JOIV : Int. J. Inform. Visualization, 7(3) - September 2023 959-966

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Max Feature Map CNN with Support Vector Guided SoftMax for Face Recognition

Herdianti Darwis [a], Zahrizhal Ali [a,*], Yulita Salim [a], Poetri Lestari Lokapitasari Belluano [a]

[a] *Department of Computer Science, Universitas Muslim Indonesia, Jl. Urip Sumoharjo, Makassar, Indonesia*

Corresponding author: [*]zahrizhalali.labfik @umi.ac.id

***Abstract*— Face recognition has made significant progress because of advances in deep convolutional neural networks (CNNs) in addressing face verification in large amounts of data variation. When image data comes from different sources and devices, the identifiability of other classes and the presence of profile face data can lead to inaccurate and ambiguous classification because other classes lack discriminatory power. Furthermore, using a complex architecture with many deep convolutional layers can become very slow in the training process due to a huge amount of Random Access Memory (RAM) usage during the reverse pass of backpropagation. In this paper, we design a light CNN architecture that addresses these challenges. Specifically, we implemented Max-feature-map (MFM) into each convolutional layer to improve the accuracy and efficiency of the CNN. The strength of the support vector-guided SoftMax (SV-SoftMax) is also used in the proposed method to emphasize misclassified points and adaptively guide feature learning. Experimental results show that the 9-Layers CNN with MFM layer and SV-SoftMax outperform *VGG-19* with 96.22% validation accuracy and the second rank below FaceNet tested on the same dataset with fewer parameters. Moreover, the model performed well on data that is obtained from various capture devices such as *webcam*, *CCTVs*, *phone cameras*, and *DSLR* cameras. The implications of this research could extend to scenarios requiring face recognition technology implementation with light size, such as surveillance and authentication systems.**

***Keywords*— Convolutional neural network; face recognition; SoftMax; deep learning.**

Manuscript received 15 Apr. 2023; revised 26 Jun. 2023; accepted 31 Jul. 2023. Date of publication 10 Sep. 2023.

I. INTRODUCTION

Over the past ten years, convolutional neural networks (CNNs) have grown in popularity as one of the methods for resolving computer vision issues. Strong and recognizable representations learned by CNNs have helped many visual tasks, including image classification, object detection, and face recognition [1]. Face recognition is a fundamental and important practical task in computer vision and pattern recognition. There are two types of face recognition tasks: face identification, which links a particular person, and face verification, which assesses if two face photographs belong to the same identity [2]. Deep convolutional neural network technology has recently been used to address the difficulty of learning discriminative features in face recognition [3] and to enhance performance on a huge data set with a lot of noisy labels and efficient computational cost using Max-Feature-Map (MFM) technique [4]. MFM is an extension of Maxout activation and can be treated as a special activation to separate informative signals from noisy signals [5]. The proposed network of MFM claimed to perform well in utilizing large-scale noisy data and prevent biased results by performing semantic bootstrapping.

CNNs is a special network with a classification loss function within the layers. The currently dominant classification loss function is the SoftMax loss. However, as pointed out by recent studies, face recognition relies on face features discriminative, *i.e.,* They have well-maximized intra-class compactness and inter-class separability, and SoftMax loss typically is not strong enough to handle this particular task [6]–[9]. The classification loss function has been redesigned by many researchers in order to create deep face recognition models. Recently, novel metric learning loss functions such as contrastive or triplet loss were developed. However, both methods are typically computationally expensive [3], [10], [11].

Hard-mining SoftMax (HM-SoftMax) was created in a recent study to enhance feature recognition through the creation of mini-batches from high-loss samples [12]. A similar contrasting study was also conducted to design a soft-

mining SoftMax called Focal Loss (F-SoftMax) [13]. They claimed that by concentrating training on a few difficult examples, many simple negatives are prevented from overloading the model while training. The results are more promising than simple hard-mining SoftMax [14]. The other researchers, Deng et al. [6] and Wang et al. [8], prefer to design margin-based loss functions that directly increase the functional margins between different classes. Later, Angular distance SoftMax (A-SoftMax) was constructed between the ground truth class and other classes to increase the variance across classes [15]. Angularly discriminative features can be learned by CNNs using A-SoftMax and represent a nice geometric interpretation by limiting learned characteristics to discriminant on a super spherical manifold, which is inherently consistent with the a priori assumption that the faces also lie on a non-linear manifold. However, it is usually unstable, and it is difficult to determine the optimal parameters.

Certain techniques [10] and [16] combine SoftMax losses based on Euclidean margins to create joint supervision. However, the characteristics learned by the SoftMax loss have an inherent angular distribution. Furthermore, Euclidean amplitude is said to be inconsistent with SoftMax loss [16], [17]. Most margin losses extend the feature margin to discover distinguishing features from the point of view of the underlying ground truth class. However, they often ignore the discriminant power of other non-ground truth classes. The definition of specific hard examples for mining-based losses is unclear and frequently determined empirically, so concrete examples remain an open problem. In addition, it is still unclear how mining and margin losses are related. A novel loss function designed by Liu et al. [15], called Support Vector guided SoftMax (SV-SoftMax), claimed that it can remove ambiguity from difficult examples and absorb the discriminant ability of other classes by concentrating on the support vectors and semantically merge mining-based and margin-based losses into a single base framework. They tested the method on the LFW [18] and MegaFace benchmarks [2], proving effective.

This paper studies the CNNs with Max-Feature-Map operation combined with SV-SoftMax losses. The carefully designed architecture consists of a 9-layer CNN with MFM embedded and customized Support Vector Guided SoftMax. The proposed architecture is designed to obtain a fewer parameters model with a low computational cost capable of absorbing the discriminative strength of other classes using support vectors. The MFM technique is proposed to test the assumption against the general Rectified Linear Unit (ReLU) activation, which can suppress low-activation neurons in each layer in order to distinguish between relevant and noise signals but also to make the model perform well in terms of speed or computational cost. However, the noisy signals in our dataset case are not the main issue since the dataset is also collected carefully using Webcam, DSLR, Phone Camera, and CCTV and labeled properly corresponding to their classes. Finally, SV-SoftMax losses are proposed instead of the general SoftMax to adaptively emphasize the misclassified points and lead to better discriminative feature learning.

CNNs are recognized as a robust feature extractor and are still one of the most active research in modern face recognition [11]. In the past, deep network-based methods for face recognition used an intermediate bottleneck layer to generalize the recognition learned during training and a classification layer trained on a collection of known face identities [19]. Recently, Schroff et al. [20] introduced FaceNet, which proposes tripled loss and achieves 95.12% accuracy on the YouTube Faces dataset. Similar work by Simonyan and Zisserman [21] trained VGG Network on the LFW dataset [18] and achieved 98.95% accuracy. They improved the model by using a tripled-based metric approach that is similar to FaceNet. Other researchers also studied the triplet loss's effectiveness combined with k-NN and SVM classifier for face recognition and achieved 96% and 95% accuracy, respectively [22].

Then, DeepID2 is introduced, which aims to utilize face identification to boost inter-personal variations and decrease intra-personal variations derived from the same identity using verification signals [10]. DeepID2 achieved 99.15% accuracy on the large LFW dataset and reduced the error rate by 67% against a similar method tested on LFW. Large datasets collected from the internet often contain massive noisy labels that can decrease the accuracy performance [23]. Therefore, Wu et al. [4] proposed a Max-Feature-Map network and used a semantic bootstrapping method to make the network predictions better match noisy labels. Other methods, however, concern the fraction of CNN architecture that can cost the training performance and lead to slow training time. Therefore, Zagoruyko and Komodakis [24] introduced a simple 16-layer CNN-wide residual network similar to ResNet blocks [25] and managed to attain state-of-the-art results on CIFAR-10 dataset [26]. Like MFM, 9-layer CNN with MFM Convolution is trained without adjusting and has low computing expenses.

A neural network that predicts a multinomial probability distribution uses the SoftMax function as the activation function in its output layer. Meanwhile, SoftMax loss computes the multinomial logistic loss of the SoftMax input and is generally identical to the SoftMax layer with a more numerically stable gradient. Some methods implemented customized SoftMax loss and managed to achieve promising results. For example, Wang et al. [7] designed additive margin (AM-SoftMax) loss to keep the optimization stable and incorporate with the classification model's margin. This approach was developed because face verification tasks can be seen as metric learning issues. Then, Deng et al. [6] introduced Additive Angular Margin Loss (ArcFace) to improve the ability to distinguish discriminative power when dealing with noisy labels by proposing sub-center ArcFace.

Sub-center ArcFace boosted the model performance by purifying raw web faces under massive noise. Another method focuses on mining-based SoftMax to improve the discriminative features by focusing on the informative examples. For example, Focal loss (F-SoftMax) [13] and Hard-mining strategy SoftMax (HM-SoftMax) [12]. Several SoftMax loss methods proposed margin-based SoftMax, such as A-SoftMax [16] and Ensemble soft-Margin SoftMax (EM-SoftMax) [7]. However, because they only achieve the feature margin from the perspective of the ground truth class y, these methods are unaware of the significance of non-ground truth classes. Utilizing SV-SoftMax is driven by the need to remove the ambiguity of hard-mining based methods and their

capacity to absorb the discriminative force of additional ground truth, resulting in more discriminative features.

II. Materials and Method

This part examines face recognition tasks using 9-layer CNN with MFM models based on an image dataset collected using a webcam, DSLR, CCTV, and phone camera. First, we introduce the data preprocessing method and datasets. Then, the model's architecture method and the support vector-guided SoftMax are presented.

A. Datasets and Preprocessing Method

The datasets are collected using four devices: webcam, DSLR camera, CCTV, and phone camera. We grabbed the face ROI using *haarcascade* face detection model throughout the entire dataset. On top of that, Open-Source Computer Vision (Open CV) library and Python are employed for the rest of the preprocessing technique. In total, there are 23 classes with frontal and variant poses.

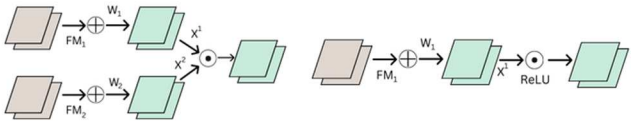
Fig. 1 Dataset sample. Face dataset sample collected and preprocessed with haarcascade face detection. The ROI for face is cropped and saved for the model training phase.

Datasets are collected by image capture and video streams. The *haarcascade* can detect real-time video streams [27]. Thus, it is easy to extract many frames from the videos, contributing to additional images in our datasets. We gathered 8,020 images belonging to 23 classes and then split them into 80/20 train and test split. Since the datasets collected are not balanced, we do not depend on accuracy as the main evaluation metric.

Studies by Wu et al. [4] used grayscale color face images rather than Red Green Blue (RGB) for both training and testing. Furthermore, the images are aligned to 144x144 pixels. This is also one of the main reasons why the model they constructed has fewer parameters than the one we designed with similar architecture. On the other hand, [28] observed that image classification performance is better for higher image resolution since more information can be captured within the network. We set the face images to 225x225 pixels, and all the pixels are normalized to range between 0 to 255 by dividing each pixel by 255. We employed tensor data batch Image Data Generator, which can perform image transformation with real-time augmentation. Certain deep learning studies used image data augmentation approaches [28]. However, in this case, image augmentation is not implemented.

B. Light CNN

In a neural network, activation function plays a crucial role in determining the feature importance and detecting important patterns throughout the feature vector of the given input image. Rectified Linear Unit (ReLU) [29] activation distinguishes informative features by comparing a maximum value to a 0 value. However, this feature may cause the loss of some crucial information, especially in the initial few convolutional layers. Features that fit inside the hidden layers also depend on weight and biases. There is no exact number of hidden layers in a convolutional neural network, and adding more layers increases the number of weights in the network and improves model complexity. While it is possible to reduce the possibility of overfitting and improve model performance, the trade-off is that the model takes more time to train, especially for image classification. Considering the problem of model complexity, we propose a Max-Feature-Map with a simple 9-layer CNN. CNN models based on MFM are light and robust because, in contrast to Maxout activation, MFM works to suppress the activations of a select few neurons [4]. Similar to Maxout, MFM utilizes the max function for neuron activation.



(a) MFM: $h(x)=max(x^1, x^2)$ (b) ReLU : $h(x)=max(0,x^1)$

Fig. 2 Example Comparison of several types of activation: (a) MFM suppresses a neuron due to a competing connection. (b) ReLU suppresses a neuron by thresholding magnitude responses with element-wise max(0,*x*).

By combining two feature maps, the MFM operation generates the element-wise maximum. Using element-wise maximum operation across all feature channels, MFM can extract 50% of informative neurons from the input feature map [4]. In contrast to MFM, ReLU only returns element-wise *max* (0*, x)* where *x* is the given feature from features, weights, and biases calculation. During backpropagation, MFM chooses the best feature at each layer learned by various nodes when adjusting the weights of the connections between neurons. Choosing the best features to use in subsequent layers can greatly improve the model's performance.

MFM suppresses neurons from binary gradients (1 and 0). It has a similar role to ordinal scales and is frequently utilized in biometrics [30]. MFM allows for a condensed display on CNN. On the one hand, the sparse gradients of MFM allow Stochastic Gradient Descent (SGD) to influence responding neurons only during the backpropagation of the training phase. By getting more competitive filters from previous convolutional layers up to two feature maps, MFM can perform feature selection and facilitate creation of sparse links [4].

Here, we will discuss the architecture for our Light CNN with MFM. To do feature selection between convolution layers similar to VGG and Network in Network [31], performing reduction in the convolution filters size by increasing the depth is the motivation for our network to acquire less number of parameters and balance between model complexity and accuracy, thus speeding up the learning process. The implemented constructed 9-layers CNN consists of 5 Max-Feature-Map and convolution layers, 4 NIN layers, and 4 max pooling layers from Wu et al. [4]. This helps produce more robust representations of images, keep the number of parameters low, and improve the quality of feature maps.

TABLE I
LIGHT 9-LAYERS MFM CNN WITH SV-SOFTMAX

Type	Filter Size/Stride, Padding	Output Shape	Number of params
Conv2d	5x5/1, 2	225x225x128	9,7K
MFM		225x225x48	-
Pool	2x2/2	112x112x64	-
Conv2d	1x1/1	112x112x64	4,1K
MFM	-	112x112x32	-
Conv2d	3x3/1,1	112x112x64	18,4K
MFM	-	112x112x32	-
Pool	2x2/2	56x56x32	-
Conv2d	1x1/1	56x56x32	9,2K
MFM	-	56x56x16	-
Conv2d	3x3/1,1	56x56x32	4,6K
MFM	-	56x56x16	-
Pool	2x2/2	28x28x16	-
Conv2d	1x1/1	28x28x16	272
MFM	-	28x28x8	-
Conv2d	3x3/1,1	28x28x16	1,1K
MFM	-	28x28x8	-
Conv2d	1x1/1	28x28x16	144
MFM	-	28x28x8	-
Conv2d	3x3/1,1	28x28x16	1,1K
MFM	-	28x28x8	-
Pool	2x2/2	14x14x8	-
Flatten	-	1568	-
FC	-	1568	803K
MFM	-	-	-
FC	-	256	131K
TOTAL	**-**	**-**	**986K**

Note that the 9-CNN layers architecture we designed is similar to Wu et al. [4] except for the model parameters. The model we designed consists of more parameters because here, we are using 225x225 pixels images throughout our own collected dataset, thus resulting in more pixels processed in the feature extractor. Images consist of 3 color channels and are normalized in the data batch tensor pipeline with real-time data augmentation. Since neural network works best with normalized features, more detail will be provided.

The last number in each output shape is the filter generated by convolutional layers. Intuitively, the architecture could easily pass one observation or batches of images from the dataset to be trained through the network simultaneously. Recall that batches of images determine how many images are in each training step batch, which is then used to update the neural network weights, and the errors are backpropagated. This will result in a performance increase on GPUs, which is also one of the reasons why it is important to constrain the number of parameters in every layer.

C. Support Vector Guided SoftMax

SoftMax loss is the pipeline combination of the last fully connected layer, SoftMax function, and cross-entropy loss. In a neural network, the input vector from the image features is fitted into the hidden layer, resulting in their corresponding *y* label. The weights are randomly stated in the hidden layer, resulting in *x* feature that gets scaled for the last fully connected layer [7]. Therefore, given the input feature vector *x* with its corresponding ground truth label *y,* the final interpretation is fitted to the last activation. The final activation, such as SoftMax, makes it easy to interpret the final prediction against the final raw output.

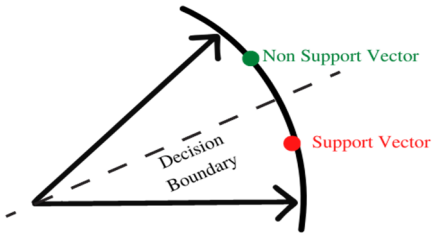


Fig. 3 SV-SoftMax geometrical interpretation with feature perspective. A misclassified label (red circle point) is the support vector that will be optimized [3].

Given that the learning problem is not greatly affected by well-separated feature vectors, it is necessary to increase the feature discriminability by using feature vectors that have been incorrectly classified to increase inter-class separability and intra-class compactness. SV-SoftMax focuses training on useful features by adaptively defining masks that indicate whether samples are selected as support vectors by a particular classifier at the current step [3]. By definition, misclassified samples will be emphasized temporarily using a binary mask defined below:

$$f(x) = \begin{cases} 0, & \cos\left(\theta_{W_y}, x\right) - \cos\left(\theta_{W_y}, x\right) \geq 0 \\ 1, & \cos\left(\theta_{W_y}, x\right) - \cos\left(\theta_{W_y}, x\right) < 0 \end{cases} \quad (1)$$

The sample will be emphasized temporarily if it is misclassified, *i.e.,* $\cos\left(\theta_{W_y}, x\right) - \cos\left(\theta_{W_y}, x\right) < 0$. Intuitively, Wang et al. [3] define SV-SoftMax by focusing on the challenging situations from the probability aspect compared to mining-based loss functions, such as Focal loss., by reducing the probability according to the decision boundary (support vectors). They often introduce a margin function from the standpoint of the ground truth class in margin-based loss functions, such as angular, additive, and additive angular margins. Meanwhile, as defined in SV-SoftMax [3], from the viewpoint of other non-ground truth classes, it widens the feature margin and utilizes its customized margin function for the misclassified features. The SV-SoftMax will be class-specific ground truth margins in multi-class classification for face recognition cases. Recent studies also utilize Support Vector Margin (SV-X-SoftMax) using various perspectives and a particular technique, such as expanding the mining range by using margin-based decision boundaries to create the support vectors. Combining the mining-based and margin-based losses into a single framework is possible. In our scenario, SV-SoftMax is utilized since the proposed loss function is trainable and easily optimized.

Fig. 4 Model Structure. The network consists of a tensor batch input layer and deep CNN with MFM followed by Dropout regularization, then a fully connected dense layer. The SV-SoftMax loss during training follows this.

This section covers the model architecture created using a 9-layer CNN and MFM with SV-SoftMax:

- We utilized tensor image data batch with real-time data augmentation. 32-batch size is used as a standard rule of thumb. The quantity of images needed to train a single forward and backward pass in a neural network is called the batch size. Studies suggested that choosing the higher batch size would not result in higher accuracy [32],
- Similar to VGG, the number of parameters can be reduced by using a small convolution kernel size in the network with MFM. A dropout layer is placed after the final convolution layer to prevent overfitting. Dropout layer values are set to 0.7,
- Similar to the architecture used in Wang et al. [3], a Fully connected layer with 512 filters and 256 filters with facial representations can be used for face verification.

The SV-SoftMax loss is formulated by Wang et al.[3] as:

$$\mathcal{L} = -\log \frac{e^{s\cos(\theta_{w_{y'},\ x})}}{e^{s\cos(\theta_{w_{y'},\ x})} + \sum_{k\neq y}^{K} h(t,\theta w_k,x,I_k) e^{s\cos(\theta_{k_{y'},\ x})}} \quad (2)$$

SV-SoftMax is employed as a loss function. $\cos(\theta_{w_y,x})$ is cosine similarly. w_k are weights, K is the number of classes (where $k \in \{1,2,\dots K\}$. $\theta_{w_k,x}$ is the angel of w_k and x [3]. t is a preset hyperparameter from $h(t,\theta w_k,x,I_k)$ as defined:

$$h(t,\theta w_k,x,I_k) = e^{s(t-1)(\cos(\theta_{w_k,x})+1)I_k} \quad (3)$$

t will be set to 1.05. Furthermore, different values set to t were analyzed.

III. Results and Discussion

In this section, we present an in-depth analysis of the results obtained from our 9-Layer MFM CNN models, comparing their performance to other state-of-the-art models such as VGG-19 and FaceNet as well as 9-Layer MFM CNN without modified SoftMax. We highlight the key findings and insights derived from our experimental evaluation, providing a comprehensive understanding of the capabilities and the comparative analysis.

A. Training

The batch size for training all of the CNN models is 32. To build the model from scratch, TensorFlow is employed. As Pang et al. [33] pointed out, TensorFlow is one of the most widely used libraries for deep learning applications. To this stage, entire models are trained using the same random seed for weights and biases. Moreover, the Adaptive Moment Estimation (Adam) optimizer has a learning rate of 0.00001.

B. Testing

In the validation step, a single model is used to analyze each of the provided outcomes thus, there are no ensemble models. The Receiver Operating Characteristics (ROC) curves will be utilized. Moreover, classification metrics such as f1-score, precision, and recall from each class are adopted due to the imbalanced data in the dataset. The method is contrasted with the baseline methods using Categorical Cross-entropy loss. Next, the 9-layers CNN with MFM compared against the original 9-layers CNN with MFM without modifying the loss functions. Furthermore, 9-layers CNN with MFM using SV-SoftMax with preset t is set to 1.05 will be tested. More details about the effect of the parameter t will also be tested. Finally, we test our constructed model against recent state-of-the-art face recognition models such as VGG, and FaceNet. These models will not be fine-tuned. However, based on the dataset used, we will modify their final output layer to fit the number of classes. Since FaceNet holds the models, we implemented the architecture by referring to Inception ResNet V2 models in their work.

C. Method Comparison and Analysis

In the validation step, a single model evaluates all the reported results; thus, there are no ensemble models. The Receiver Operating Characteristics (ROC) curves will be utilized. Moreover, classification metrics such as f1-score, precision, and recall from each class are adopted due to the imbalanced data in a dataset. We contrast the method with the standard Categorical Cross Entropy loss in comparison to the other baseline methods.

Next, we will compare the 9-layers CNN with MFM against original 9-layers CNN without changing the loss functions. Furthermore, 9-layers CNN with MFM using SV-SoftMax with preset t is set to 1.05 will be tested. More details about the effect of the parameter t will also be analyzed. Finally, we test our constructed model against recent state-of-the-art face recognition models such as VGG and FaceNet. These models will not be fine-tuned. However, based on the dataset used, we will modify their final output layer to fit the number of classes.

TABLE II
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS IN ACCURACY METRICS

Models	Speed (ms/step)	Number of Params	Accuracy
VGG-19 [22]	152 ms/step	20M	87.39
FaceNet [21]	142 ms/step	54M	98.44
9-Layers MFM + SV Softmax	142 ms/step	986K	96.22
9-Layers MFM	143 ms/step	986K	95.38

In the above results, VGG-19 is trained without fine-tuning, but the pre-trained weights are loaded to the model, similar to FaceNet. The final layer, which determines how many classes need to be classified, is changed to 23. All the models are trained in 100 iterations with 199 steps. FaceNet achieved better results. However, the number of parameters is two times bigger than VGG, making the model size bigger. Moreover, CNN-9 Layers MFM with SV-SoftMax is 0,008% better than CNN-9 Layers MFM with the same speed performance per step. Regarding memory requirements, it is obvious that CNN-9 Layers benefit the most since the average filter use is 3x3 filters with stride 1 and the *same* padding, resulting in much faster calculations.

TABLE III
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS IN MACRO AVERAGE METRICS

Models	Precision (%)	Recall (%)	F1-Score (%)
VGG-19 [22]	86.90	91.03	88.03
FaceNet [21]	98.53	98.54	98.52
9-Layers MFM + SV Softmax	95.12	96.12	95.37
9-Layers MFM	94.47	95.79	95.02

When dealing with the imbalanced dataset, we choose metrics other than accuracy to trust the performance of all models. Therefore, f1-score is adopted. F1-score, particularly called *harmonic mean* of precision and recall, gives much more weight to low values of classes. Hence, we get high F1-score if both recall and precision are high. Here, we utilized the macro average metrics type of f1-score. This is mainly because the macro average is computed using the arithmetic mean, thus treating all classes equally important. The precision and recall trade-off for 9-Layers CNN MFM + SoftMax is only 0,01%.

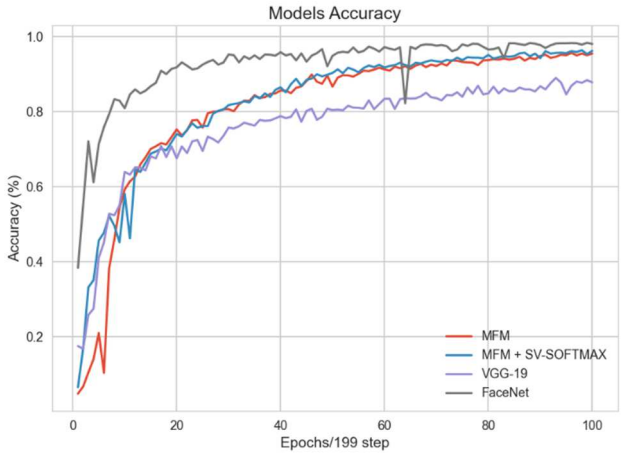


Fig. 5 Models Accuracy. State-of-the art methods performance in accuracy compared with 9-Layers CNN with SV-SoftMax.

The above results show that 9-CNN Layers MFM with SV-SoftMax is slightly better than original MFM layers and VGG-19, and there is only 0,02% below FaceNet accuracy compared to our model. Interestingly, at epochs 60-62 FaceNet performance drastically reduced, whereas 9-Layers CNN MFM with SV-SoftMax and 9-Layers CNN is consistently improved its performance. Although VGG-19 was underperformed compared to 9-Layers MFM, it outperformed the 9-Layers MFM on the first 20 epochs. Similarly, 9-Layers MFM with SV-SoftMax outperforms the 9-Layers MFM without modified SV-SoftMax.

While all of the architectures employed in this scenario are well-suited for the face recognition task, the parameters required to develop effective representations, the choice of activation functions, and the initial weights contribute to this. In this experiment, the initial weights are randomly initialized before training begins, thereby impairing the capacity of the model to acquire meaningful representations during the early phases of training. The model will outperform others during the first few epochs when it starts with a better initial weight.

As shown in Fig. 6, the confusion matrix below proved the precision score we showed earlier in Table 3. Note that each class is labeled based on the person's identity number instead of name. On average, the wrongly predicted maximum is 8 to 9 false positive, and the minimum is 0. Some classes with less than 10 image samples can even classified correctly. Recall that t is the preset parameter for SV-SoftMax. As shown in Fig. 7, we trained 9-Layers CNN MFM with SV-SoftMax for 50 iterations for different t values. First, we trained the model with the previously configured t set to 1.05.

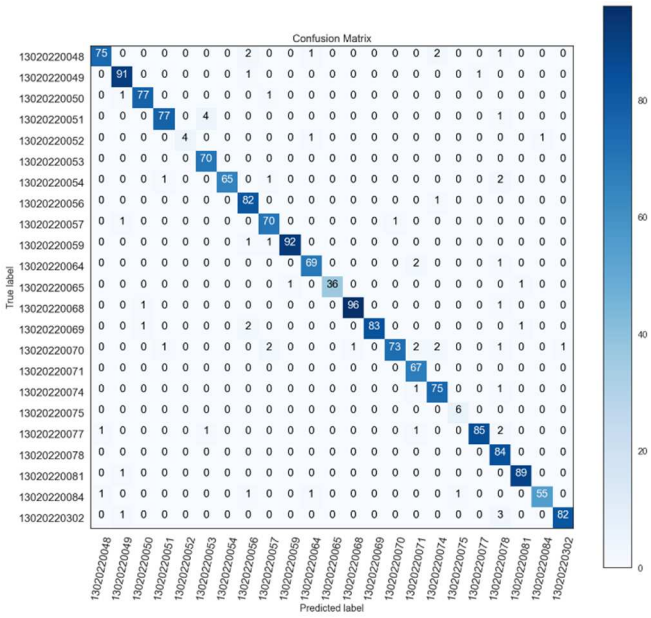


Fig. 6 Confusion Matrix. 9-Layers CNN with MFM + SV-SoftMax from all classes.

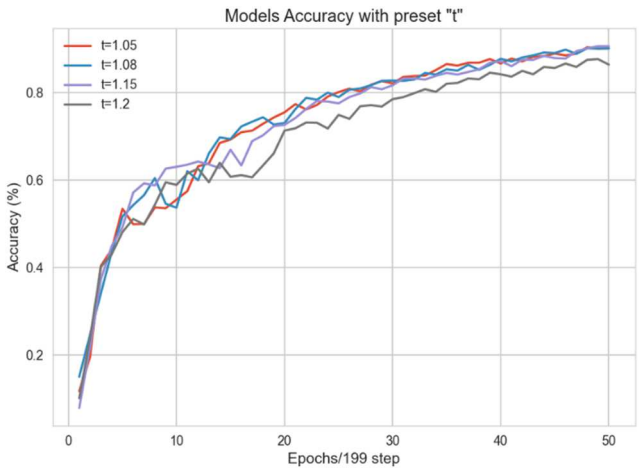


Fig. 7 Hyperparameter. Preset modified parameter t of SV-SoftMax loss.

Then, we slightly increase the parameter to 1.08, 1.15, and 1.2, respectively. This increment is motivated by the performance of these parameters implemented by the original work in [3]. Here, we found that t=1.15 and t=1.05 have the best performance. However, the model still overfits the training data. This can be fixed by increasing the number of iterations so that the model will learn more, as previously done in experiments in Table 2. SV-SoftMax does not contribute to additional parameters in our architecture.

IV. Conclusions

In this study, the CNN-9 Layers MFM architecture was carefully designed to address the challenges of achieving a low-dimensional yet reliable face representation. The use of small convolutional layer kernel sizes proved effective in capturing essential facial features while minimizing the risk of overfitting. The model's performance was also significantly enhanced by reducing its size, making it more efficient and suitable for real-world industry applications. The experimental results showcased the potential of the 9-Layers CNN with MFM and SV-SoftMax over several well-

established CNN methods. Its ability to outperform other models' accuracy and efficiency demonstrates its potential as a practical solution for various facial recognition tasks. With only 986K parameters, it accelerates the training and inference processes and reduces memory requirements, making it feasible for deployment on resource-constrained devices, such as mobile phones or embedded systems.

A significant contribution of this research lies in introducing a customized SoftMax loss function. This loss function effectively targets and improves the classification of misclassified points, allowing the model to focus on crucial areas where it may initially struggle. The findings indicate that the proposed CNN-9 Layers MFM architecture, combined with the SV-SoftMax loss, holds immense potential for real-time facial recognition applications. The ability to balance model complexity and accuracy makes it a favorable choice for practical implementations in security systems, access control, and surveillance applications. In conclusion, this paper presents a compelling approach to address the challenges in facial recognition using the CNN-9 Layers MFM architecture. The results demonstrate the model's efficacy in delivering accuracy and efficiency, making it a promising candidate for widespread adoption across industries in diverse facial recognition systems.

ACKNOWLEDGMENT

This work is funded by the Research and Innovation Scholarship Program from The National Research and Innovation Agency of Indonesia (Badan Riset dan Inovasi Nasional) in collaboration with the Computer Science laboratories of Universitas Muslim Indonesia.

REFERENCES

[1] N. Mohammad, A. M. Muad, R. Ahmad, and M. Y. P. M. Yusof, "Accuracy of advanced deep learning with tensorflow and keras for classifying teeth developmental stages in digital panoramic imaging," *BMC Med. Imaging*, vol. 22, no. 1, 2022, doi: 10.1186/s12880-022-00794-6.

[2] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, "The MegaFace benchmark: 1 million faces for recognition at scale," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.527.

[3] X. Wang, S. Wang, S. Zhang, T. Fu, H. Shi, and T. Mei, "Mis-Classified Vector Guided Softmax Loss for Face Recognition Xiaobo," *arXiv*, 2018.

[4] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 11, 2018, doi: 10.1109/TIFS.2018.2833032.

[5] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *30th International Conference on Machine Learning, ICML 2013*, 2013.

[6] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, 2022, doi: 10.1109/TPAMI.2021.3087709.

[7] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive Margin Softmax for Face Verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, 2018, doi: 10.1109/LSP.2018.2822810.

[8] X. Wang, S. Zhang, Z. Lei, S. Liu, X. Guo, and S. Z. Li, "Ensemble soft-margin softmax loss for image classification," in *IJCAI International Joint Conference on Artificial Intelligence*, 2018. doi: 10.24963/ijcai.2018/138.

[9] F. Huang, M. Yang, X. Lv, and F. Wu, "Cosmos-loss: A face representation approach with independent supervision," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3062069.

[10] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems*, 2014.

[11] G. Gao, Y. Yu, J. Yang, G. J. Qi, and M. Yang, "Hierarchical Deep CNN Feature Set-Based Representation Learning for Robust Cross-Resolution Face Recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, 2022, doi: 10.1109/TCSVT.2020.3042178.

[12] Y. Dong, C. Yang, and Y. Zhang, "Deep metric learning with online hard mining for hyperspectral classification," *Remote Sens.*, vol. 13, no. 7, 2021, doi: 10.3390/rs13071368.

[13] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, 2020, doi: 10.1109/TPAMI.2018.2858826.

[14] X. Li, C. Lv, W. Wang, G. Li, L. Yang, and J. Yang, "Generalized Focal Loss: Towards Efficient Representation Learning for Dense Object Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, 2023, doi: 10.1109/TPAMI.2022.3180392.

[15] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "SphereFace: Deep hypersphere embedding for face recognition," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. doi: 10.1109/CVPR.2017.713.

[16] W. Liu, Y. Wen, B. Raj, R. Singh, and A. Weller, "SphereFace Revived: Unifying Hyperspherical Face Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 2, 2023, doi: 10.1109/TPAMI.2022.3159732.

[17] Z. Zhang, W. Lu, X. Feng, J. Cao, and G. Xie, "A Discriminative Feature Learning Approach With Distinguishable Distance Metrics for Remote Sensing Image Classification and Retrieval," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 16, 2023, doi: 10.1109/JSTARS.2022.3233032.

[18] A. Popielarska, "(LFW) Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," *Neurol. Neurochir. Psychiatr. Pol.*, vol. 5, no. 3, 1955.

[19] Y. Sun, X. Wang, and X. Tang, "Deeply learned face representations are sparse, selective, and robust," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/CVPR.2015.7298907.

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. doi: 10.1109/CVPR.2015.7298682.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[22] H. Pranoto and O. Kusumawardani, "Real-time triplet loss embedding face recognition for authentication student attendance records system framework," *Int. J. Informatics Vis.*, vol. 5, no. 2, 2021, doi: 10.30630/joiv.5.2.480.

[23] X. Liu, H. Wang, and Z. Li, "An Approach for Deep Learning in ECG Classification Tasks in the Presence of Noisy Labels," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2021. doi: 10.1109/EMBC46164.2021.9630763.

[24] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," in *British Machine Vision Conference 2016, BMVC 2016*, 2016. doi: 10.5244/C.30.87.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. doi: 10.1109/CVPR.2016.90.

[26] H. Feng, V. Misra, and D. Rubenstein, "The CIFAR-10 dataset," *Electr. Eng.*, vol. 35, no. 1, 2007.

[27] Y. Zheng, B. Wang, and Y. Zheng, "68 Face Feature Points Detection Based on Cascading Convolutional Neural Network with Small Filter," *Highlights Sci. Eng. Technol.*, vol. 9, 2022, doi: 10.54097/hset.v9i.1731.

[28] V. Thambawita, I. Strümke, S. A. Hicks, P. Halvorsen, S. Parasa, and M. A. Riegler, "Impact of image resolution on deep learning performance in endoscopy image classification: An experimental study using a large dataset of endoscopic images," *Diagnostics*, vol. 11, no. 12, 2021, doi: 10.3390/diagnostics11122183.

[29] L. Parisi, D. Neagu, R. Ma, and F. Campean, "Quantum ReLU activation for Convolutional Neural Networks to improve diagnosis of Parkinson's disease and COVID-19," *Expert Syst. Appl.*, vol. 187, 2022, doi: 10.1016/j.eswa.2021.115892.

[30] S. Sheena and S. Mathew, "Performance Evaluation of New Feature based on Ordinal Pattern Analysis for Iris Biometric Recognition," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 10, 2022, doi: 10.14569/IJACSA.2022.0131058.

[31] H. Alaeddine and M. Jihene, "Deep network in network," *Neural Comput. Appl.*, vol. 33, no. 5, 2021, doi: 10.1007/s00521-020-05008-0.

[32] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, 2020, doi: 10.1016/j.icte.2020.04.010.

[33] B. Pang, E. Nijkamp, and Y. N. Wu, "Deep Learning With TensorFlow: A Review," *Journal of Educational and Behavioral Statistics*, vol. 45, no. 2. 2020. doi: 10.3102/1076998619872761.