


## MẪU BÁO CÁO CỦA MỖI HV

Họ và tên (IN HOA)	NGUYỄN THÀNH DUY (CH2001027)
Ảnh	
Số buổi vắng	0
Bonus	11 lần comment trên Google Classroom
Tên đề tài (VN)	THÊM GÓC PHỤ CHO HỌC SÂU NHẬN DIỆN KHUÔN MẶT
Tên đề tài (EN)	ARCFACE: ADDITIVE ANGULAR MARGIN LOSS FOR DEEP FACE RECOGNITION
Giới thiệu	<ul style="list-style-type: none"><li><b>Bài toán/vấn đề mà đề tài muốn giải quyết:</b> Xây dựng phương pháp mới nhằm khắc phục các hạn chế của phương pháp cũ trong nhận dạng một hoặc nhiều khuôn mặt thay đổi liên tục với sự khác nhau về tuổi tác và ngoại hình một cách chính xác và hiệu quả nhất.</li><li><b>Lí do chọn đề tài, khả năng ứng dụng thực tế, tính thời sự</b> Mô hình Deep Convolutional Neural Networks (DCNN) thường dùng cho việc bóc tách các đặc điểm của khuôn mặt. Việc còn lại đó là khiến</li></ul>

cho các véc tơ mang đặc điểm này được phân loại tốt nhất giúp cho việc nhận diện khuôn mặt trở nên chính xác hơn.

Có hai hướng chính để thực hiện việc đó:

- Ưu tiên việc tạo thêm một mô hình phân loại như sử dụng softmax.
- Huấn luyện máy trực tiếp sử dụng những véc tơ embeddings như sử dụng hàm mất mát triplet.

Tuy rằng cả hai phương pháp đã cho kết quả rất tốt nhưng chúng vẫn còn tồn tại một số những khuyết điểm:

- Softmax: khiến cho kích thước của ma trận biến đổi tuyến tính tăng tỉ lệ với số lượng danh tính mà chúng ta muốn phân loại, đồng thời việc huấn luyện theo phương pháp này khiến mô hình phân loại khá tốt với những vấn đề phân loại kín (khi mà tập hợp đầu vào và tập hợp đầu ra có chung số lượng class) cho thấy phương pháp này là không quá thực tế khi số lượng khuôn mặt khác nhau (số lượng class) mà chúng ta cần nhận diện thường thay đổi (tuổi, dáng người).
- Hàm triplet loss: xử lý được vấn đề này nhưng nó cũng tồn tại những khuyết điểm riêng. Vì triplet loss được lấy ý tưởng từ việc so sánh 3 mẫu một lần nên với số lượng dữ liệu tăng lên, số lượng bộ 3 cũng sẽ tăng theo cấp số nhân dẫn đến số lượng vòng lặp cũng tăng đáng kể, hơn nữa, giải pháp được cho là tối ưu khi huấn luyện với triplet loss là semi-hard sample training thì khá là khó để huấn luyện hiệu quả.

Vì những khuyết điểm này mà phương pháp hàm mất mát mới Additive Angular Margin Loss (ArcFace) được đưa ra.

- **Input output:**

- **Face identification:**

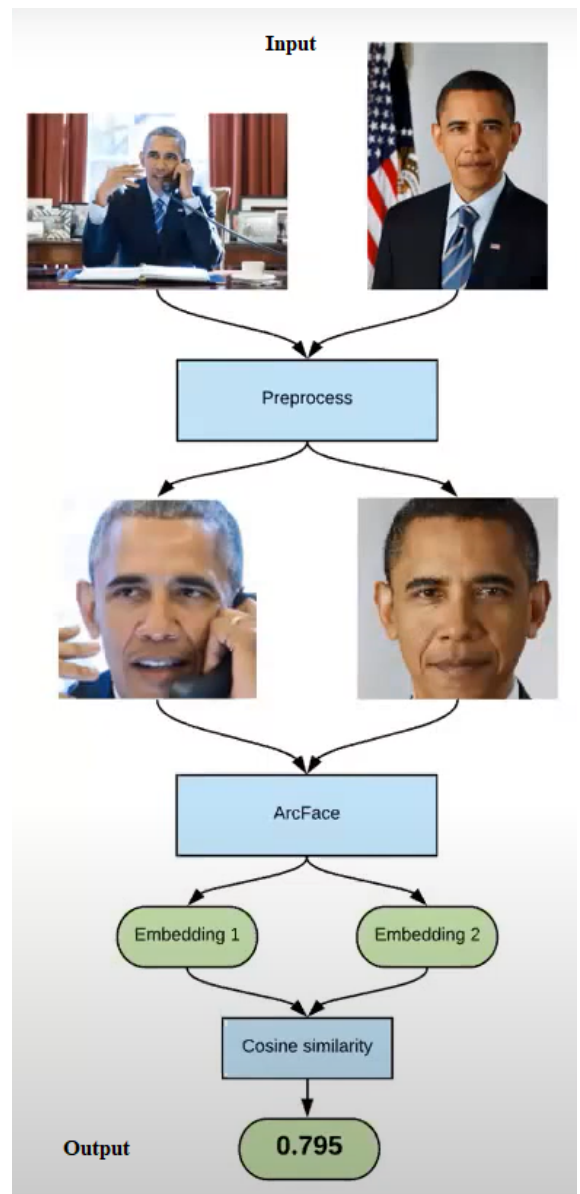
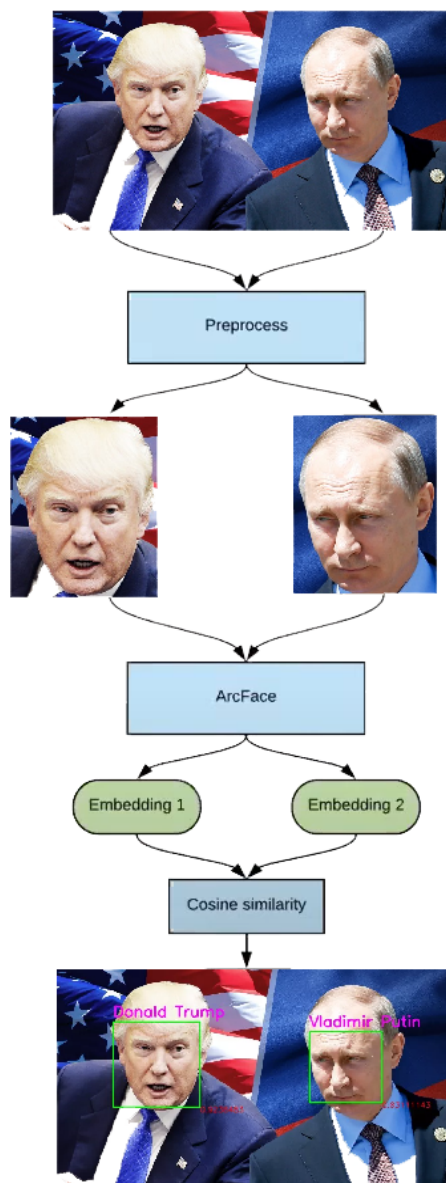
- Input:* là ảnh hoặc video của một hoặc nhiều khuôn mặt.

- Output:* là ảnh hoặc với nhãn tên của người trong ảnh.

- **Face verification:**

- Input:* 2 ảnh khuôn mặt

- Output:* 2 ảnh có cùng 1 người hay không



<b>Mục tiêu</b>	<ul style="list-style-type: none"> <li>• Xây dựng phương pháp mới để nhận diện khuôn mặt và xác thực khuôn mặt (những ảnh người khác nhau về độ tuổi, cân nặng,...) với độ chính xác cao hơn những mô hình cũ (Softmax function, Triplet loss, CosFace, SphereFace).</li> </ul>
<b>Nội dung và phương pháp thực hiện</b>	<ul style="list-style-type: none"> <li>• <b>Nội dung:</b>  <p>Trong phạm vi của đề cương này, các nội dung sẽ được thực hiện sẽ là: Tiến hành phân tích cách thức thực hiện nhận dạng khuôn mặt qua mô hình CNN với các hàm softmax. Sau đó, xây dựng dựng một hàm mới dựa trên hàm softmax này. Cuối cùng, là tiến hành so sánh hiệu quả giữa các giải thuật Softmax function, Triplet loss, CosFace, SphereFace.</p> </li> <li>• <b>Phương pháp:</b> <ul style="list-style-type: none"> <li>➤ <b>Xây dựng giải thuật:</b>  <p>Hàm mất mát Additive Angular Margin Loss có thể được xem như một sự cải tiến cho hàm softmax.</p> <p><b><i>Giải thích hình học:</i></b></p> <p>Thay vì sử dụng khoảng cách euclid, ArcFace tính toán khoảng cách trắc địa trên siêu cầu. Không gian trắc địa là không gian mà mọi khoảng cách đều được đo bằng đường ray. Đường thu được giữa hai điểm được gọi là đường trắc địa. Nó mô tả khoảng cách ngắn nhất giữa các điểm, còn được gọi là khoảng cách trắc địa. Trong hình minh họa bên dưới, chúng ta có thể thấy diễn giải hình học của ArcFace:</p> <p>(a) Sự tương ứng trực quan giữa góc và lẽ cung. Lẽ góc của ArcFace tương ứng với lẽ cung, khoảng cách trắc địa trên bề mặt siêu cầu.</p> <p>(b) Các điểm màu xanh lam và xanh lục biểu thị các tính năng nhúng từ hai lớp khác nhau. ArcFace có thể áp đặt trực tiếp lẽ góc (vòng cung) giữa các lớp.</p> </li> </ul> </li> </ul>



Giải thích toán học:

☐ Công thức hàm softmax:

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

$x_i$  denotes the deep feature of the  $i$ -th sample, belonging to the  $y_i$ -th class.  
 $W_j^T$  denotes the  $j$ -th column of the weight  $W$  and  $b_j$  is the bias term.  
 The batch size and the class number are  $N$  and  $n$ , respectively.

☐ Công thức hàm ArcFace:

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

where  $\theta_j$  is the angle between the weight  $W_j$  and the feature  $x_i$   
 $s$  - feature scale, the hypersphere radius  
 $m$  - angular margin penalty

Ta thấy, sự khác biệt duy nhất giữa hai hàm mất mát là logit. Ta có:

$$\cos \theta = W^T \cdot x$$

Để lấy ra theta ta có công thức hàm lượng giác:

$$\theta = \arccos(\cos(\theta))$$

Bây giờ, chúng ta thêm lề vào góc, ví dụ, 0,5, và tính cosin của nó. Cuối cùng, chúng tôi nhân nó với  $s$ , bán kính trên hypersphere, chẳng hạn, 30 và chúng tôi nhận được logit của ArcFace:

$$\theta + m$$

$$s * (\cos(\theta + m))$$

*Quy trình training một DCNN bằng ArcFace:*

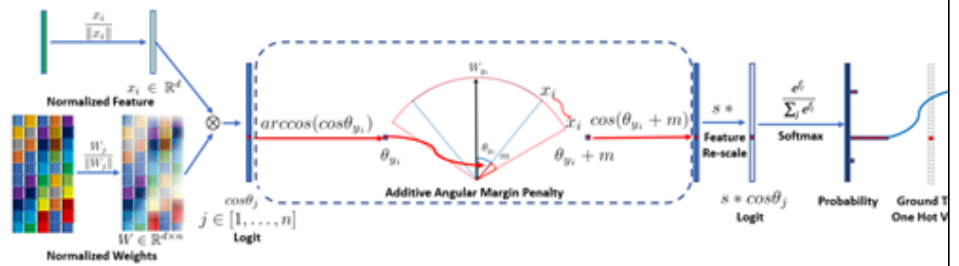


Figure 2. Training a DCNN for face recognition supervised by the ArcFace loss. Based on the feature  $x_i$  and weight  $W_j$ , we get the  $\cos \theta_j$  (logit) for each class as  $W_j^T x_i$ . We calculate the  $\arccos(\cos \theta_{y_i})$  and get the angle between the feature  $x_i$  and weight  $W_{y_i}$ . In fact,  $W_j$  provides a kind of centre for each class. Then, we add an angular margin penalty  $m$  on the angle  $\theta_{y_i}$ . After that, we calculate  $\cos(\theta_{y_i} + m)$  and multiply all logits by the feature scale  $s$ . The logits then go through the Softmax function and contribute to the cross entropy loss.

Chi tiết các bước như sau:

B1: Sau khi normalization weights và feature vectors, ta lấy được  $\cos \theta_j$  với  $j = 1, 2, \dots, C$   $\forall j = 1, 2, \dots, C$  ( $C$  là số class).

B2: Ta cần tính  $\theta_j$  (arccos là được).  $\theta_j$  là góc giữa ground truth weight  $W_{y_i}$  và feature vector  $x_i$

B3: Sau đó ta tính  $\cos(\theta + m)$ . Nếu bạn còn nhớ vòng tròn lượng giác, thì trong khoảng từ 0 đến  $\pi$ , góc càng tăng cos càng giảm.

B4: Tính  $s * \cos(\theta + m)$ . Sau đó đưa vào softmax để lấy ra phân phối xác suất probability của các nhãn.

B5: Cuối cùng, ta có ground truth vector (là label đã được one-hot) cùng probability, đóng góp vào cross entropy loss.

➤ **So sánh với các giải thuật:**

Chọn khoảng chừng 1500 ảnh cho 1 khuôn mặt với tổng cộng 8 khuôn mặt khác nhau để huấn luyện mạng với lần lượt hàm Softmax function, Triplet loss, CosFace, SphereFace và hàm mất mát ArcFace. Tiến hành so sánh các kết quả đạt được.

**Kết quả dự kiến**

● *Phần mềm ứng dụng*

Hoàn thành ứng dụng nhận diện khuôn mặt người qua camera trong quán net với python.

- *Thuật toán*

Hoàn thành thuật toán với phương pháp được đưa ra.

- *So sánh giữa các phương pháp:* Softmax function, Triplet loss, CosFace, SphereFace. Dự kiến phương pháp ArcFace có kết quả phải tốt hơn hoặc bằng kết quả cao nhất của các phương pháp này với 3 bộ dữ liệu LFW, CFP-FP, AgeDB-30

Hàm Loss	LFW	CFP-FP	AgeDB-30
ArcFace(0.5)	$\geq 99.51$	$\geq 95.44$	$\geq 94.56$
SphereFace (1.35)	99.11	94.38	91.70
CosFace (0.35)	<b>99.51</b>	<b>95.44</b>	<b>94.56</b>
Softmax	99.08	94.39	92.33
Triplet (0.35)	98.98	91.90	89.98

- *Bộ dữ liệu*

Datasets	#Identity	#Image/Video
LFW	5,749	13,233
CFP-FP	500	7,000
AgeDB-30	568	16,488

**Tài liệu  
tham  
khảo**

- [1] <http://data.mxnet.io/models/>. 8
- [2] <http://trillionpairs.deepglint.com/overview>. 2, 4, 5, 8
- [3] Stanford cs class cs231n: Convolutional neural networks for visual recognition. <http://cs231n.github.io/neural-networks-case-study/>. 9
- [4] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv:1603.04467, 2016. 2
- [5] J. S. Brauchart, A. B. Reznikov, E. B. Saff, I. H. Sloan, Y. G. Wang, and R. S. Womersley. Random point sets on the spherehole radii, covering, and separation. Experimental Mathematics, 2018. 10
- [6] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In FG, 2018. 1, 2, 4, 5, 7, 8

	<ul style="list-style-type: none"> <li>• [7] B. Chen, W. Deng, and J. Du. Noisy softmax: improving the generalization ability of dcnn via postponing the early softmax saturation. In CVPR, 2017. 2</li> <li>• [8] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv:1512.01274, 2015. 2, 5</li> <li>• [9] J. Deng, Y. Zhou, and S. Zafeiriou. Marginal loss for deep face recognition. In CVPR Workshop, 2017. 2, 6</li> </ul>
--	--