International Journal of Innovative
Computing, Information and Control
Volume **17**, Number **5**, October **2021**

ICIC International ©2021 ISSN 1349-4198
pp. 1513–1530

THE DEVELOPMENT OF FACE RECOGNITION MODEL IN INDONESIA PANDEMIC CONTEXT BASED ON DCNN AND ARCFACE LOSS FUNCTION

Wirianto and Tuga Mauritsius

Information Systems Management Department
BINUS Graduate Program – Master of Information Systems Management
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
wirianto@binus.ac.id; tmauritsus@binus.edu

Received March 2021; revised July 2021

Abstract. *The advancement of technology opens opportunities for implementation that benefits the social and economic aspects of human life. Given the latest achievement in face recognition technology that surpasses human ability to identify a face, the research explores the application of this scientific discovery in the Indonesian context during the current pandemic situation. Toward the effort to achieve this goal, the study develops an Indonesia Labelled Face in the Wild (ILFW) that collects face images of famous Indonesian people from the Internet in various poses, expressions, lighting/illumination, and fashion attribute. In response to the recent COVID-19 pandemic situation, the study also augmented a face mask to a portion of collected face images. Using DCNN, RetinaFace as the face detection model, and Arcface loss function, and adopting CRISP-DM, the research contributes by providing a method to develop a face dataset with 1,200 identities, and face recognition model with 92 percent accuracy and be able to recognize Indonesian people with a face mask. The researchers also recommend use cases for real-time face recognition in the business organization. It uses CCTV to perform automatic attendance, security surveillance, and employee location tracking and exhibits deployment consideration. Future research could increase the accuracy of face recognition model by adding more identities to the face dataset.*

Keywords: Face recognition, Deep convolutional neural network, ResNet, Arcface, CRISP-DM

1. **Introduction.** Throughout history, the advancement of science and technology has brought a positive social and economic impact on humankind [1] by solving daily problems [2], transformed and disrupted the organization, businesses, and individuals, enabling them to innovate and carry out activities in more effective and efficient ways [3,4]. Among the many nowadays, Artificial Intelligence (AI) is considered as a progressive and popular science sub area. It possesses the potential to transform the world as we know it by representing humans in dealing with and addressing daily tasks and challenges in much more effective and efficient way [5,6]. Within the AI space, there is an innovation called computer vision that allows a machine/computer to capture and understand the information in an image or video [7]. One of its applications is the ability to recognize human identity through a picture of a face [8,9]. Face recognition technology has rapidly advanced such that it manages to surpass the accuracy of face recognition by a human [10,11]. Arcface by Deng et al. [10] is a loss-function based on DCNN model with face

DOI: 10.24507/ijicic.17.05.1513

feature discrimination improvement as its novelty, resulting in 99.83 percent of face recognition accuracies. It also demonstrates a high-performance real-time face recognition.

To benefit from a face recognition technology in a business and social context, a specific method and technique should be developed that considers its locality [12]. In the Indonesian context, the face recognition model should consider diversities within the Indonesian people, such as multi-race/ethnicity, skin colors, face contours, functional fashion accessories, glasses, and religious-related fashion attributes. In the current COVID-19 pandemic context, the model also should be able to recognize a face using a medical mask.

In Indonesia's frame of reference, several recent studies have covered the explanation of methods and techniques to develop face recognition model. Naufal et al. have attempted to develop a face recognition model using deep learning technology for attendance system [13]. Another study by Andiani and Soewito used a pre-trained FaceNet based face recognition model for work attendance [14]. Chowanda and Sutoyo have tried to develop a face recognition model based on convolutional neural network in mobile phone environment [15]. Nurhopipah and Harjoko have tried to perform face recognition from a video stream using a model developed based on Counter-Propagation Network (CPN) [16]. And yet none of the earlier studies are covering the creation of an Indonesia's person face dataset to be referred by this research.

The study aimed to develop a face recognition model based on DCNN and Arcface in Indonesia's pandemic context. To achieve the desired goal, the study needs to develop a face recognition model using Indonesia's person face dataset. The research has contributed novelties in the form of 1) Indonesia's person face dataset named Indonesia Labelled Face in the Wild (ILFW), with a variety of pose, facial expression, fashion accessories, lighting condition/illumination, fashion attributes, and face mask due to recent pandemic situation, and 2) face recognition model that is developed from scratch using the ILFW dataset, DCNN, and Arcface loss function.

The face images for ILFW are collected from the Internet of well-known Indonesian individuals. A list of the names is being prepared and used as a search keyword to obtain respective face images. The researchers performed face extraction, face alignment, integrity checking, image standardization, and face mask augmentation on the collected images to prepare the ILFW dataset. The study also used Apache MXNet [17] as its machine learning library to prepare and develop a face recognition model based on DCNN and Arcface loss function. The developed face recognition model is evaluated based on its accuracy and the performance was compared to the pre-trained and pre-tuned model. The pre-tuned model is a face recognition model that is further trained based on a pre-trained model. The study refers to the CRISP-DM methodology [18] to guide the entire process of the work.

The paper consists of different sections, including the 1) introduction and the background of the research, 2) the review of relevant literature/past study as an academic reference, 3) the adopted methodology to guide the carried-out activities, 4) the discussion about the result, including iterations of training sessions and its recorded model performance, and 5) the result's analysis as the conclusion of the work.

2. **Literature Review.**

The study refers to various literature and earlier studies as its academic reference.

2.1. **Residual Learning Network (ResNet).**

Deep Learning (DL) is essential for the image recognition task. DL is a subfield of Machine Learning (ML), the science field and art of programming that enables a machine to learn from data [19]. ML employed layer by layer learning and adopted Deep Neural Network (DNN) architecture. DNN is a kind

of an Artificial Neural Network (ANN), which comprises a collection of layers and nodes that are connected, resembling a human neuron system [25]. At the minimum, ANN consists of input, hidden, and output layers. DNN is an ANN with more than one hidden layer.

DNN introduces a problem. The previous study reveals that a DNN that consists of more than 20 layers exhibits an accuracy degradation. Therefore, the popular conception that a deeper network will produce better accuracy is inaccurate.

To resolve this issue, another researcher found a solution by introducing a residual learning approach. This approach has successfully demonstrated a positive result by reducing training and testing error on a deep and ultra-deep neural network.

The goal of the residual learning approach is to ease the backpropagation calculation in ultra-deep layers. It creates a shortcut or skipped connection as an identity function that bypasses the backpropagation calculation of a group of layers called a residual block. In this way, the model can achieve the best learning performance utilizing ultra-deep neural network layers and at the same time reduce the error rate by having a more efficient and purer backpropagation through a shorter gradient descent path. This technique has been implemented successfully on the ImageNet dataset with multiple layers deep, including 18, 34, 50, 101, and 152 layers with the best result of 5.71 test error percentage [20].

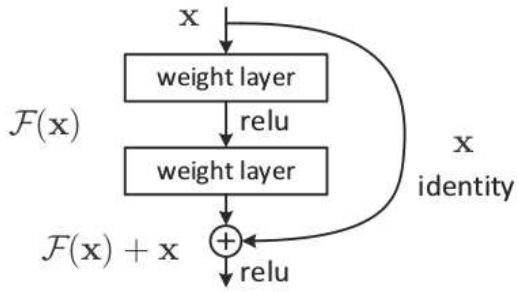

FIGURE 1. Residual learning concept [20]

2.2. **Deep Convolutional Neural Network (DCNN).** DCNN is furtherance of Computer Vision (CV) using DL. CV is a multi-disciplinary science field that focuses on developing the computer's capacity to comprehend images or video [7,21,22]. DCNN is operated based on Convolutional Neural Network (CNN) algorithm. DCNN comprises two different types of layers. They are convolutional and pooling/subsampling. DCNN

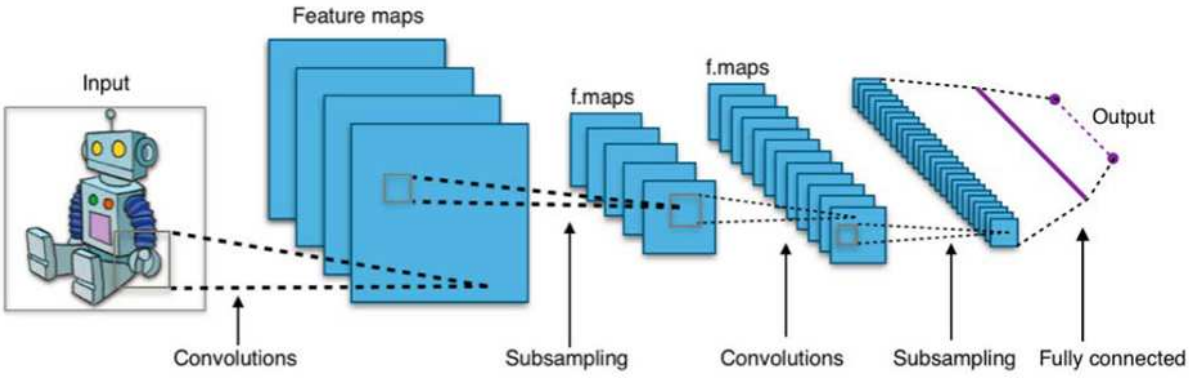

FIGURE 2. DCNN key concepts [23]

consists of a learning process called convolution operation to recognize pixel to pixel relative position as features (line, dots, angle, shape, contour) in an image [23].

DCNN encompasses 3 important concepts. First is the *local receptive fields* as data regions in an image that are subsequently collected. Those data regions will be merged by a later process mathematically. The local receptive fields concept resembles the visual cortex in the human sight neuron system. The human vision system interprets regions of images separately and then merges them to conclude the meaning. Second is the *shared weight*, the parameter for the creation of local receptive fields by kernel or filter.

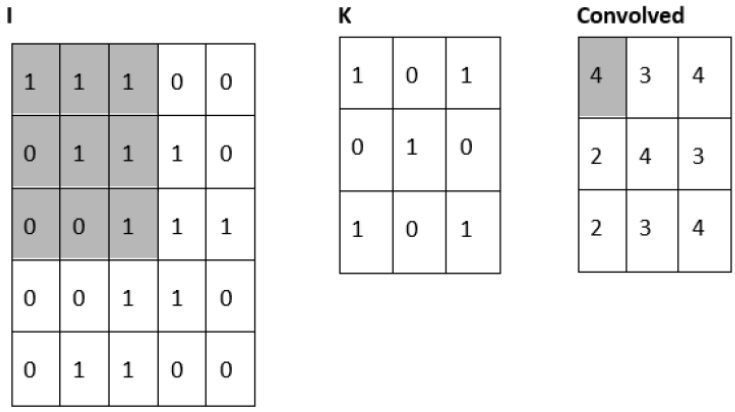

FIGURE 3. Illustration of local receptive field [23]

Third is the *pooling layers* that provide the approach to summarize detected feature within a local receptive field for further merge in the convolutional layer, which concludes the image's features to produce the desired output. In the end, DCNN uses a flattened, fully connected layer, and softmax layer to estimate its classifications.

2.3. **Face recognition.** Face recognition is a technology that can identify a face within an image or video stream [24] based on the pre-collected face database. The face is one of the biometrics, a measurement and calculation of a human individual's specific characteristics [25], that served as a relatively accurate and non-intrusive identification for an individual [26]. The discriminant power of the face is less than other biometric features, such as fingerprint or iris. However, it is getting popular because of the contactless and comfortable nature during use. Essentially, the face recognition process is the identification of visual patterns within an image [24]. It consists of 4 main flows: detecting, aligning, extracting the feature, and matching the face.

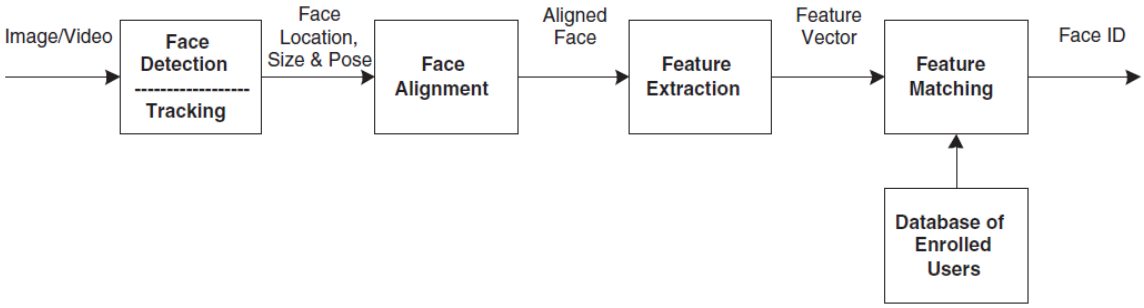

FIGURE 4. The face recognition technical process flow [24]

The face detection process is responsible for locating a face within an image. The face alignment process standardizes the position, scale, size, and pose of an identified face image. The feature extraction process yield maps the main features and landmarks such as eye, nose, mouth, and skeleton's contour of the face. The last process calculates the matching probability between the extracted face's feature with the enrolled data stored in the database.

2.4. **Real-time face recognition.** Real-time face recognition receives a continuous stream of still images called a frame from a video surveillance camera as a source [24]. The number of Frames Per Second (FPS) could vary from one camera to another. The face recognition engine needs to recognize the face(s) within a frame and show it as sequentially contiguous images. Real-time face recognition poses some technical challenges. It must deal with potential illumination variations, primarily for the video camera to operate in an open space. The face recognition engines will require high computation resources to recognize faces from multiple video cameras simultaneously. The video camera could capture the face image in various sizes/scales depending on the relative position. The person's facial image could be detected in different poses depending on its circumstances. The facial expression could influence the result of recognition. Lastly, due to the recent COVID-19 pandemic situation, the individual's face captured via CCTV might be using a face mask.

2.5. **CRISP-DM.** CRISP-DM is an open-source methodology for analytical model development during data mining activities [18,27]. CRISP-DM outlines a circle of processes, representing phases in modeling activities within data mining.

Accordingly, the study is organized into the following steps: 1) business understanding of goals and needs; 2) data understanding and profiling of training, testing, and validation data; 3) data preparation used for model training; 4) modeling face recognition; 5) evaluation of the developed model; and 6) deployment of face recognition model to the end-user/production environment.

2.6. **RetinaFace and Arcface.** RetinaFace is a work by Deng et al. [28] that allows efficient and high-performance face detection in an open environment/in the wild. RetinaFace is a DCNN model that helps the developed system recognize a face in the image rapidly and accurately. Arcface is another work by Deng et al. [10] that has designed a loss function of the ML model to improve the discriminant power of DCNN based face recognition model. Loss-function is a method that evaluates the fitness of the algorithm within the model to its training data. A loss function produces zero value if the prediction perfectly matches the target value and a non-zero number value if it deviates from the target value.

The result of the study by Deng et al. showed that the performance comparison on the LFW and YTF dataset for the Arcface loss function outperformed others with 99.83 accuracies. It was performed against 11 other loss functions, including Softmax, Triplet loss, and SphereFace [10]. This is the main reason why this study used Arcface as its loss function.

3. **Methodology.** The research aimed to produce a face recognition model that can recognize Indonesian people's faces with reasonable accuracy. To achieve this objective, the study needs to use a face dataset of Indonesian with adequate quantity and variations of race/ethnicity, image resolution, illumination, face expression, fashion accessories, pose, scale, color, contour, and angle/position. Due to the current pandemic situation, the need to recognize a face with a medical mask is also part of the requirement. Currently, there

is no ready-to-use Indonesia's person face dataset available. Therefore, it needs to be developed prior to the face recognition model training.

3.1. **ILFW dataset development.** The work of the Indonesia Labelled Face in the Wild (ILFW) face dataset in this study was based on the earlier study of Huang et al. [29] with modifications/enhancements. Similarly, ILFW was designed to be used as a face dataset for the recognition tasks. Both datasets provide images with unique label and 2 types of pair: matched and mismatched, for the model to learn unique identities and whether the pair exhibit the same individual or not, as shown in Figure 5. Both face datasets also provide variations of race/ethnicity, image resolution, illumination, face expression, fashion accessories, pose, scale, color, contour, and angle/position.

FIGURE 5. Matched and unmatched pair in ILFW dataset

The modification of the ILFW dataset compared to LFW was driven by the locality requirements, technology usage, approach, and its purpose. For the person's label, the ILFW dataset used a full name convention name to avoid similar abbreviations. For example, "Christophorus Tri Harsono" will be labeled as "Christophorus_T_Harsono" in LFW, while in ILFW, the label will be "Christophorus_Tri_Harsono". For face detection and alignment, this study used the RetinaFace model [30] to detect face within an image, and OpenCV [31] to align it. In this case, LFW development used a Viola-Jones face detector [32].

ILFW limits its face image collections to Indonesian only. This is related to the purpose of the ILFW dataset which is to support the development of the face recognition model that will be implemented in Indonesia. ILFW also provides a variety of face image (image resolution, illumination, face expression, fashion accessories, pose, scale, color, contour, and angle/position) to represent a single identity (Figure 6). In LFW, 71 percent or 4096 of 5749 people were only represented by one image. In this case, the ILFW provides richer variety for each identity in the dataset.

ILFW contains 27,376 images of 1,200 identities, and most of the identities consist of more than 10 images, or on average 20 photos per identity. The face masks also augmented

FIGURE 6. Variety of face image within a single identity in ILFW dataset

into the face image using 1/3 of the collected photos to accommodate the face recognition in a pandemic situation.

The original Arcface study recommended image size of 112×112 for the face dataset that is used in model training. LFW uses a 250×250 pixel image size for its photo image. The study also performed data cleansing activities to ensure the integrity of collected face images for each identity. For this particular purpose, the study uses RetinaFace MobileNet0.25 [30].

To summarize, the following aspects are the differences between the LFW and the ILFW datasets.

TABLE 1. Modifications of ILFW compared to LFW

Aspect	LFW	ILFW
Image label convention	A unique name based on first name, middle name abbreviation, and last name. For example, "George_W_Bush"	A unique name based on full name. For example, "George_Walker_Bush"
Scope	Individuals from various countries	Indonesian only
Face detection	Viola Jones [32]	RetinaFace [28]
Alignment	N/A	Aligned using OpenCV
Variety within a single identity	Limited variety over 29 percent of its face image	All
Number of identities	5,749	1,200
Number of images	13,233	27,376
Face mask	N/A	Augmented face mask of 1/3 of face images [33]
Image size	250×250 pixels	112×112 pixels
Face comparisons	N/A	RetinaFace MobileNet0.25

The development of ILFW dataset is carried out based on the following workflow.

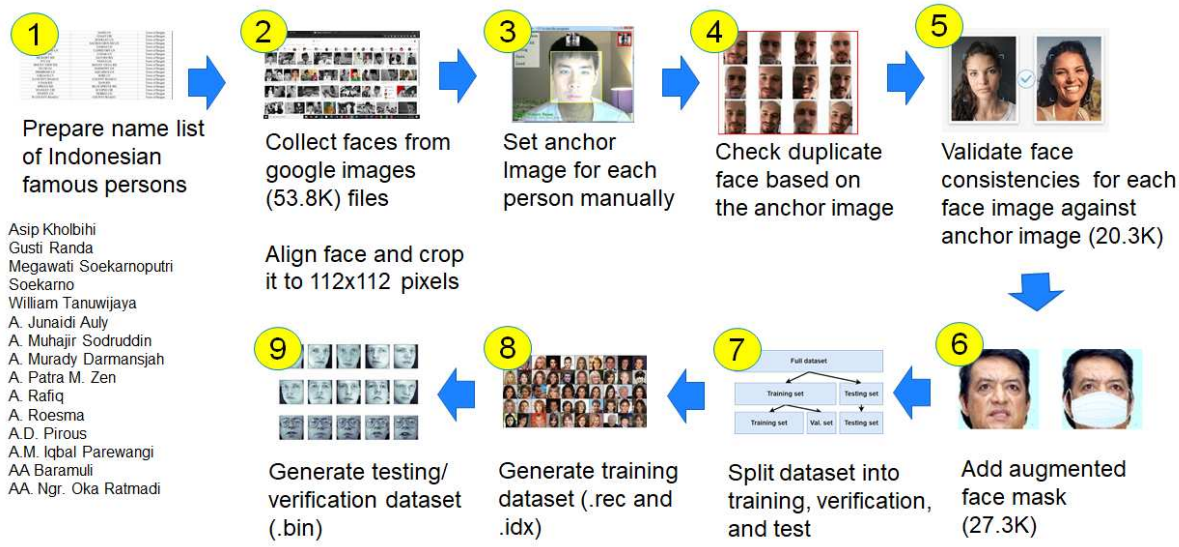

FIGURE 7. Dataset preparation workflow

3.1.1. *Prepare the name list of Indonesian famous persons.* The researchers prepared a list of 5,000 named Indonesian well-known individuals from different professions/backgrounds, such as presidents, politicians, actors/actresses, singers, or social activists. Whenever possible, the full name of the person is used for the list. Each person's name is separated by a new line. To avoid duplicates, the name was sorted alphabetically and was checked for uniqueness using 2 bases: 1) the full name, and 2) the abbreviated middle name. The script will report if a duplicate exists in any line of the list.

3.1.2. *Collect faces from google image website.* Based on the prepared name list, a pre-built python script searched the google image website using the full name of the person as a keyword. The script crawled through the search result downloaded the image programmatically and detected faces from each photo. If any of the images contains multiple faces, the script will only take the first image based on its order of occurrence. Then each face image is aligned, standardized as a 112×112 pixels image.

FIGURE 8. Illustration of face alignment process, before and after

The face image is stored in a specific folder and filenames according to the person's name, replacing space with an underscore. Each image will be indexed based on its download sequence. The script used a four-digit sequence number as the suffix to differentiate multiple images for a single person. For example, the image file for the fourth Indonesian President, Abdurrahman Wahid, will be stored in a folder called "`Abdurrahman_Wahid`", and the first file will be named "`Abdurrahman_Wahid_0001.jpg`". Each person could have more than 10 photo images with various image quality, orientation, ages, and illumination conditions. To avoid duplication of images, during download, the second and next following images will be compared with the earlier downloaded image. The python script will identify duplicate face image files by calculating and comparing the image hash value.

FIGURE 9. An illustration of collected face

3.1.3. *Set anchor image manually by hand.* The researchers pick a single image for each named individual as the anchor image by hand. The anchor image will be used to look for duplicate images between identities and face consistencies at the later step. The anchor image is the first face image for each folder sorted by the filename alphabetically.

3.1.4. *Check for duplicate image.* Using a script, the researchers looked for duplicate faces between each identity based on each anchor image. The reason being is that it is possible that the researchers registered the same person twice or more in the name list using a different alias name. For example, the fourth president of Indonesia, Abdurrahman Wahid, was known also by another nickname, which is "Gus Dur". The other possible scenario, that the researchers picked the wrong anchor image for a person. The unique face

image for each identity is a key factor to the success of model training. During modeling, the researchers once found out that duplicate images caused the model accuracy to stop improving at around 40 percent.

3.1.5. *Validate faces consistency.* During image searching, it is possible that the google image engine yielded someone else faces other than the name provided as a search keyword. The script checks the face consistencies to ensure that the collected face images within the same folder belong to the same person. A written python script used the anchor image as the reference face for each identity and compared it to the rest of the collected images. This script moves the inconsistent images into a folder named "invalid".

3.1.6. *Augmented face mask.* To cover the various visual scenario of faces originated from CCTV, including the possibility of people using a mask during the COVID-19 pandemic, the researchers use the face augmentation technique. The script generates new images to augment the face mask based on the 1/3 of currently collected valid images [33].

FIGURE 10. An illustration of face mask augmentation process [33]

3.1.7. *Split datasets.* For modeling purposes, the researchers split the name list into training, validation, and test parts, with 80 : 10 : 10 proportion. Another python script is developed to shuffle the name list and write each portion of the name list for each part into separate files named "`train.part`", "`ilfw.part`", and "`ilfw-test.part`". The process will ensure that each name exclusively belongs to one part only.

3.1.8. *Generate training datasets.* The next script reads the "`train.part`" file, and generates another file called "`train.lst`". The file is written based on the following tab-delimited text file structure.

TABLE 2. The ".lst" file structure

Image alignment flag	File name path	Unique label for the identity
0: to indicate that the face image is not aligned 1: to indicate the face image is aligned	Full path name of image file	In this study we are using a running number from "0" to represent the label for each identity. After the first person, the next person will use label "1"

The sample of ".lst" file content is as follows.

```
1 <tab> e:/dataset/faces/AM_Fatwa/AM_Fatwa_0001.jpg <tab> 1
```

Based on the content of "`train.lst`" file, the script will generate another file in predefined Apache MXNET Gluon Face Dataset format [34] training dataset, that consists of 2 files, which are "`train.rec`" and "`train.idx`". The script will read the image file based on the "`train.lst`" and write all the images as contiguous content into "`train.rec`" file. The "`train.idx`" file will contain the list of pointers to the image content in the

"`train.rec`" file. This format is part of the strategy to speed up the model training process in the Apache MXNET framework. The script also wrote one additional file called "property". The file contains: 1) the number of identities within the training dataset and 2) the size of face images in pixels. All value is separated by comma. The sample of the "`property`" file is as follows:

```
320, 112, 112
```

3.1.9. *Generate validation and testing datasets.* For model validation and testing purposes, another script is prepared to generate a matched and unmatched pair list for each part. The script reads the "`ilfw.part`", and "`ilfw-test.part`" file, and writes a matched pair section into a file called "`ilfw-pairs.txt`" for validation and "`ilfw-test-pairs.txt`" for testing, in the following tab-delimited text file structure.

TABLE 3. The matched pair line structure

Identity folder name	The first image index	The second image index
The folder name assigned to a particular identity	The first image sequence index	The second image sequence index

The sample of matched pair content is as follows:

```
Dennis_van_Leeuwen <tab> 1 <tab> 14
```

The above matched pair record will be used in testing process to match image file `Dennis_van_Leeuwen_0001.jpg` with `Dennis_van_Leeuwen_0014.jpg`.

Following the matched pair section, the script wrote another *unmatched pair* section into the same file in the following tab-delimited text file structure.

TABLE 4. The unmatched pair line structure

The first identity folder name	The first image index	The second identity folder name	The second image index
The folder name of a first identity	The first identity image index	The folder name of a second identity	The second identity image index

The sample of unmatched pair content is as follows:

```
Desi_Arryani <tab> 13 <tab> Ahmad_Muzani <tab> 26
```

The above unmatched pair record will be used in the testing process to ensure that the model can predict that the image file `Desi_Arryani_0013.jpg` is a different person compared to `Ahmad_Muzan_0026.jpg`.

Based on the "`ilfw-pairs.txt`" and "`ilfw-test-pairs.txt`", the script will generate a final validation file "`ilfw.bin`" and "`ilfw-test.bin`" in object serialization format using *python pickle* library. This file will contain all the image file content that is organized based on matched and unmatched pairs that are developed earlier.

3.2. **Develop face recognition model.** During the modeling phase, the researchers build the model by performing training using the prepared dataset from the previous activities. The research uses The Apache MXNET Machine Learning framework to build the model.

3.2.1. *Model architecture.* The study produces a model that adopts DCNN network architecture called ResNet100, with the following specification.

TABLE 5. Model architecture based on Arcface study [10]

No.	Stage	Specification					Convolutional layer	
		Type	Kernel size	Strides	Filter	Repeated	Count	Position
1	–	Convolutional	3×3	1×1	64	1	1	1
2	1	Convolutional	3×3	2×2	64	1	1	2
		Batch normalization						
		Batch normalization				3	6 (2×3)	3-8
		Convolutional	3×3	1×1	64			
		Batch normalization						
		Relu activation						
		Convolutional	3×3	1×1	64			
		Batch normalization						
3	2	Batch normalization				13	26 (2×13)	9-34
		Convolutional	3×3	1×1	128			
		Batch normalization						
		Relu activation						
		Convolutional	3×3	1×1	128			
		Batch normalization						
4	3	Batch normalization				30	60 (2×30)	35-94
		Convolutional	3×3	1×1	256			
		Batch normalization						
		Relu activation						
		Convolutional	3×3	1×1	256			
		Batch normalization						
5	4	Batch normalization				3	6 (2×3)	95-100
		Convolutional	3×3	1×1	512			
		Batch normalization						
		Relu activation						
		Convolutional	3×3	1×1	512			
		Batch normalization						
6	–	Average pool, fully connected, softmax					1	

The study used ResNet100 network architecture as it exhibits the best performance among the other network architecture explained in the Arcface study [10]. The model also utilizes Arcface as its loss function.

3.2.2. *Model training approaches.* The previous research on Arcface provides a pre-trained model. The study decided to use 2 approaches for model training: 1) train the model from scratch and 2) fine-tune the model from a pre-trained model. The main difference between the two approaches is whether a pre-existing model from the model zoo is used and loaded in the beginning as a checkpoint by the training process. In a fine-tuned model approach, the training process loads the baseline model first and uses the prepared training dataset. In this way, the training session will adjust the model parameter, weight, and bias based on the new training dataset. The second approach offers a faster accurate result. The researchers also fine-tuned the model's hyperparameter to get better model performance.

3.3. **Incremental and iterative process.** The researchers developed the ILFW dataset and performed model training in iterative processes. For the model training, the study

used Apache MXNET as the machine learning framework [35] and conducted multiple model training sessions along with respective modifications of the number of identities in the dataset and model parameters to achieve the desired accuracy and performance on each iteration. To start with, 400 identities are initially added to the ILFW dataset and based on it, the model training session is started. Each iteration consists of a model training from scratch and a fine-tuned approach, plus a performance verification for the Arcface pre-trained model. The training model is stopped until the model performance is no longer improving. Due to time and resources limitation, the training iterations finally stopped after 5 iterations, with 1,200 identities in the ILFW dataset.

3.4. **Hyperparameter setting.** The study altered the hyperparameters to fine-tune the model training process to achieve the desired model's accuracy and training time. The learning rate hyperparameter controls how the model training process resolves the error value. The backpropagation process during model training uses the learning rate parameter value to adjust hidden layers' weight and bias based on the calculated training error. The research uses 0.01 as its learning rate parameter value to yield the best model's training performance and accuracy. The model training process could use a smaller learning rate value since it will potentially smoothen the adjustment of weight and bias but will take a longer time to train.

The study set the batch size hyperparameter value to 64 records in considering the memory limitation. Whenever possible, setting this parameter to a larger value is preferable since it will improve the overall model accuracy and training time. By doing so, the learning rate parameter value also could be decreased.

4. **Results and Discussions.** Due to its incremental and iterative nature of the development processes, the ILFW dataset's size and the model performance vary accordingly.

4.1. **ILFW dataset instances.** Based on the number of iterations, there are 5 instances of ILFW datasets that are generated. Each instance is an accumulated version of the earlier and consists of 5 files: 1) "`ilfw.bin`", 2) "`ilfw-test.bin`", 3) "`property`", 4) "`train.idx`", and 5) "`train.rec`". The first instance contains 400 identities, and the next has 200 identities more than its predecessor. Every identity and image file are unique across the dataset. Each instance is used for model training and the respective performance is measured. The comparisons of generated ILFW dataset instance can be summarized as an exhibit in Table 6. Each ILFW dataset instance averagely contains around 23 image files per identity.

Table 6. Instances of the generated ILFW datasets

Comparisons	Iterations				
	1	2	3	4	5
Number of identities	400	600	800	1,000	1,200
Size (in Bytes)	74,457,088	11,512,504	153,819,452	190,701,855	226,909,893
Number of image files	9,193	13,942	18,455	22,959	27,376
Average number of files per identity (rounded)	23	23	23	23	23

4.2. **ILFW training and model performance.** The model training is performed in 5 iterations, each using its corresponded ILFW dataset instance. At the end of each model training iteration, the developed script used the Apache MXNET library to calculate the evaluation metric/indicator to provide information about the model performance. For the face recognition task, the relevant evaluation metric is accuracy. The reason is that one of the use cases for the face recognition model is to record employee attendance automatically. The false-positive identification of a person is not desirable because it potentially causes a false recording of employee attendance. The face recognition engine will need to use an optimized matching threshold to avoid such cases. The model's training performance is recorded based on the accuracy for two types of models, which are 1) the ILFW model that is developed from scratch and 2) the fine-tuned Arcface model. The following diagram exhibited the model training performance results.

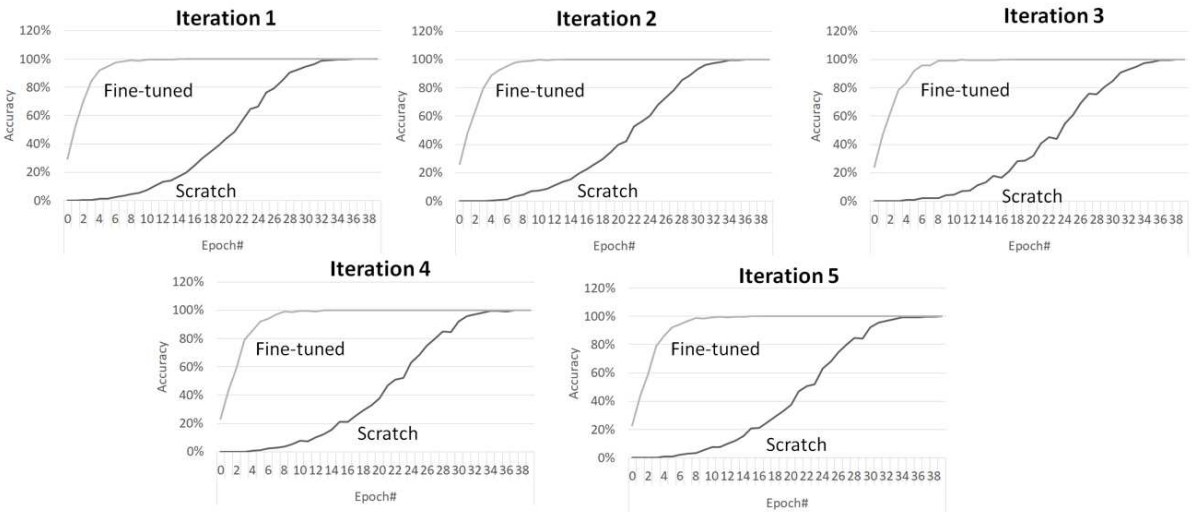

FIGURE 11. Model training performance results

The model training is conducted using Arcface loss function, ResNet100 Network Architecture, and the following hyperparameters: 1) learning rate of 0.01 and 2) batch size: 64. The Arcface fine-tuned model achieved faster and better training accuracy compared to the ILFW model that is built from scratch.

The final model's testing performance will include the evaluation of the pre-trained Arcface model performing using the ILFW dataset. The following table shows the final model testing performance measurement results.

TABLE 7. Model testing performance comparisons

Model	The ILFW dataset iterations									
	1		2		3		4		5	
	400 identities		600 identities		800 identities		1,000 identities		1,200 identities	
	Acc.	Ep.#	Acc.	Ep.#	Acc.	Ep.#	Acc.	Ep.#	Acc.	Ep.#
Pre-trained	97.87%	–	98.64%↑	–	98.08%↓	–	98.08%–	–	97.71%↓	–
Fine-tuned	96.75%	34	96.84%↑	22	97.45%↑	26	96.70%↓	25	97.60%↑	60
Scratch	86.32%	34	87.97%↑	11	88.70%↑	44	88.16%↓	43	92.00%↑	41

The model's testing accuracy showed an increasing trend for most of the iterations and both types of models. The result indicated that 1) the provided ILFW dataset and the model architecture could effectively support the achievement of the reasonable model's performance and 2) as more identities being added into the ILFW dataset instance, it correlates with a better model performance.

The training process uses a single Tesla T4 Graphical Processing Unit (GPU) environment with 8GB RAM. The training session stopped for ever iteration is stopped when accuracy was no longer improving. The epoch ("Ep.#") column of Table 7 indicates the iteration when the training session records the achieved accuracy rate for the first time. The study finally stopped the training process due to time and hardware resource limitations. The researchers believe that future follow-up research could increase the ILFW model's accuracy by adding more identities into the ILFW dataset.

4.3. **Model inference test.** The researcher performed a simple model inference test, to test the result of face recognition. The results of the test represent a cosine similarity score between 2 faces. The comparison test is performed between a frontal face with a face using two types of medical mask, no glasses, side-way position, up, down, with a different person, and lighting condition. The results are as follows.

TABLE 8. Model inference test results comparisons

Face comparisons			Pre-trained model	Fine-tuned	Scratch
		Frontal to frontal	0.9999996	1.0000001	**0.9999999** (Best)
		Frontal to masked	0.3876624	**0.46413758** (Best)	0.12811269
		Frontal to masked-green	0.5920889	**0.59673184** (Best)	0.37431887
		Masked to masked-green	**0.50916225** (Best)	0.48077944	0.1020531
		Frontal to no glasses	0.4829249	**0.5647622** (Best)	0.4561047
		Frontal to side	0.16450742	**0.41668436** (Best)	0.22649498
		Frontal to up	**0.36997595** (Best)	0.25528038	0.08434215
		Frontal to down	**0.39259762** (Best)	0.14442188	0.029435651
		Frontal to different	0.20063815	**0.11830093** (Best)	0.19301784
		Frontal to lighting	0.49470302	**0.7065262** (Best)	0.6838537

The fine-tuned model exhibits the best performance, because it combines the pre-trained model and new training with the ILFW dataset. The face recognition model that is built from scratch in some of the test performed better than the *pre-trained* model which indicates that the training with ILFW dataset improves the model capability to recognize faces in unconstrained environment.

4.4. **Develop model from scratch.** The development of a face recognition model from scratch is necessary for future business implementation due to the restrictive use of the pre-trained Arcface model only for non-commercial use. Therefore, although the pre-trained Arcface model showed the best performance, it cannot be used in production for commercial use.

4.5. **Business use case.** The use case of face recognition technology varies, including for 1) security, by identifying and locating a person; 2) health, detecting DiGeorge Syndrome by recognizing facial features; 3) marketing, for supporting the know-your-customer (KYC) process, 4) finance, implementing selfie payment [36]; or 5) retail, tourism, defenses, and cybersecurity [37]. A business organization can use the resulted model as a real-time face recognition engine for using Close-Circuit Television (CCTV) for 1) detecting employee's real-time location in the office building, primarily when the seat arrangement is non-permanent [9]; 2) detecting unknown individuals in a restrictive area or intra-organization private space; and 3) implementing automatic employee attendance recording [13]. The model can recognize the individual with or without a face mask as an essential capability given the current pandemic situation.

4.6. **Deployment considerations.** For implementation, the study recommended harvesting the employee face dataset using the actual CCTV images as additional training data for the engine to ensure its high accuracy. The face dataset should be collected and prepared with careful consideration of privacy protection. A quick note on the automatic attendance use case, during auto check-in and check-out, the system needs to validate the employee's location using the GSM locator through a mobile app to avoid unwanted mistakes. Different threshold for face recognition with face mask is also required since mask usage will reduce the similarity score between faces in the database.

5. **Implications, Limitations, and Conclusions.** This research has developed a novel Indonesian people face dataset with 1,200 identities named Indonesia Labelled Face in the Wild (ILFW). It consists of a facial image of well-known Indonesian figures, with the various pose, face expressions, age, illumination condition, fashion attributes, and including face mask due to the recent COVID-19 pandemic. The research also developed a face recognition model based on the ILFW dataset. The developed model can recognize the face of Indonesian individuals with/without a face mask in various poses, face expression, fashion accessories, illumination, fashion, and other outdoor-related factors. The model adopted deep convolutional neural network architecture called ResNet100, with Arcface loss function to achieve 97.17 percent model testing accuracy using the fine-tuned approach, and 92 percent accuracy for the model that is built from scratch. The study also has demonstrated a method to build a custom face dataset and a face recognition model from scratch with reasonable accuracy.

The future study should be focused on adding more identities into the ILFW dataset to achieve better accuracy. Organizations and companies can utilize the model for various real-time face recognition use cases, including automatic attendance, security surveillance, and individual locator within the building. Since the model can recognize a face with a

mask, the face recognition engine could help organizations/companies to implement a touchless and risk-free attendance system given the current pandemic situation.

Acknowledgment. Authors thank the NVIDIA & BINUS AI Research and Development Center (AIRDC) for providing the Graphics Processing Unit (GPU) environment for model training.

REFERENCES

[1] M. Coccia, Why do nations produce science advances and new technology?, *Technol. Soc.*, vol.59, DOI: 10.1016/j.techsoc.2019.03.007, 2019.
[2] D. H. Jonassen, J. Howland, J. Moore and R. M. Marra, *Learning to Solve Problems with Technology: A Constructivist Approach*, Merrill Prentice Hall, the United States of America, 2003.
[3] J. L. Bower and C. M. Christensen, Disruptive technologies: Catching the wave, *Harvard Business Review*, 1995.
[4] C. M. Christensen, M. E. Raynor and R. McDonald, What is disruptive innovation, *Harvard Business Review*, vol.93, no.12, pp.44-53, 2015.
[5] D. M. West and J. R. Allen, How artificial intelligence is transforming the world, *Report. April*, vol.24, 2018.
[6] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, 2010.
[7] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1982.
[8] M. Bramer, *Artificial Intelligence in Theory and Practice*, Springer-Verlag New York Inc., 2006.
[9] E. Briant and K. Horio, Face tracking and detection of children faces in omnidirectional video, *ICIC Express Letters, Part B: Applications*, vol.11, no.1, pp.85-91, 2020.
[10] J. Deng, J. Guo, N. Xue and S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.4690-4699, 2019.
[11] C. Lu and X. Tang, Surpassing human-level face verification performance on LFW with Gaussian-Face, *arXiv.org*, arXiv: 1404.3840, 2015.
[12] H. Patel and R. Connolly, Factors influencing technology adoption: A review, *The 8th International Business Information Management Conference*, Dublin, Ireland, 2018.
[13] G. R. Naufal, R. Kumala, R. Martin, I. T. A. Amani and W. Budiharto, Deep learning-based face recognition system for attendance system, *ICIC Express Letters, Part B: Applications*, vol.12, no.2, pp.193-199, 2021.
[14] F. M. Andiani and B. Soewito, Face recognition for work attendance using multitask convolutional neural network (MTCNN) and pre-trained FaceNet, *ICIC Express Letters*, vol.15, no.1, pp.57-65, 2021.
[15] A. Chowanda and R. Sutoyo, Convolutional neural network for face recognition in mobile phones, *ICIC Express Letters*, vol.13, no.7, pp.569-574, 2019.
[16] A. Nurhopipah and A. Harjoko, Motion detection and face recognition for CCTV surveillance system, *Indonesian J. Comput. Cybern. Syst. (IJCCS)*, vol.12, no.2, pp.107-118, 2018.
[17] T. Chen et al., MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems, *Computer Science*, 2015.
[18] C. Schröer, F. Kruse and J. M. Gómez, A systematic literature review on applying CRISP-DM process model, *Procedia Comput. Sci.*, vol.181, pp.526-534, 2021.
[19] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2017.
[20] K. He, X. Zhang, S. Ren and J. Sun, Deep residual learning for image recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.770-778, DOI: 10.1109/CVPR.2016.90, 2016.
[21] T. Huang, *Computer Vision: Evolution and Promise*, 1996.
[22] M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis, and Machine Vision*, Cengage Learning, 2014.
[23] A. Gulli and S. Pal, *Deep Learning with Keras*, Packt Publishing Ltd., 2017.
[24] A. K. Jain and S. Z. Li, *Handbook of Face Recognition*, Springer, New York, 2011.
[25] National Science and Technology Council (NSTC), Glossary of Biometric Terms, *Fulcrum Biometrics*, https://www.fulcrumbiometrics.com/Articles.asp?ID=268, Accessed on Oct. 17, 2020.

[26] S. H. Lin, An introduction to face recognition technology, *Informing Sci.*, DOI: 10.28945/569, 2000.
[27] J. Han, J. Pei and M. Kamber, *Data Mining: Concepts and Techniques*, Elsevier, 2011.
[28] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia and S. Zafeiriou, RetinaFace: Single-stage dense face localisation in the wild, *arXiv.org*, arXiv: 1905.00641, 2019.
[29] G. B. Huang, M. Mattar, T. Berg and E. Learned-Miller, *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*, http://vis-www.cs.umass.edu/lfw/lfw.pdf, 2008.
[30] J. Deng, J. Guo, E. Ververas, I. Kotsia and S. Zafeiriou, RetinaFace: Single-shot multi-level face localisation in the wild, *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.5202-5211, DOI: 10.1109/CVPR42600.2020.00525, 2020.
[31] Opencv.org, *OpenCV: Open Source Computer Vision Library*, https://github.com/opencv/opencv, 2021.
[32] M. Jones and P. Viola, Fast multi-view face detection, *Mitsubishi Electr. Res. Lab TR-20003-96*, vol.3, no.14, 2003.
[33] P. Bhandary, Mask classifier, *GitHub Repository*, https://github.com/prajnasb/observa tions, 2020.
[34] Apache MXNet, *GluonFace Documentation Release 1.0.1*, https://gluon-face.readthedocs.io/_/downloads/en/latest/pdf/, 2019.
[35] Apache MXNet, A flexible and efficient library for deep learning, *Apache Softw. Found.*, 2018.
[36] Thales, *Facial Recognition: Top 7 Trends (Tech, Vendors, Markets, Use Cases & Latest News)*, https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition, Accessed on Sep. 04, 2020.
[37] Giscle, *Facial Recognition Use-Cases in All the Industries*, https://medium.com/giscle/facial-recognition-use-cases-in-all-the-industries-8c960aaf91fd, Accessed on Oct. 15, 2020.