Face Identification System Using Convolutional Neural Network for Low Resolution Image

Muhammad Arafah[1,5], Andani Achmad[2], Indrabayu[3], Intan Sari Areni[4]
[1]Doctoral Student of Electrical Engineering Department
[2,4] Electrical Engineering Department
[3]Informatics Department
Universitas Hasanuddin, Makassar, Indonesia
[5]Study Program Informatics
STMIK AKBA, Makassar, Indonesia
arafah@akba.ac.id, {andani, indrabayu, intan}@unhas.ac.id

***Abstract* — This research aims to determine the performance of face identification on closed circuit television (CCTV) cameras. There are two data classifications used, namely training data and testing data. The training data use the CASIA-Webface dataset. Meanwhile, the testing data consists of two data, namely the source data and the target data. The source data are form of photos taken using a Digital Single Lens Reflex (DSLR) camera, while the target data use video data taken with CCTV. The source data consists of 10 IDs, each of them has 1 image for each size, so the total images used in the source data are 50 IDs. While the target data are 20 IDs, each of them has 20 face images with low resolution characteristics, less light and face capture not parallel to CCTV. This research uses Convolutional Neural Network (CNN) method with ResNet50 architecture, ArcFace as a loss function in the training process and Cosine Similarity for the face identification process. ResNet50 and ArcFace use an embedding size of 512 and in the training process, ArcFace's scale and margin parameters are 64 and 0.5. The results indicate differences in accuracy, True Positive Rate (TPR) and False Positive Rate (FPR) from the face identification process between the image sizes used and the respective IDs in the source data. The method used had the highest performance for face image identification scenarios of 128 x 128 pixels with accuracy and FPR reached 99.30% and 0.02%. From the TPR, the method used had high performance at size of 512 x 512 pixels namely 91.50%.**

Keywords* — *Face identification, Low resolution, Convolutional Neural Network, ResNet50, ArcFace, Cosine similarity, Performance.

I. INTRODUCTION

Video surveillance systems have received more attention in recent years, this is because the increasing demand for security and safety, especially in public services such as airports, train stations, bus terminals, shopping malls, educational institutions, hospitals, hotel waiting rooms, etc. The installation of CCTV cameras is usually aimed for recording, monitoring and reviewing incidents that may occur. Comparing to still images, the use of video contains more information, both temporal and multi-view, in addition, video is more extensive on the aspects of security and law enforcement. Surveillance systems combined with identifying people on surveillance video cameras are an important task in face recognition systems [1], [2], [3].

Face image identification is one of the most frequently used biometric technologies. Face identification has become a research topic for researchers in the fields of pattern recognition, computer vision and artificial intelligence. The utilization of convolutional neural networks (CNN) has achieved great success in several research topics such as object classification, scene understanding and action recognition. Then the most important thing is that CNN has experienced tremendous improvements in face recognition performance. The current accuracy of the use of face recognition algorithms has reached more than 98% using the CNN-based model, the use of ArcFace on CNN as a loss function has the highest accuracy compared to others [4], [5], [12].

Face recognition systems using CCTV often handle low resolution images due to the long distance between the CCTV and the target. The pictures taken usually have poor quality, inadequate lighting conditions and the installation of a CCTV camera that is not parallel to the face. Another problem with the image condition is the difference in the resolution of the CCTV image with the photo used as a reference in the identification process [6], [7], [15].

A research conducted by M Arafah, et al. determined the best distance for the CCTV installation from the passenger inspection area at the airport for face identification cases. The results indicate that for a CCTV camera with a height of 250 cm had the best distance of 300 cm from the target position [9]. The data collection scenario in that research is used as a reference for data collection in this research. The face images obtained with this scenario have a face image resolution ranging from 32 x 32 pixels. In addition, the focus of that research was the best distance for CCTV, while this research focuses on determining the performance of face identification.

II. PROPOSED METHOD

The system design used in this research can be seen in Figure 1, there are generally two stages carried out, namely the training stage and the testing stage.

A. Training

In the training stage, the first step taken is to determine the dataset to be used, this research used the CASIA-WebFace dataset with a total of 490.623 face images divided into 10.572 classes. The dataset is used as a reference for determining the initial trained model using the CNN method with the ResNet50 architecture with an embedding size of 512.

For the ArcFace process, it is a loss function in the training process using an embedding size of 512, while the size for scale parameters is 64 and a margin of 0.5. In the epoch section, an increment process is carried out after each iteration process is completed. To determine the optimization process, the training model is limited to epoch 120. Furthermore, the training model dataset will be used in the testing process.

B. Testing

The testing stage is classified into two parts, namely Source Data and Target Data, the collection of Source Data and Target Data refers to research [8].

Source data are the form of face images taken using a DSLR camera. Each face image is then processed with various sizes such as 512 x 512, 256 x 256, 128 x 128, 64 x 64 and 32 x 32 pixels. In the Source Data, there are 10 IDs, each of them has 1 photo for each size. Source data used can be seen in Figure 2.

For the Target Data, the devices used in data collection are CCTV cameras. The device is mounted at a height of 250 cm and positioned 300 cm from the area the target will pass. Video data from CCTV cameras contains 20 IDs, 10 of them have the same ID as the Source Data. Face image from 20 IDs extracted from video data will be used as Target Data. Target data are divided into 5 different sizes such as 512 x 512, 256 x 256, 128 x 128, 64 x 64 and 32 x 32 pixels. Figure 3 shows an example of a face image of the Target Data.

In the testing process, the photos in the Source Data will be used as a reference to find photos that are the same as the data contained in the Target Data. This process will use cosine similarity. The results of using cosine similarity will later determine that the photos with the Source Data are different or the same as those contained in the Target Data. This process is described in more detail in the Performance Evaluation section.

C. ArcFace

The loss function commonly used in the classification process is Softmax Loss, which has the following formula [9], [10], [11], [12]:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{w_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{w_j^T x_i + b_j}}, \quad (1)$$

Where $x_i \in \mathbb{R}^d$ is a feature of the sample to -i in class y_i dan d is an embedding dimension measuring 512. $W_j \in \mathbb{R}^d$ is the weight of the class to-j, and $b_j \in \mathbb{R}^n$ is the bias from the class to-j. Then N is the number of samples that will be used, and n is the number of classes. To optimize the Softmax Loss which can provide a high degree of feature similarity between samples belonging to the same class while increasing the difference in features between different classes, the research [6] proposes the ArcFace method.

$$L_{ArcFace} = -\frac{1}{N}\sum_{i=1}^{N} log \frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(Cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}} \quad (2)$$

Where $cos\,\theta_j$ is the result of $\frac{W_j}{\|W_j\|} \otimes \frac{x_i}{\|x_i\|}$. θ_{y_i} is resulted from $\arccos(cos\,\theta_{j=y_i})$. m is angular margin penalty for θ_{y_i}.

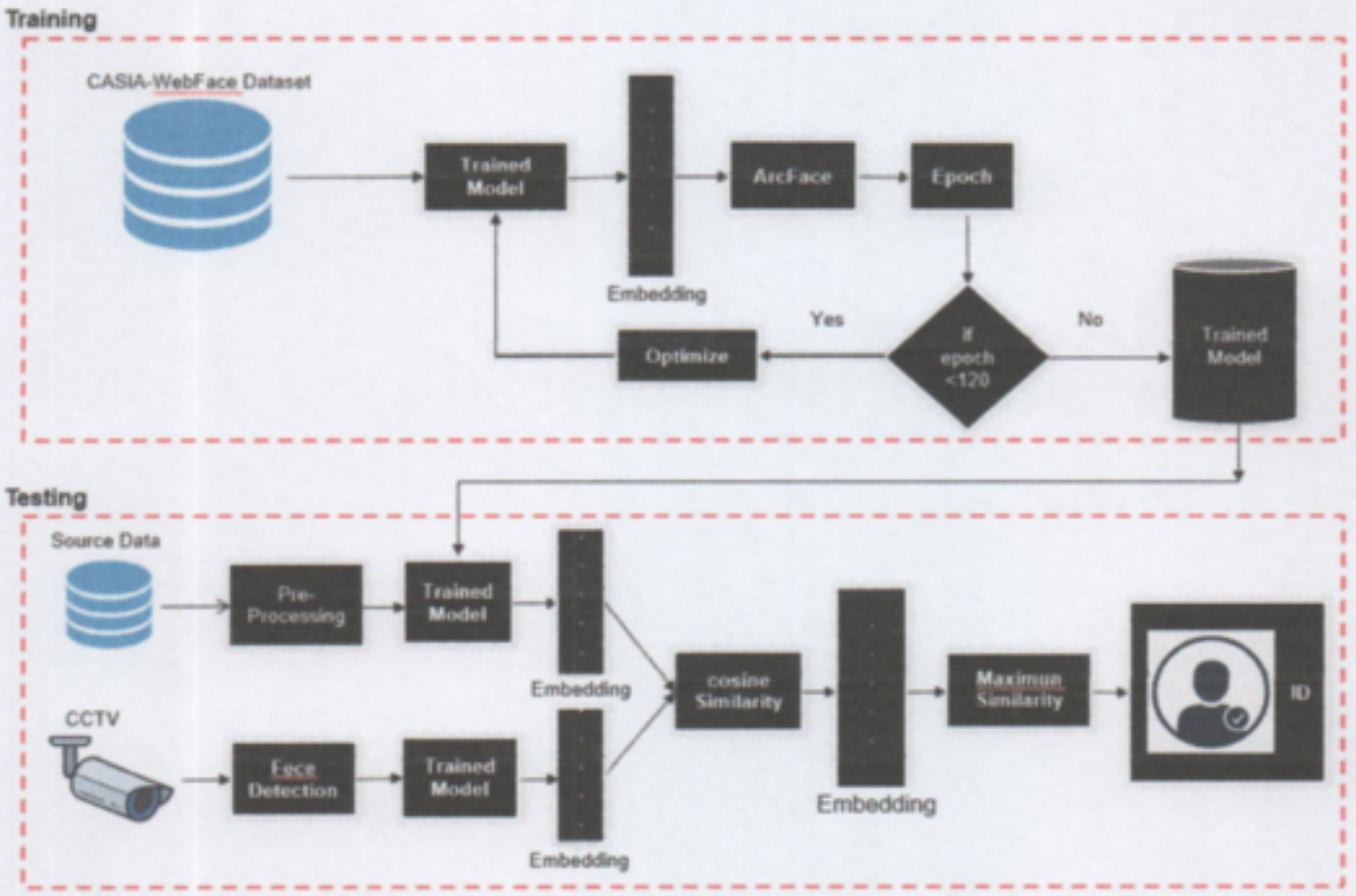

Figure 1. System Design

After cos $(\theta_{y_i} + m)$ and $cos\ \theta_j$ are counted, the result will then be re-scaled by the s.

D. Cosine Similarity

Cosine similarity (CosSim) is a method used to calculate the level of similarity of two vectors. The equation between two vectors A and B can be seen in the formula (3) [13].

$$CosSim(X,Y) = \frac{X^T Y}{\|X\| \|Y\|} \quad (3)$$

Where $CosSim(X,Y)$is Similarity level, X is Vector X, its similarities will be compared, Y is Vector Y, its similarities will be compared, $\|X\|$ is length of vector X, $\|Y\|$ is length of vector Y. Cosine Similarity has special properties that make it suitable for metrics, the resulting similarity measure is always in the range −1 and +1. This allows the objective function to be simple and effective.

E. Performance Evaluation

In the evaluation process, the trained ResNet50 model is used to calculate the embedding value of the face images in the source data and target data (s_a, t_b). The two embedding values

Figure 2. Test data face images for source data

Figure 3. Test data face images for target data

are calculated for their similarity using formula (3). All face image pairs from the source data and the target data (a, b) that have the same identity are represented by the symbol P_{same}. in the pairs of images with a different identity with P_{diff}. The face image pairs that are correctly recognized, have the same identity defined in formula (4). The pairs of pictures with different identities identified as having the same face are defined in formula (5). The threshold value d is used as a condition to determine whether the two embeds have the same or different identities. If the similarity value resulted is greater than the threshold, then both embedding are said to have the same identity and vice versa [14].

$$TA(d) = \{(a,b) \epsilon P_{same}, with\ CosSim(s_a, t_b) \geq d\} \quad (4)$$

$$FA(d) = \{(a,b) \epsilon P_{diff}, with\ CosSim(s_a, t_b) \geq d\} \quad (5)$$

The True Positive Rate TPR (d) and False Positive Rate FPR (d) values are calculated using the formulas (6) and (7) [14].

$$TPR(d) = \frac{TA(d)}{P_{same}} \quad (6)$$

$$FPR(d) = \frac{FA(d)}{P_{diff}} \quad (7)$$

III. Result and Discussion

This research uses the CNN method based on the ResNet50 architecture, then uses ArcFace as a loss function for the training process, then utilizes the cosine similarity in the face identification process. The embedding size used in ResNet50 and ArcFace in the training process is 512, then the parameters used in ArcFace are scale with a size of 64 and margin with a size of 0.5.

The results refer to three aspects, namely accuracy, TPR and FPR, to measure the performance level of face image identification. There are two scenarios used in this study, namely: first, calculating the performance of the model at each image size, if the identification process uses one threshold value for all IDs in the Source Data. Second, calculating the performance of the model at each image size, if when identifying each ID in the Data Source, its respective threshold values are given. For the determination of the threshold value, both in the first and the second scenario, it was done by giving the threshold value on the face image starting from the lowest to the highest value, then in the process, the threshold value that had been obtained will be compared with reference to accuracy value, then one threshold value will be taken based on the best accuracy value.

The performance of the first scenario can be seen by referring to table I. The TPR value generated by the model continues to decrease if the image size used gets smaller. Meanwhile, the FPR value did not experience major changes even though the image used was getting smaller. The accuracy of the model when using face images with sizes 512, 256 and 128 pixels is not much different (comparable). Meanwhile, using face images with sizes of 64 and 32 pixels experienced a significant decrease in accuracy.

The performance of the second scenario can be seen in Table II - Table VI. Based on the results of the five tables, it is known that the model's performance has decreased on the 32 pixel face image and on certain IDs. This can be seen by referring to an ID with a TPR value below 100.The target face image that fails to be identified at a size of 32 pixels can be seen in Figure 4, while for a face image that fails to be identified at a size of 64 pixels can be seen in Figure 5.

Figure 4. Target of 32 pixel that failed to be identified

Figure 5. Target of 64 pixel that failed to be identified

TABLE I. IDENTIFICATION PERFORMANCE ON FACE IMAGES OF ALL SIZES

Image Size	Accuracy	TPR	FPR	Threshold
512	99.25	**91.50**	0.34	0.351
256	99.17	87.00	0.18	0.370
128	**99.30**	86.50	**0.02**	0.382
64	98.85	83.50	0.34	0.365
32	97.57	56.50	0.26	0.331

TABLE II. IDENTIFICATION PERFORMANCE IN 512 PIXEL FACE IMAGES

ID	Accuracy	TPR	FPR	Threshold
000	98.00	85	1.316	0.307
001	99.75	100	0.263	0.343
002	100	100	0	0.424
003	100	100	0	0.459
004	100	100	0	0.527
005	100	100	0	0.520
006	100	100	0	0.554
007	99.5	100	0.526	0.358
008	100	100	0	0.358
009	100	100	0	0.520

TABLE III. IDENTIFICATION PERFORMANCE IN 256 PIXEL FACE IMAGES

ID	Accuracy	TPR	FPR	Threshold
000	98.25	80	0.789	0.307
001	99.5	95	0.263	0.345
002	100	100	0	0.421
003	100	100	0	0.436
004	100	100	0	0.532
005	100	100	0	0.518
006	100	100	0	0.577
007	99.5	90	0	0.407
008	100	100	0	0.383
009	100	100	0	0.554

TABLE IV. IDENTIFICATION PERFORMANCE IN 128 PIXEL FACE IMAGES

ID	Accuracy	TPR	FPR	Threshold
000	97.75	80	1.316	0.279
001	99.5	100	0.526	0.359
002	100	100	0	0.427
003	100	100	0	0.466
004	100	100	0	0.517
005	100	100	0	0.522
006	100	100	0	0.531
007	99.5	90	0	0.382
008	99.75	100	0.263	0.327
009	100	100	0	0.555

TABLE V. IDENTIFICATION PERFORMANCE IN 64 PIXEL FACE IMAGES

ID	Accuracy	TPR	FPR	Threshold
000	96.50	45	0.789	0.278
001	99.5	95	0.263	0.403
002	100	100	0	0.490
003	99.75	100	0.263	0.365
004	100	100	0	0.540
005	100	100	0	0.546
006	100	100	0	0.516
007	98.5	85	0.789	0.353
008	99.25	100	0.789	0.327
009	100	100	0	0.585

TABLE VI. IDENTIFICATION PERFORMANCE IN 32 PIXEL FACE IMAGES

ID	Accuracy	TPR	FPR	Threshold
000	94.75	0	0.263	0.316
001	96.75	55	1.053	0.310
002	98.75	75	0	0.406
003	98.25	80	0.789	0.287
004	100	100	0	0.369
005	99.25	85	0	0.376
006	100	100	0	0.340
007	96.5	40	0.526	0.257
008	94.75	0	0.263	0.342
009	100	100	0	0.408

IV. CONCLUSION

The implementation of the ArcFace and Cosine Similarity methods using the ResNet50 model with 512 embedding is to determine the level of object similarity in the form of face images with two different conditions, namely, taking face images using a DSLR camera as a source data compared to face images taken using a CCTV camera as data Target. This research shows that there are differences in accuracy of TPR and FPR of the face identification process between the image sizes used and the respective IDs in the Source Data. In the first scenario, the model's performance continues to decline if the image size used is getting smaller, this can be seen from the change in the TPR value, then the FPR value generated by the model, based on image size, does not have a significant effect even though the size of the face image used is getting bigger. Whereas in the second scenario, the model's performance decreased in the face image with the smallest face image size, 32 pixels and only on certain IDs.

This research focuses on determining the performance of face identification using CCTV with low-resolution image conditions. Future research is hoped to determine the performance of face identification with facial hair or occlusion on the face area.

REFERENCES

[1] S. Afra and R. Alhajj, "Early warning system: From face recognition by surveillance cameras to social media analysis to detecting suspicious people," *Phys. A Stat. Mech. its Appl.*, vol. 540, p. 123151, 2020, doi: 10.1016/j.physa.2019.123151.

[2] D. Manju and V. Radha, "A Novel Approach for Pose Invariant Face Recognition in Surveillance Videos," *Procedia Comput. Sci.*, vol. 167, no. 2019, pp. 890–899, 2020, doi: 10.1016/j.procs.2020.03.428.

[3] C. Ding and D. Tao, "Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 1002–1014, 2018, doi: 10.1109/TPAMI.2017.2700390.

[4] G. Gao, Y. Yu, M. Yang, P. Huang, Q. Ge, and D. Yue, "Multi-scale patch based representation feature learning for low-resolution face recognition," *Appl. Soft Comput. J.*, vol. 90, 2020, doi: 10.1016/j.asoc.2020.106183.

[5] J. Sun, Y. Shen, W. Yang, and Q. Liao, "Classifier shared deep network with multi-hierarchy loss for low resolution face recognition," *Signal Process. Image Commun.*, vol. 82, no. March 2019, p. 115766, 2020, doi: 10.1016/j.image.2019.115766.

[6] M. Saad Shakeel, K. M. Lam, and S. C. Lai, "Learning sparse discriminant low-rank features for low-resolution face recognition," *J. Vis. Commun. Image Represent.*, vol. 63, p. 102590, 2019, doi: 10.1016/j.jvcir.2019.102590.

[7] C. Engineering and C. Gables, "Low Resolution Face Recognition in Surveillance Systems Using Discriminant Correlation Analysis," pp. 912–917, 2017, doi: 10.1109/FG.2017.130.

[8] M. Arafah, A. Achmad, Indrabayu, and I. S. Areni, "Face recognition system using Viola Jones, histograms of oriented gradients and multi-class support vector machine," in *Journal of Physics: Conference Series*, 2019, vol. 1341, no. 4, doi: 10.1088/1742-6596/1341/4/042005.

[9] Deng J., Guo J., Liu T., Gong M., Zafeiriou S. (2020) Sub-center ArcFace: Boosting Face Recognition by Large-Scale Noisy Web Faces. In: Vedaldi A., Bischof H., Brox T., Frahm JM. (eds) Computer Vision – ECCV 2020. ECCV 2020. Lecture Notes in Computer Science, vol 12356. Springer, Cham. https://doi.org/10.1007/978-3-030-58621-8_43

[10] H. Wang *et al.*, "CosFace: Large Margin Cosine Loss for Deep Face Recognition," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, pp. 5265-5274, doi: 10.1109/CVPR.2018.00552.

[11] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6738–6746, 2017, doi: 10.1109/CVPR.2017.713.

[12] J. Deng, J. Guo, N. Xue and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 4685-4694, doi: 10.1109/CVPR.2019.00482.

[13] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6493 LNCS, no. PART 2, pp. 709–720, 2011, doi: 10.1007/978-3-642-19309-5_55.

[14] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.

[15] H. W. Sino, Indrabayu and I. S. Areni, "Face Recognition of Low-Resolution Video Using Gabor Filter & Adaptive Histogram Equalization," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 2019, pp. 417-421, doi: 10.1109/ICAIIT.2019.8834558.