approach among algorithms in the field of stochastic population-based optimization is to consider the optimization process in two stages. The first phase aims to discover promising regions of the search space with a sudden and high randomness in the set of solutions. The second is usage stage that random variations are significantly less at this stage than at the discovery stage.

2.5. Differential evolution (DE)

Differential Evolution (DE) [12] is a stochastic, population-based optimization method. The DE method is simple, easy to use, robust and fast. DE tries to iteratively improve a candidate solution. Such methods are often known as meta-heuristics, as they make few suppositions about the problem and can discover very large areas of candidate solutions. DE is used for multidimensional real-valued functions, ignoring the gradient of the optimized problem. Therefore, DE does not require the optimization problem to be differentiable. As a result, DE is noisy, time-varying, etc. can be used in optimization problems. DE algorithm summary is as follows:

- Initialization individuals with NP (NP Population Size).
- Calculate the fitness values for all individuals.
- While the termination condition is not satisfied:
    a. For each individual (j) in the population.
        i. Generate three vectors $(r_1, r_2, r_3 \in (1, NP))$ from the current population randomly.
        ii. Generate random number $(i_{rand} \in (1, n))$.
        iii. Create mutant vectors using a mutation using $(v_{i,j} = r_1 + F(r_2 - r_3))$ formula.
        iv. Evaluate three vectors using their fitness values using following equation.
            1. $u_{i,j} = \begin{cases} v_{i,j}, & if\ r_{i,j} <= Crossover\ Probability\ or\ i = i_{rand} \\ x_{i,j}, & otherwise \end{cases}$
        v. Select winning vectors in the new generation.

3. Application

All codes runned for this study using a system with Intel(R) Xeon(R) W-2245 processor and 32 GB RAM. The codes were writen in python 3.8 using niapy [13] library.

Features for the datasets are extracted using the model trained using the iResNet100 backbone shared in the MagFace [4] repository. Then, the most valuable ones were tried to be selected by using these features. The cosine distance metric was used as the objective function of the algorithms. Each algorithm was run with 30 population size and 100 iterations. The k-fold value of 10 is selected. Thus, it is aimed to obtain the results more objectively and accurately.

A study was conducted to select the most valuable face recognition embedding features, as shown in Figure 1. First, embedded features were extracted from the images in the dataset using the open-set face recognition model MagFace [4]. The MagFace model generates 512 embedded features. Then, it is aimed to select the most valuable features from the 512 embedded features. In the literature, optimization algorithms are among the most preferred algorithms for feature selection. In this study, PSO [9], GA [10], SCA [11], DE [12] optimization algorithms were used to select the most valuable features. These algorithms were run for 100 iterations respectively and the results were obtained. In the study, LFW [6], CFP [14], AGEDB [15] datasets from the MagFace article were used as datasets. In addition, to generalize the optimization, the existing datasets (LFW + CFP + AGEDB) were combined and the fourth dataset was included in the study.

In Table 1, the number of the most valuable features selected from each data set using optimization algorithms is shared. As a result of the experiments, between 193 and 285 valuable features were obtained. It has been determined that the number of valuable features obtained is almost half of the total.

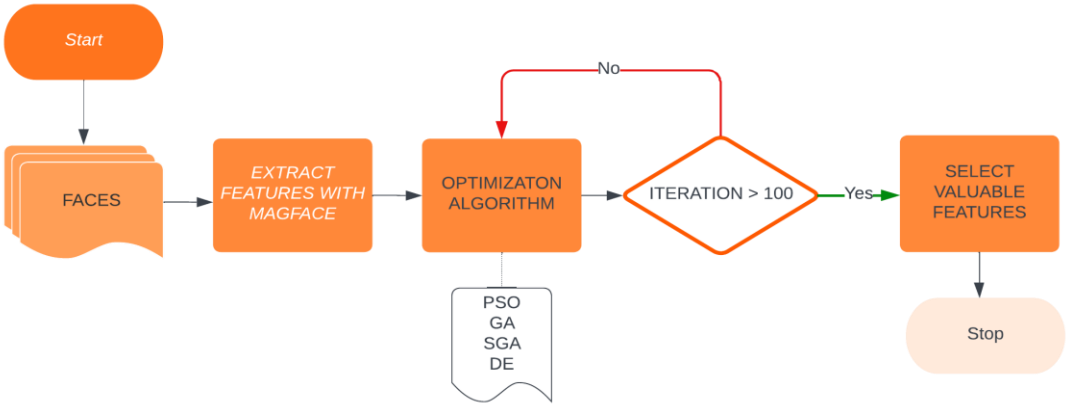

Figure 1. Flowchart of selection valuable features.

Table 1. Number of feature selections.

Algorithm	LFW	CFP	AGEDB	LFW + CFP + AGEDB
PSO	193	252	280	260
GA	231	269	245	265
SCA	234	269	254	256
DE	205	285	275	257

In Table-2, a study was conducted to examine the effects of the selected features in any data set on other data sets by using their indices. In other words, the results in other datasets were examined by using the features obtained in any dataset. It is used to evaluate the accuracy metric. The following steps are performed to obtain the accuracy value.

- The indexes corresponding to the features selected from the dataset are obtained.
- The values in the indexes obtained for other datasets are selected.
- The accuracy value is calculated for the selected features.

Table 2. Selected features accuracies.

Algorithm	Selected Features	LFW	CFP	AGEDB
SphereFace [1]	-	99.67	96.84	97.05
CosFace [2]	-	99.78	98.26	98.17
ArcFace [3]	-	99.81	98.40	98.05
MagFace [4]	-	**99.83**	98.46	98.17
PSO	LFW(193)	**99.83**	97.55	97.81
	CFP(252)	99.76	**98.57**	97.86
	AGEDB(280)	99.8	97.67	**98.65**
	LFW+CFP+AGEDB(260)	99.75	98.02	98.33
GA	LFW(231)	**99.83**	97.4	97.63
	CFP(269)	99.78	98.18	97.88
	AGEDB(245)	99.73	97.54	98.3
	LFW+CFP+AGEDB(265)	99.8	98.08	97.93
SCA	LFW(234)	99.81	97.77	97.81
	CFP(269)	99.8	98.17	98.05
	AGEDB(254)	99.76	97.84	98.18
	LFW+CFP+AGEDB(256)	99.78	97.77	98.06
DE	LFW(205)	**99.83**	97.57	97.81
	CFP(285)	99.7	98.48	98
	AGEDB(275)	99.78	98.1	98.55
	LFW+CFP+AGEDB(257)	99.76	98.05	98.23

MagFace [4] has the best accuracy in LFW dataset, but with the help of valuable features obtained using PSO algorithm, the same accuracy has been achieved by using less features. On the other hand, in CFP and AgeDB datasets, more accuracy results were obtained with the help of valuable features obtained by using the PSO algorithm. As a result, accuracy values of 99.83, 98.57, 98.65 were obtain by using fewer features in LFW, CFP, AGEDB datasets, respectively.

4. Conclusion

In current study, the most valuable features of the embedding features produced by MagFace model were tried to be selected. Four different optimization algorithms were used. These are PSO, GA, SCA, and DE optimization algorithms. Accuracy values of 99.83, 98.57, and 98.65 were obtained in LFW, CFP, and AGEDB data sets, respectively. It was observed that the PSO algorithm made a more successful selection in LFW, CFP, and AGEDB data sets. Moreover, the same or higher accuracies were obtained with approximately half of the embedding features thanks to these algorithms. In addition, a cost-effective proposal was presented thanks to the comparison of fewer features in face recognition.

The study can be used as a reference for future research. Also, valuable features can be obtained by applying different face recognition algorithms using this study. Cost-effective systems can design using fewer embedding features.

References

[1] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface:Deep hypersphere embedding for face recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 212–220, 2017.

[2] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5265–5274, 2018.

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4690–4699, 2019.

[4] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. Magface: A universal representation for face recognition and quality assessment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14225–14234, 2021.

[5] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. arXiv preprint arXiv:1411.7923, 2014.

[6] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In Workshop on faces in'Real-Life'Images: detection, alignment, and recognition, 2008.

[7] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In CVPR 2011, pages 529–534. IEEE, 2011.

[8] Ira Kemelmacher-Shlizerman, Steven M Seitz, Daniel Miller, and Evan Brossard. The megaface benchmark: 1 million faces for recognition at scale. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4873–4882, 2016.

[9] James Kennedy and Russell Eberhart. Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks, volume 4, pages 1942–1948. IEEE, 1995.

[10] John H Holland. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

[11] Seyedali Mirjalili. SCA: A Sine Cosine Algorithm for solving optimization problems. Knowledge-Based Systems, 96:120– 133, March 2016.

[12] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4):341–359, 1997.

[13] Grega Vrbancˇicˇ, Lucija Brezocˇnik, Uroš Mlakar, Dušan Fister, and Iztok Fister Jr. NiaPy: Python microframework for building nature-inspired algorithms. Journal of Open Source Software, 3, 2018.

[14] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M. Patel, Rama Chellappa, and David W. Jacobs. Frontal to profile face verification in the wild. In 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pages 1–9, 2016.

[15] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: The first manually collected, in-the-wild age database. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 1997–2005, 2017.