CMPE 58Z Introduction to Biometrics: Spring 2025 Midterm Take-Home Exam

Kutay Eroğlu, 2024700051 (kutay.eroglu@std.bogazici.edu.tr) 18 May 2025

1 Introduction

Face recognition is the task of identifying a face using a database of faces. This task is divided into two categories. If there is a claimed identity, meaning that there is a one-to-one decision to verify whether the person matches the claimed identity, the task is called face verification. On the other hand, if the decision is one-to-many, meaning that there is no claimed identity, the task is called face identification.

The focus of the study is to evaluate two state-of-the-art face recognition algorithms for face verification on the Labeled Faces in the Wild (LFW) dataset [1]. Selected methods are: ArcFace [2], and SFace [3].

The outline is as follows. Firstly, information about the evaluation dataset is presented. Then, methodology of the selected works is discussed in depth. Evaluation metrics, including the F1-score, classification accuracy, ROC AUC, and confusion matrix, are presented in Section 4. Following the metrics, misclassified examples for both methods are presented, accompanied by a brief commentary. Finally, additional remarks are included to provide further insight into the implementation of the evaluation protocol.

1.1 LFW Dataset

The Labeled Faces in the Wild (LFW) dataset [1] is a database of more than 13,000 face images of 5,749 individuals collected from the internet. Images are unconstrained, in the sense that conditions that affect the characteristic of an image such as lighting, background, pose, and camera setup are not controlled. There are two common configurations of the dataset: people and pairs. Since this study focuses on evaluating methods for face verification, the latter–which includes labeled image pairs–was selected.

2 ArcFace

2.1 Overview

A commonly utilized function in deep face recognition tasks is the loss, which is presented in Figure 1. One downside of this function for our task is the lack of optimization of feature embeddings to impose inter-class diversity, or intra-class compactness, which leads to drop in performance under large-scale tests, and intra-class variations (e.g., pose, age gaps).

$$L_1 = -\log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{N} e^{W_j^T x_i + b_j}},$$

Figure 1: Softmax loss function. Source: Adapted from [2].

To simplify the softmax function, bias is fixed $b_j = 0$. Following this, the logit is transformed as $W_j^\top x_i = \|W_j\|\|x_i\| \cos\theta_j$. ($\theta_j$ corresponds to the angle between feature x_i and weight W_j). The individual weight $\|W_j\|$ is fixed to 1 using $L2$ normalization. In addition to this, the embedding feature $\|x_i\|$ is fixed by $L2$ normalization and re-scaled to s. As a result of normalizing the weights and features, predictions depend solely on the angle between the weight and the feature. This version of the loss is referred to as `Norm-Softmax`, for which the equation is presented in Figure 2.

$$L_2 = -\log \frac{e^{s \cos \theta_{y_i}}}{e^{s \cos \theta_{y_i}} + \sum_{j=1, j \neq y_i}^{N} e^{s \cos \theta_j}}.$$

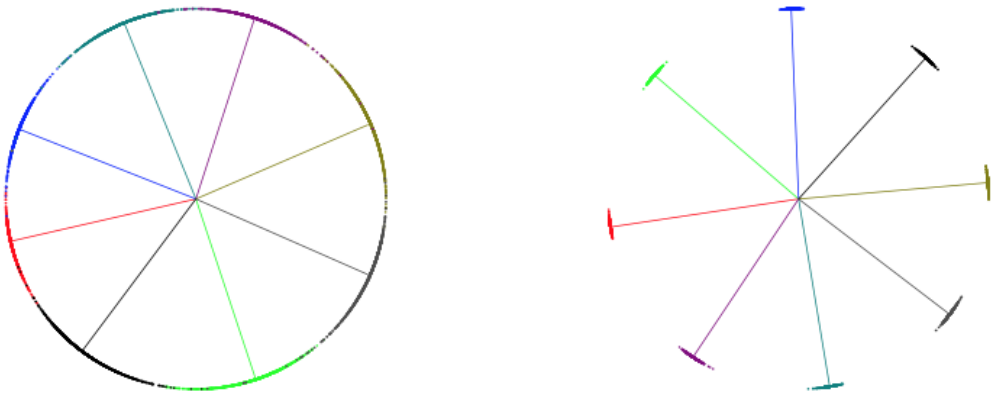
Figure 2: Norm-Softmax loss function. Source: Adapted from [2].

To increase compactness of intra-class samples and discrepancy of inter-class samples, an additive angular margin penalty m is introduced between x_i and W_{y_i}. The rationale behind this decision is that embedding features are spread around the central points of each feature on the hypersphere. The loss obtained as a product of these modifications is referred to as `ArcFace`, with the formal equation depicted in Figure 3.

$$L_3 = -\log \frac{e^{s \cos(\theta_{y_i} + m)}}{e^{s \cos(\theta_{y_i} + m)} + \sum_{j=1, j \neq y_i}^{N} e^{s \cos \theta_j}}.$$

Figure 3: ArcFace loss function. Source: Adapted from [2].

To compare the ability to enforce margins between nearest classes, 2-D feature embedding networks were trained with `Norm-Softmax` and `ArcFace` loss. Based on feature normalization introduced in `Norm-Softmax`, all face features are spread around the arc space, as illustrated in Figure 4. However, despite being able to generate separable feature embeddings, `Norm-Softmax` introduces significant ambiguity in decision boundaries. In contrast, as presented in Figure 4, `ArcFace` is able to introduce a considerable margin between the nearest classes.



(a) Norm-Softmax (b) ArcFace

Figure 4: Comparison of Norm-Softmax and ArcFace loss on 8 identities with 2D features. Samples are represented with dots, while center direction of each identity corresponds to the lines. Source: Adapted from [2].

2.2 Numerical Stability

To obtain various high-performance target logit curves (shown in Figure 5), two distinct margin penalties are combined with `ArcFace`: multiplicative angular (m_1)[4, 5] and additive cosine (m_3)[6, 7]. This combined framework results in the loss illustrated in Figure 6 where m_1, m_2, m_3 are hyperparameters, and $(\cos(m_1\theta + m_2) - m_3)$ corresponds to the unified margins.

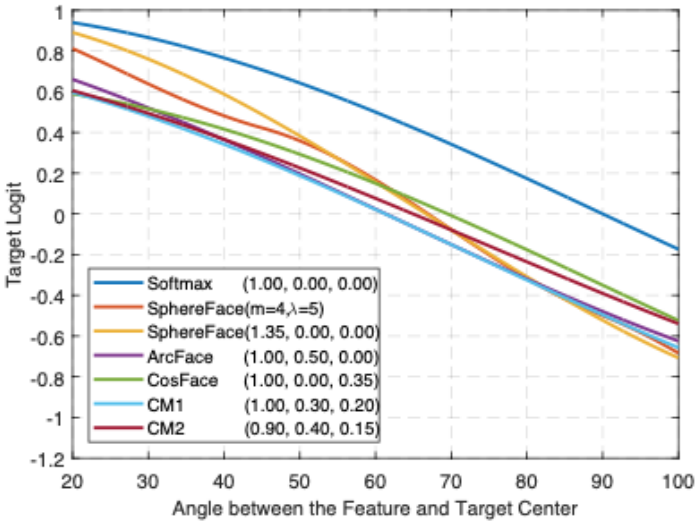


Figure 5: Target logit curves for softmax, relevant works, ArcFace, and combined margin penalty. Source: Adapted from [2].

$$L_4 = -\log \frac{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)}}{e^{s(\cos(m_1\theta_{y_i}+m_2)-m_3)} + \sum_{j=1,j\neq y_i}^{N} e^{s\cos\theta_j}}.$$

Figure 6: Combined margin loss. Source: Adapted from [2].

2.3 Geometric Difference

While `ArcFace` and prior methods are numerically similar, the proposed additive angular margin m improves the geometric attribute, due to direct correlation with the geodesic distance[8]. This improvement is made more evident by comparing the decision boundaries of `ArcFace` and prior methods in the binary classification setting, as shown in Figure 7.

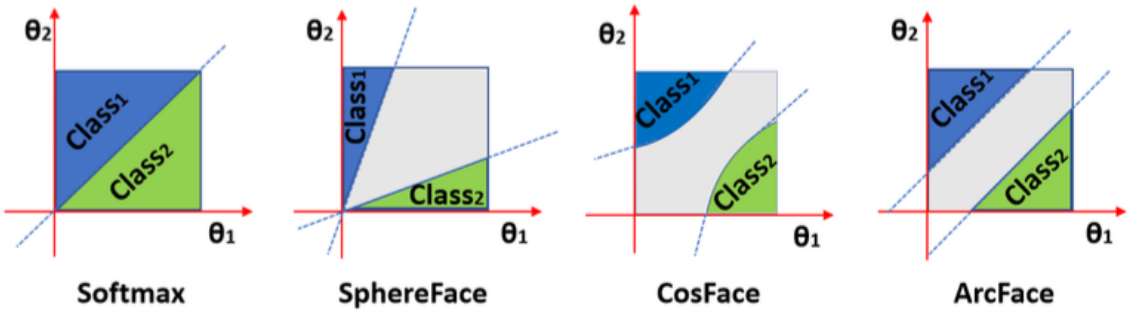


Figure 7: Decision margins of ArcFace loss and prior methods under binary classification setting. The decision boundary is represented by the dashed line, while the decision margins are shown as grey areas. Source: Adapted from [2].

3 SFace

3.1 Overview

To optimize inter-class and intra-class distances to some degree, and improve the generalization capability of face recognition models, a novel loss function called sigmoid-constrained hypersphere loss (SFace[3]) is

proposed by the authors. Building on prior work [2, 4, 6] which demonstrated the benefits of enforcing discrimination by deep face features on a hypersphere manifold, this approach restricts the direction of gradients by mapping deep face features to a hypersphere manifold, and optimizing cosine similarity. This restriction enforces the moving directions of target centers and samples to always be along the tangent of the hypersphere, as illustrated in Fig 8.

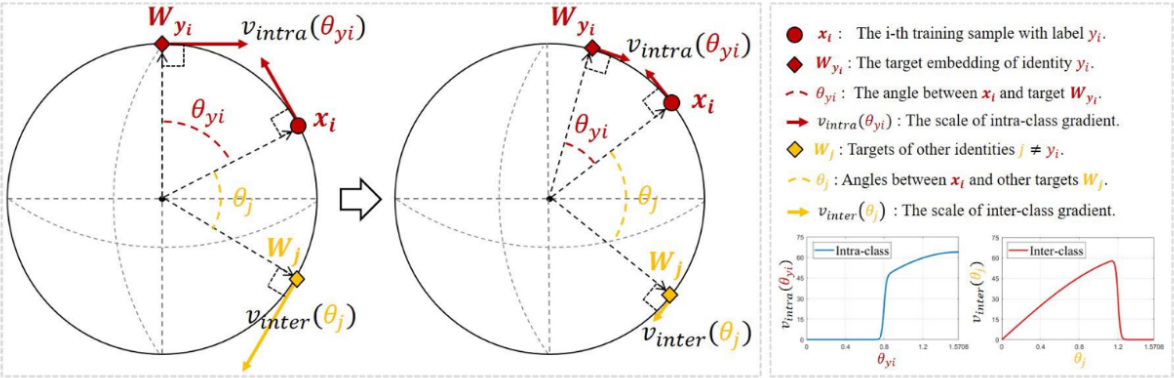


Figure 8: Visualization of the sigmoid-constrained hypersphere loss. Source: Adapted from [3].

3.2 Sigmoid-Constrained Hypersphere Loss

Given that the embedding of the i-th training sample in d-dimensional Euclidean space is denoted by $\mathbf{x}_i \in \mathbb{R}^d$; label of $\mathbf{x}_i$ by $\mathbf{y}_i$; weight of the last fully connected layer by $W = \{W_1, W_2, \ldots, W_C\} \in \mathbb{R}^{d \times C}$, where the number of identities included in training is denoted by C; target center feature of identity $\mathbf{y}_i$ is represented by $W\,y_i \in \mathbb{R}^d$.

Since the primary goal is to increase inter-class distance and decrease the intra-class distance in a moderate manner, formulation of sigmoid-constrained hypersphere loss (SFace) of $\mathbf{x}_i$ can be viewed as LSFace $= L_{\text{intra}}\,(\theta_{y_i}) + L_{\text{inter}}\,(\theta_j)$, θ_{y_i} representing the angular distance between $\mathbf{x}_i$ and $W\,y_i$; θ_j, the angular distance between $\mathbf{x}_i$ and $W\,j$, under the condition that $j \neq y_i$. The formulation of $L_{\text{intra}}(\theta_{y_i})$ and $L_{\text{inter}}(\theta_j)$ is presented in Figure 9.

$$L_{intra}\left(\theta_{y_i}\right) = -[r_{intra}\left(\theta_{y_i}\right)]_b \cos\left(\theta_{y_i}\right),$$
$$L_{inter}\left(\theta_j\right) = \sum_{j=1,\ j\neq y_i}^{C} [r_{inter}\left(\theta_j\right)]_b \cos\left(\theta_j\right).$$

Figure 9: Intra-class and inter-class loss formulas. Source: Adapted from [3].

The degree of control over the optimization speed, illustrated by the 2-D graphs in Figure 8, for intra-class and inter-class respectively, is enhanced by the introduction of two functions: $r_{\text{intra}}(\theta_{y_i})$ and $r_{\text{inter}}(\theta_j)$.

$$\frac{\partial L_{SFace}}{\partial \boldsymbol{x_i}} = -[r_{intra}\left(\theta_{y_i}\right)]_b \frac{\partial \cos\left(\theta_{y_i}\right)}{\partial \boldsymbol{x_i}} + \sum_{j=1,\, j\neq y_i}^{C} [r_{inter}\left(\theta_j\right)]_b \frac{\partial \cos\left(\theta_j\right)}{\partial \boldsymbol{x_i}},$$

Figure 10: Backward propagation process of SFace. Source: Adapted from [3].

These functions are used to re-scale intra-class and inter-class objectives respectively. Both functions

are modulated using the block gradient operator $[\cdot]_b$, which works as a unit that blocks the contribution of its input to the gradient computation. For clarity, the backward propagation process is presented in Figure 10 which shows the effect of block gradient operator on intra-class and inter-class re-scaling functions.

3.3 Gradient Re-Scale Function

The chosen gradient re-scaling functions are presented in Figure 11, where the upper asymptote of the two sigmoid curves is denoted by s; control parameter for slope of the sigmoid curves by k; parameters that suppress moving speed by controlling the horizontal intercept of the two sigmoid curves by a and b. It should be noted that a and b are vital parameters that affect the performance of the model, and should be selected based on the training set characteristics.

$$r_{intra}\left(\theta_{y_i}\right) = \frac{s}{1 + e^{-k*\left(\theta_{y_i} - a\right)}},$$
$$r_{inter}\left(\theta_j\right) = \frac{s}{1 + e^{k*\left(\theta_j - b\right)}}.$$

Figure 11: Intra-class and inter-class gradient re-scaling functions. Source: Adapted from [3].

4 Evaluation Metrics

Experiments demonstrate that aligning input face images improves the recognition accuracy by 6% [9]. Therefore, models were evaluated with and without face alignment. Consequently, 4 sets of metrics were obtained. For each model, only the best-performing option is presented. For ArcFace [2], face alignment on input images improved performance metrics, whereas for SFace [3], it had the opposite effect.

	best_accuracy_score
Human-beings	97.5
ArcFace	96.9
SFace	90.4

Figure 12: Comparison of classification accuracies

Both methods [2, 3] report accuracy that exceeds 99.5% on the same dataset, in their original works. However, as shown in Figure 12, both methods exhibit significantly lower performance in our evaluation. There are two primary factors that contribute to this discrepancy.

First, the evaluation procedure differs: While the standard protocol for evaluation on LFW involves 10-fold cross-validation, our setup utilizes only a single fold retrieved via `sklearn.datasets.fetch_lfw_pairs` [10]. Second, there are differences in the overall face recognition pipeline due to our dependence on implementation of Serengil et. al [9], which can substantially impact performance.

The confusion matrices presented in figs. 13 and 14, along with Figure 15—which displays AUC values of 0.99 and 0.96, and F1-scores of 0.96 and 0.90 for ArcFace [2] and SFace [3], respectively—show that both models perform reasonably well in the evaluation set. Nonetheless, the performance drop for both methods, specifically SFace[3], could be investigated to gain more information on the entire facial recognition pipeline.

5 Misclassified Examples

Even though there are multiple pairs that are misclassified by both methods, disjoint sets were selected to improve the diversity of the examples. All misclassified pairs, presented in figs. 16 to 19, exhibit common attributes. Subjectively, the most apparent of these is the variation in pose. In addition, skin color also

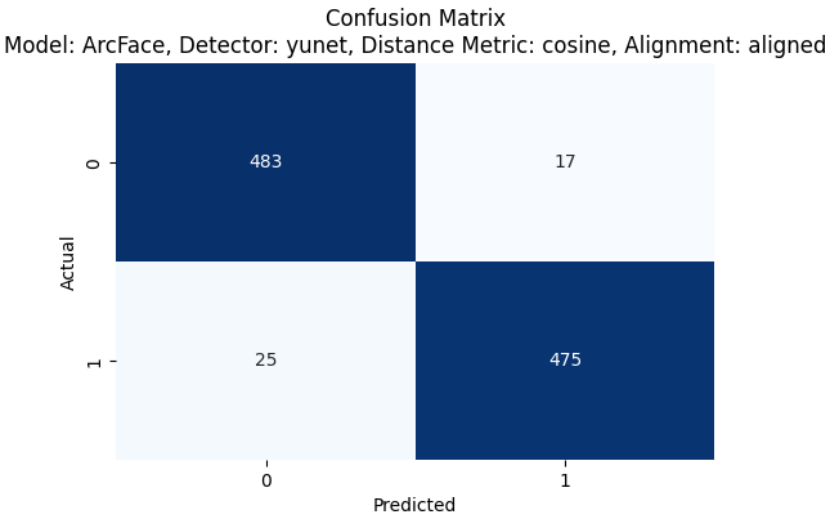


Figure 13: Confusion Matrix for Arcface with face alignment

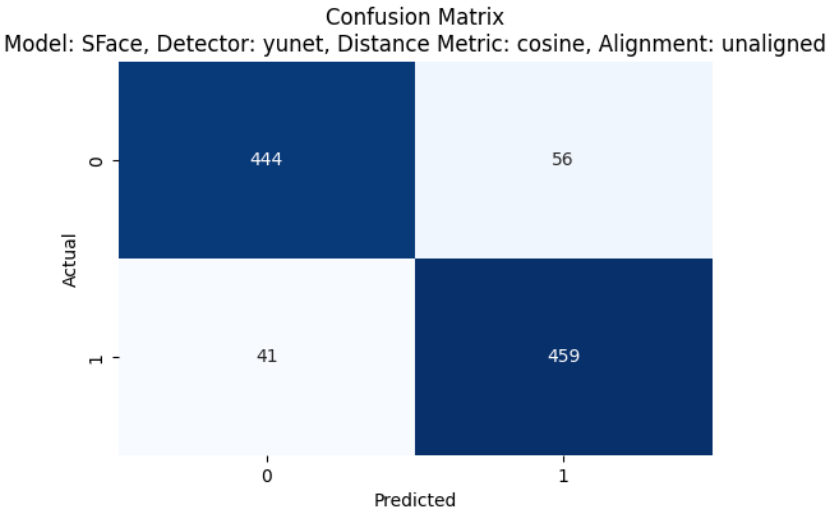


Figure 14: Confusion Matrix for SFace without face alignment

varies, possibly due to differences in illumination. All of these variations increase the complexity of the task, increasing the possibility of an instance being misclassified.

6 Additional Remarks

The source code used for the evaluation procedure leverages the work of Serengil et al. [9]. The implementation of obtaining the F1 score, determining the optimum distance-threshold, and plotting misclassified samples were developed from scratch.

References

[1] LFW dataset. This dataset contains labeled face images collected from the web with names of the people in the images as the labels, 2025. [Online; accessed 18-May-2025].

[2] 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.

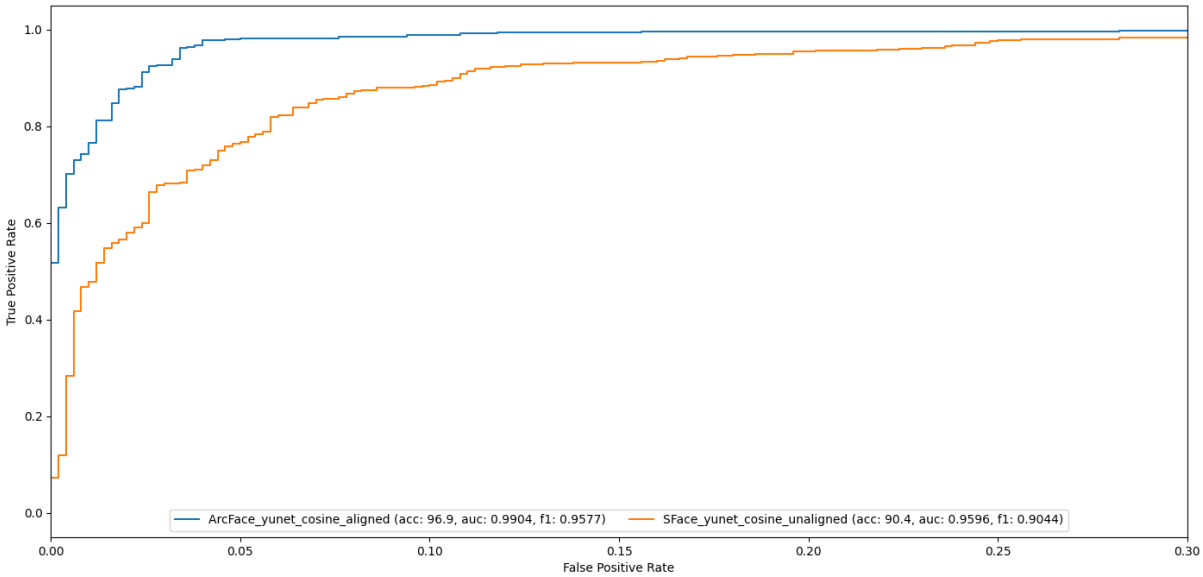


Figure 15: ROC Curve and F1-score for best performing models

Figure 16: Example misclassified sample pair for Arcface with input alignment

[3] Yaoyao Zhong, Weihong Deng, Jiani Hu, Dongyue Zhao, Xian Li, and Dongchao Wen. Sface: Sigmoid-constrained hypersphere loss for robust face recognition. *IEEE Transactions on Image Processing*, 30:2587–2598, 2021.

[4] 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.

[5] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. *arXiv preprint arXiv:1612.02295*, 2016.

[6] 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.

[7] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930, 2018.

[8] Wikipedia. Geodesic — Wikipedia, the free encyclopedia, 2025. [Online; accessed 18-May-2025].

[9] Sefik Serengil and Alper Ozpinar. A benchmark of facial recognition pipelines and co-usability performances of modules. *Journal of Information Technologies*, 17(2):95–107, 2024.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and

Figure 17: Example misclassified sample pair for Arcface with input alignment

Figure 18: Example misclassified sample pair for sface without alignment

E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Figure 19: Example misclassified sample pair for sface without alignment