IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 October 2023, accepted 3 November 2023, date of publication 7 November 2023,
date of current version 14 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3330897

METHODS

Siamese GC Capsule Networks for Small Sample Cow Face Recognition

ZIHAN ZHANG, JING GAO, FENG XU, AND JUNJIE CHEN
College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot 010011, China
Inner Mongolia Autonomous Region Key Laboratory of Big Data Research and Application for Agriculture and Animal Husbandry, Hohhot 010018, China

Corresponding author: Jing Gao (gaojing@imau.edu.cn)

This work was supported in part by the Inner Mongolia Science and Technology Major Special Projects under Grant 2019ZD016, in part by the Natural Science Foundation of China under Grant 61462070, in part by the Department of Science and Technology of the Inner Mongolia Autonomous Region under Grant 2021ZD0005, and in part by the Natural Science Foundation of Inner Mongolia Autonomous Region under Grant 2019MS03014.

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by the Experimental Animal Welfare and Ethics Committee of Inner Mongolia Agricultural University under Application No. NND2021093.

ABSTRACT Individual cattle identification is pivotal for dairy farming, food quality tracing, disease prevention and control, and registration against fraudulent insurance claims. When employing neural network models for cattle face recognition, challenges arise due to limited individual data, varying cattle face positions and angles, and significant image background noise. This often results in the model's low robustness in recognizing untrained individuals. To address this, we introduce an algorithm based on the Siamese Group Chunking (GC) Capsule Network (SGCCN). Firstly, the GC block serves as the feature extractor for the primary capsules. By utilizing separate filters, the GC block learns the unique representations of cattle faces, enhancing feature extraction capabilities while reducing model parameters. Secondly, a adjusted cosine similarity is employed to capture both directional and absolute numerical differences between cattle face vectors, bolstering the network's robustness. Experimental results reveal that, compared to the conventional Siamese capsule network, the SGCCN reduces parameter usage by 57.65% yet increases recognition accuracy by 7.67%. The recognition rate on the validation set reaches 92.67%, and 89.33% for untrained individuals.

INDEX TERMS Capsule networks, cow face recognition, one time learning, adjusted cosine similarity, Siamese neural network.

I. INTRODUCTION

With the increasing demand for meat and rising standards for food quality, the livestock industry is evolving from small-scale operations to large-scale farming and specialized grazing. To enhance product yield and quality, there's a clear need for automated and precise livestock management to ensure product quality traceability. Dairy cows, as animals of high economic value, produce milk, meat, and other products that are indispensable in our daily lives. Accurate and reliable cow identification plays a vital role in tracking cows from birth to becoming meat products and in preventing false insurance claims. This addresses the mismatch between claimed and insured cows and the issue of food traceability.

In contemporary large-scale livestock farming, the predominant method for cow identification relies on electronic tags based on radio-frequency technology [1], [2]. However, these electronic tags are susceptible to tampering or loss, leading to potential issues of identity substitution and inaccuracies in identification. Moreover, they might inflict physical harm on the cows. Furthermore, during the insurance claim process for dairy cows, challenges frequently arise in determining, through images, whether a particular cow was insured, a dilemma for which no satisfactory solution currently exists. Owing to the biological similarities between cow faces and human faces, both covered in hair

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

and possessing abundant texture elements that might form distinctive patterns, cow faces offer unique biological characteristics for identification. Consequently, various approaches, encompassing both traditional and deep learning methods, have been progressively explored for cow face recognition.

In traditional methodologies, Kim et al. [3] identified Japanese Black cattle using facial images processed through associative neural memory techniques, validating the approach through brightness, distortion, noise, and angle transformations of facial images. Cai et al. [4] introduced the Local Binary Pattern (LBP) feature for extracting local texture attributes from cow faces. Kumar and colleagues [5], [6] presented a comparative analysis of ICA, PCA, LBP, Speeded-Up Robust Features (SURF), and Linear Discriminant Analysis (LDA) for extracting local texture attributes from images at various Gaussian filter levels for automated cow facial recognition. When employing traditional approaches for identification, several challenges arise: as the number of categories increases, feature extraction becomes progressively complex; it's also difficult to pinpoint key features that differentiate various target categories, leading to reduced robustness in traditional models.

Within the realm of deep learning methodologies, Qiao et al. utilized CNN and LSTM (Long Short-Term Memory) networks to recognize beef cattle through image sequences. They harnessed the Inception-V3 network to extract features from a dataset of cow face videos and then fed these extracted features into an LSTM model for training, thereby identifying each individual [7]. Minling and colleagues proposed an algorithm and model for cow face recognition and detection primarily based on CNN, but also integrating ResNet and SVM. This method exhibited faster training convergence compared to conventional CNN architectures [8]. Bisen introduced a deep separable convolutional network named cow-net, utilizing focal loss to address performance issues arising from data imbalances, thus enhancing recognition accuracy [9]. Xu adopted a lightweight backbone network and integrated state-of-the-art face recognition loss functions into the network, putting forth a novel cow face recognition framework that combines lightweight RetinaFace-mobilenet with Additive Angular Margin Loss (ArcFace). This model leverages ArcFace to bolster intraclass compactness and interclass distinctions during the cow face recognition training process [10].

All the aforementioned methods utilize extensive individual data for cow face recognition. In farms, the count of dairy cows ranges from 200 to 1000. However, obtaining numerous facial photographs for each cow is challenging. Throughout the data collection phase, the cow's head is constantly in motion, leading to various orientations of the cow face in images. Consequently, gathering a substantial amount of pertinent data entails considerable time and manpower costs.

In the context of few-shot cow face recognition, Wang et al. achieved facial identification of cows by utilizing the VGG-16 network in conjunction with parameter transfer for few-shot data [1]. Xu proposed a cow face recognition algorithm based on SDBCN. The SDBCN combines the density block and the capsule network, and through the Siamese Network structure, it utilizes the correlation of binary features for cow face image recognition. Furthermore, this approach can recognize cow faces that have not been previously trained [11]. However, it also faces the challenge of having a substantial number of parameters, which can consume significant equipment resources.

In the realm of one-shot image classification, the typical approach is to train a universal model based on labeled samples from training classes and then directly employ the learned model to classify each unlabeled sample in the test classes independently [12], [13], [14], [15], [16], [17]. In contrast, one-shot learning often resorts to metric-based learning methods for training [18]. The objective is to learn a similarity classifier in the feature space by randomly sampling labeled training samples and partitioning them into support and query sets, thereby constructing various episodes. Vinyals et al. introduced a matching network to learn embeddings [13], while Snell and colleagues proposed a prototype network to construct class-centric prototype representations [14]. Additionally, Sung et al. presented a relation network, where a simple neural network can be harnessed to learn non-linear distance metrics, as opposed to utilizing fixed linear metrics [15]. Xu et al. integrated an initiation module into Siamese Neural Networks, a strategy that enhanced the speed and accuracy of face recognition training [19]. During one-shot learning, each category has only a limited number of labeled samples. When no labeled samples are available for a category, one-shot learning transitions into zero-shot learning [20], [21]. Zero-shot learning operates either by leveraging shared attributes for transfer learning or by direct prediction. It employs a network trained in a pre-defined semantic space to determine feature categorizations [15], [22], [23], [24].

In summary, by integrating the principles of few-shot learning and one-shot learning, this paper addresses the challenges of bovine facial recognition where individual data for cattle is limited, and variations in cattle facial positioning and angles are diverse. These challenges are compounded when image backgrounds are noisy, leading to low robustness of models in recognizing untrained individuals. To tackle these issues, we propose a bovine facial recognition algorithm based on the Siamese Grouped Convolution Capsule Network. Leveraging the Siamese structure, we obtain pairwise features and analyze the relationships between these features for effective cattle facial recognition.

The main contributions of this paper are as follows:

- Utilizing the combination of grouped convolution and GC blocks as a shallow feature extractor for the capsule network, each filter is designed to learn unique facial features of cattle and relay them to the capsule network. This enhancement reduces the model's parameter count, while amplifying its capability to extract distinctive representations of bovine faces.

- The adjusted cosine similarity is employed to measure the similarity between two vectors, taking into account not only the directional differences but also emphasizing the impact of numerical discrepancies on the similarity between vectors.
- The proposed network model is adept at addressing the challenge of one-shot learning for bovine facial recognition with limited samples. Moreover, it demonstrates robust performance when confronted with unfamiliar bovine faces, positioning it as a promising tool for one-shot learning scenarios.

The organization of this paper is as follows. Section II presents the relevant technologies and algorithmic formulations. Section III delineates our proposed Siamese GC Capsule Network algorithm. Section IV provides details on the experimental setup. Section V discusses the experimental results. Finally, Section VI concludes the paper.

II. RELATED TECHNOLOGIES

A. SIAMESE NEURAL NETWORKS

The Siamese network is constructed upon a coupled architecture consisting of two artificial neural networks. In the Siamese Neural Network (SNN), two images are concurrently fed into an embedding function composed of multiple convolutional layers for feature extraction [25].The Euclidean distance [26], [27], [28] between the features of the two images is calculated and then transformed into a probability, which is subsequently classified using (4).The Euclidean distance between the features of the two images is calculated and then converted into a probability. This probability is used by (1) to determine if the two images belong to the same category. In (1), σ represents the sigmoid activation function, while α indicates other parameters learned by the model during training.

$$p(x_i, x_j) = \sigma(\alpha|f_\theta(x_i) - f_\theta(x_j)|) \tag{1}$$

SNN employs two identical networks with distinct images as inputs. During computation, parameters are shared across the networks. This architecture processes distinct images through the same feature extraction pipeline, yielding equivalent output features. Typically, SNN utilize a contrastive loss function [29], [30].

Formula (2) is the comparison loss function, where N represents the number of samples, d represents the distance between two features. Euclidean distance is commonly used in Siam networks, and Y is the label of the image pair. When $Y = 1$, two images belong to the same category, and L minimizes the distance between the two features. When $Y = 0$, two images belong to different categories. If the distance between the two features is less than M, the distance between the two features increases to M.

$$L = \frac{1}{2} * \frac{1}{N} \sum_{n=1}^{N} [(Y)(d)^2 + (1 - Y)\{\max(0, M - d)\}^2] \tag{2}$$

B. CHANNEL GROUPING

Channel grouping facilitates the learning of distinct features across different groups. This enhances the model's ability to capture diverse aspects and characteristics of the input data, thereby bolstering its representational capacity.Channel grouping has found a multitude of applications in the realm of neural network architectures.

The utilization of channel grouping has been extensively employed in various neural network architectures. In grouped convolution, the input feature maps are divided by channels into n distinct groups, with each group subjected to standard convolution operations. Following this division, the channel count for each subset of the feature map is effectively reduced to 1/n. As a result, the channel count for each convolutional kernel is subsequently decreased to 1/n. This methodology inherent to grouped convolution considerably diminishes both computational requirements and the total parameter count. Qing-Long Zhang introduced a mechanism termed 'Shuffle Attention' (SA) [31], which segments the channel dimensions into multiple sub-features, subsequently processing them in parallel. For each of these sub-features, SA employs a shuffle unit to encapsulate dependencies across both spatial and channel dimensions. Following this, all the sub-features are aggregated. Finally, a 'channel shuffle' operator is invoked, ensuring efficient communication of information between different sub-features. Wu Rong introduced a mechanism termed 'Channel Group-wise Drop' (CGD) module [32]. This model leverages group-level channel attention to aggregate activations of certain objects and enhance fine-grained features. However, the response within these channel groups isn't particularly pronounced. Thus, by employing a group-level mask to randomly eliminate some positional responses, activations at the same positions are invigorated, mitigating redundant aggregations within the channel group. Consequently, an appropriate channel grouping of features proves beneficial for both fine-grained feature extraction and feature aggregation.

C. CAPSULE NETWORK

The network was originally proposed by Hinton and colleagues, introducing a neural unit termed as 'capsule'. Within these capsules, individual activations no longer represent distinct features, but rather various attributes of the same entity.Subsequently, Sabour et al. [34] introduced the first architecture named CapsNet. Capsule networks instantiate parameters by encapsulating convolutional neurons into feature vectors of neurons representing specific entity types. Each capsule encompasses spatial information, such as position, texture, orientation, and the probability of a specific entity's presence. This preserves the part-to-whole relationship between sub-objects and the main object, thereby enhancing feature detection and recognition.

Traditional capsule networks are frequently utilized for the MNIST dataset. Input images undergo two convolutions with a kernel size of 9, resulting in features of a $6 \times 6 \times 32 \times 8$ dimension. These features are reshaped to 1152×8,

indicating 1152 output capsules, with each capsule characterized by an 8-dimensional vector.These vectors are linearly combined with the weights w to produce the output u_i. The weight w encodes the spatial relationships and other significant associations between low-level and high-level features.The vector u_i is processed through a dynamic routing mechanism three times, ultimately resulting in 10 high-level capsules of length 16.

The dynamic routing algorithm employs an iterative process, updating coefficients based on the consistency between input and output capsules.Initially, for all input lower-level capsules i and output higher-level capsules j, a temporary variable b_{ij}^r is defined, with its initial value set to 0. r represents the r-th routing iteration.

At the start of the first iteration, the weighting coefficients of the lower-level capsules, denoted as c_{ij}^1, are determined using (3). The softmax function ensures that all these weights remain non-negative and their sum is unity. Initially, all values of c_{ij}^1 are equivalent, but as the iterations advance, this uniform distribution undergoes modifications.

$$c_{ij} = \text{softmax}(b_{ij}) \tag{3}$$

Using (4), the output vector s_j^1 is computed by taking a weighted sum of u_i with the capsule weights c_{ij}^1. Subsequently, the vector s_j^1 is processed through (5) to obtain the high-level capsule vector v from the first iteration. (5) defines a squeezing function that preserves the direction of the vector while constraining its magnitude to the range [0, 1].

$$s_j = \sum_i c_{ij} u_{j|i} \tag{4}$$

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \cdot \frac{s_j}{\|s_j\|} \tag{5}$$

Finally, the subsequent value of b_{ij}^2 for the next iteration is determined by (6). In (6), a dot product is first performed between v_j^1 and $u_{j|i}$, updating the weights by assessing the similarity between the input and output of the capsule. The iterations then proceed, and upon completion of three cycles, a 160-dimensional vector is outputted.

$$b_{ij}^{r+1} = b_{ij}^r + v_j \cdot u_{j|i} \tag{6}$$

However, the dynamic routing in capsule networks consumes a vast amount of parameters, leading to a more resource-intensive and time-consuming training and inference process. To address this, researchers have shifted towards modifying convolution kernel sizes and employing channel grouping [35], [36]. Specifically, they utilized 3 × 3 convolutions and depthwise separable convolution operations for discriminative learning, maximizing the use of the network's filters to capture all capsule-centric features. This approach substantially streamlines and reduces the number of parameters needed in the capsule formation, resulting in decreased computational complexity and expedited training and inference times.

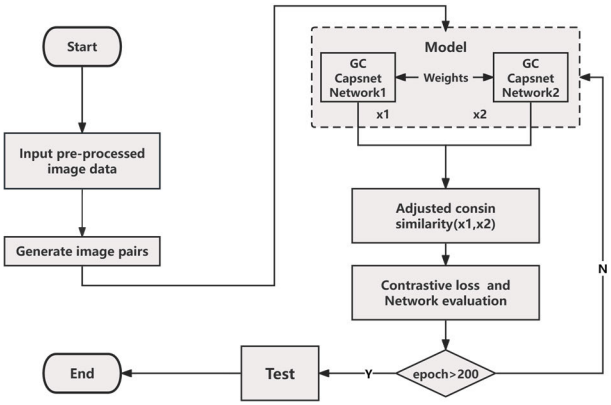


FIGURE 1. Flowchart of SIAMESE GC CAPSULE NETWORK algorithm.

III. SIAMESE GC CAPSULE NETWORK

In this paper, we introduce the Siamese GC Capsule Network (SGCCN) algorithm. Within SGCCN, two identical GC Capsule Networks act as feature extractors, enabling concurrent feature extraction from two images. The weights are shared between the two GC Capsule Networks, ensuring consistent feature acquisition from both sub-networks for identical images. The adjusted cosine similarity metric is employed to gauge the similarity between the two input images, addressing the one-shot learning challenge for small-sample bovine facial recognition. Figure 1 illustrates the flowchart of the SGCCN algorithm, which takes a pair of images and their label as input. When the two images belong to the same category, the label is set to 1; conversely, if the images pertain to different categories, the label is assigned a value of 0.

A. GENERATE IMAGE PAIRS

In our study, we create a binary dataset by selecting pairs of images at random, detailed in Algorithm 1. The input to this algorithm consists of the directory path where images are stored and the number of image pairs desired for generation. The resultant output is a list, containing the paired images accompanied by their associated labels. The algorithm iterates over the specified number of pairs to be generated. To circumvent any imbalance in the number of pairs from the same category versus those from different categories, the label assigned to each image pair is randomly set to either 0 or 1. A label of 1 signifies that the images within the pair are from the same category, whereas a label of 0 indicates that they belong to different categories. This method of random label generation ensures an equitable probability of generating both positive and negative pairs.

B. GC CAPSULE NETWORK

The GC Capsule Network consists of several layers, including the input layer, convolutional layer, group convolutional layer, GC layer, PrimaryCaps layer, CowFaceCaps layer, and the output layer. Figure 2 provides a schematic representation of the GC Capsule Network architecture.Compared

Algorithm 1 Pseudocode for Building an Image Pairs

Input: image folder path img_dir; image pairs number num
Output: img_pairs[]

```
Start
Defifine empty lists img_pairs[]
folder_dataset = dset.ImageFolder(root=img_dir) #image path
for i in range(num)
    img_0, label_0 = random.choice(folder_dataset.imgs)
    should_get_same_class = random.randint(0,1)
    if should_get_same_class
        while True
            #keep looping till the same class image is found
            img_1, label_1 = random.choice(folder_dataset.imgs)
            if label_0==label_1
            break
    else
        while True
            #keep looping till a different class image is found
            Img_1, label_1 = random.choice(folder_dataset.imgs)
            if label_0 == label_1
                break
    img_pairs.append([img_0,img_1,should_get_same_class])
end for
End.
```

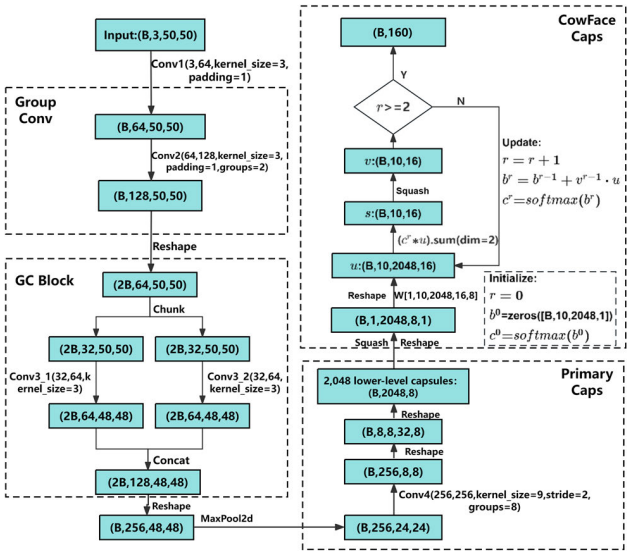


FIGURE 2. Structure of the GC capsule network model.

to the conventional capsule networks, the GC Capsule Network substitutes the standard convolutions with grouped convolutions and integrates the GC block as a shallow feature extractor.Grouped convolution learns the block-diagonal sparsity in the channel dimension in a more structured manner. The GC block facilitates diverse feature extraction across different channels of bovine facial features, where each filter group captures a unique representation of the cow's face.This enhancement significantly reduces the number of parameters in the network and mitigates overfitting. A capsule is a carrier with numerous neurons, where each value in the vector neurons represents a specific attribute, such as pose, deformation, color, texture structure, and so forth. By integrating the GC block as an early-stage feature extractor, independent representations of the cow's face are fed into the higher-level capsules. This amplifies the expressive capacity of the values in the vector neurons concerning the attributes of bovine facial features.

When a bovine face image with dimensions of 50 × 50 is fed into the network, it first traverses the convolutional layer, conv1. Here, a 3 × 3 convolution kernel is employed to extract 64 layers of features. Subsequently, features are extracted through a grouped convolutional layer. This layer consists of a 3 × 3 convolution, Conv2, with a grouping parameter set to 2, resulting in feature maps of size 128 × 50×50. Although we experimented with more extensive feature grouping, we observed that it challenges the efficient extraction and aggregation of features. Consequently, we restricted our approach to bifurcate the convolution into two groups at this layer.

Following this, the features pass through a GC block. Within the GC block, a shuffle unit segregates the feature maps into two groups. Each group of features undergoes a 'chunk' operation, splitting them into two distinct feature blocks. These blocks are then fed into convolutional layers with 3 × 3 kernels, labeled Conv3_1 and Conv3_2, respectively. Post convolution, the results from these two feature blocks are concatenated. After shuffling, each group of features is restored to a dimensionality of 256 × 48×48. The GC block aids in refining deep semantic features and retains the more pertinent information within each feature group. Subsequently, these features pass through a max-pooling layer, resulting in shallow feature maps with dimensions of 256 × 24×24.

Subsequently, the shallow features are fed into the PrimaryCaps layer. Through a convolutional operation, Conv4, characterized by a kernel size of 9 × 9, a stride of 2, and 8 groups, a feature representation of dimensions 32 × 8×8 × 8 is obtained. This representation is transformed into 2,048 lower-level capsule units, with each capsule encapsulating an 8-dimensional vector. These lower-level capsules, upon progression through the CowFaceCaps layer, culminate in 10 higher-level capsules, where each is described by a 16-dimensional vector. The CowFaceCaps layer employs a tri-fold dynamic routing algorithm and a Softmax function to regulate the count and dimensions of the capsules between the two layers. Ultimately, a 160-dimensional vector is the output from the higher-level capsules. When the SGCCN is presented with a pair of images, the model produces two 160-dimensional vectors for the pair, and these vectors are utilized to compute the similarity between the images.

$$v_j = \frac{||s_j||^2}{m + ||s_j||^2} \cdot \frac{s_j}{||s_j||} \tag{7}$$

In the three iterations of the dynamic routing algorithm, a Squash activation function is employed. This function compresses the vector to the interval [0, m] while preserving its original direction. The Squash function is defined by (7),

where s_j represents the sum of the weighted outputs from all primary capsules, v_j is the value post-compression, and m is the compression ratio. m is a hyperparameter, and in this study, it is set to 0.5.

C. ADJUST CONSIN SIMILARITY

In Siamese Neural Networks (SNN), the Euclidean distance is commonly employed to ascertain whether two features belong to the same class. Features from samples within the same class tend to be closer, while those from different classes are more distant. However, unlike traditional methods that use scalars to represent features, capsule networks utilize vectors to depict them.Thus, when gauging the distance between two capsule vectors, one shouldn't only consider the absolute numerical differences, such as the Euclidean distance, but also the variations in the vector directions, like the cosine similarity and Pearson coefficient. Cosine similarity predominantly distinguishes differences in direction and is less sensitive to absolute magnitudes.

When measuring the distance between two vectors, we believe that both the magnitude difference and directional difference should be taken into consideration. To enhance the sensitivity of cosine similarity to the numerical differences between two vectors, we employ a adjusted cosine similarity to assess the resemblance of two feature vectors. Unlike the Pearson correlation, where vectors are subtracted from their respective means, in the adjusted cosine similarity, both vectors are subtracted from the overall mean of the two vectors, as shown in (8) and (9).

In (9), x_i and x_j are the actual variables, c represents the overall mean of the two vectors, and d denotes the adjusted cosine similarity between the vectors.

When vectors are compressed to the range [0, m] through the squeezing function in (8), their direction remains unchanged, where m represents the compression ratio. Subsequently, when both vectors are subtracted by the overall mean, the resulting effect is that as the numerical difference between the two vectors increases, the angle between them grows larger, leading to a reduced cosine similarity. Therefore, the adjusted cosine similarity not only captures the directional difference between the two vectors but also their numerical disparities.

$$c = \frac{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}{2n} \tag{8}$$

$$d = \frac{\sum_{i=1}^{n} ((x_i - c) \times (y_i - c))}{\sqrt{\sum_{i=1}^{n} (x_i - c)^2} \times \sqrt{\sum_{i=1}^{n} (y_i - c)^2}} \tag{9}$$

D. CONTRASTIVE LOSS

Contrary to the Euclidean distance, a smaller value in the adjusted cosine similarity indicates a lower similarity between the two vectors. The objective function is represented by (10).

$$L = \frac{1}{N} \sum_{n=1}^{N} [(1 - Y) \times (d)^2 + (Y) \times \{max(0, M - d)\}^2] \tag{10}$$

In (10), d denotes the similarity between two features calculated using the adjusted cosine similarity, and Y is the label for the image pair. When $Y = 1$, the two images belong to the same category, and L aims to minimize the angle between the two vectors, making d approach M. Conversely, when $Y = 0$, the two images belong to different categories, and L seeks to maximize the angle between the two vectors, pushing d towards 0.

IV. EXPERIMENTS

A. DATA SETS AND PREPROCESSING

We conducted method comparison experiments using three datasets: the small-scale cow face dataset provided by the laboratory, the CASIA WebFace dataset, and the LFW face dataset.

The laboratory-curated small-scale cow face dataset is derived from facial images of 130 cows captured in their natural habitat. These images span various poses, including upward glances, downward tilts, and head turns to the left or right. The 130 cows were sequentially numbered from 0 to 129. Each cow was treated as an individual class, with each class encompassing 15 RGB images depicting different poses, aggregating to 1,950 cow face images. All these images were resized to 50 × 50 pixels, thus constituting a small-scale dataset. For one-shot learning, 100 cows were randomly selected from the cohort of 130. From each cow, 10 facial images were randomly selected for training, leaving the remaining 5 images for validation. 600 image pairs were subsequently generated from the training data to form the training set, while 300 image pairs were derived from the validation data to create the validation set. The images from the remaining 30 cows were utilized as test data, from which 300 image pairs were generated to compose the test set.

The CASIA-WebFace dataset [37] encompasses a total of 494,414 face images, representing 10,575 distinct identities. Contrastingly, the LFW dataset [38] contains over 13,000 face images predominantly sourced from the internet, with approximately 1,680 individuals represented by more than one image. Within the LFW dataset, a specific subset of 6,000 paired face images is designated as the standard test set for unconstrained face verification. Although this positions LFW as a comprehensive testing benchmark, it remains inadequate for training due to the majority of identities in LFW being characterized by a single face image. As a result, much like contemporary high-performance face verification algorithms [39], we are compelled to rely on a more expansive external dataset for our training endeavors. In this light, we employ CASIA-WebFace for training purposes and reserve LFW for testing.

B. EXPERIMENT

The experiments were conducted on a machine equipped with a Tesla P40 23 GB GPU and operated on the Centos 7.9 platform with the Pytorch 1.10.0 deep learning framework. The evaluation metrics employed for the experiment included accuracy, F1-score, and loss. The F1-score serves as a composite measure of recall and precision, as defined in (11). A higher F1-score indicates a more robust classification model.

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{11}$$

In the experiments presented in this paper, the Adam optimizer was employed with a learning rate of 0.0001. The parameter m in the squash function was set to 0.5. Each experimental epoch consisted of 200 training iterations with a batch size of 32. Experimental results revealed that the loss converges most effectively when the margin value of the contrastive loss function is set to 1.

We orchestrated four sets of experiments targeting bovine face recognition using a small-sample bovine face dataset crafted in our laboratory. In the first set of experiments, under consistent conditions, our proposed Siamese GC Capsule Network (SGCCN) was benchmarked against other Siamese deep convolutional network models. In the second set of experiments, we juxtaposed SGCCN with other Siamese networks utilizing the adjusted cosine similarity metric. In the third experimental set, we investigated the effects of different group counts in the primary capsule group convolution within our SGCCN. For the fourth set, under the condition where $M = 1.7$ in the loss function, we compared the performance results between our SGCCN and the Siamese Dense Block Capsule Network (SDBCN). Ultimately, to corroborate the effectiveness of the proposed methodologies in this manuscript, we conducted evaluations on the LFW dataset.

V. RESULTS AND DISCUSSION

In this paper, we perform comparison experiments using SGCCN and several Siamese deep convolutional networks. To ensure the fairness of the comparison experiments, we keep the same parameter sizes as much as possible, and in the following subsections, we present the details of each network separately.

A. COMPARISON OF DIFFERENT SIAMESE NETWORKS

Under identical conditions, the experiment compared the feature extraction capabilities of the SGCCN with other Siamese deep convolutional networks. Within the experiment, networks utilizing cosine similarity are denoted as _C.

1. SNN_C represents a Siamese deep convolutional network employing standard convolution. In this study, the SNN_C network serves as the baseline.

2. SCN_C denotes a Siamese capsule network utilizing standard convolution.

3. SGN_C represents a Siamese network that employs grouped convolution and GC blocks.

TABLE 1. Comparison of SGCCN and other Siamese capsule network validation set results.

Model	Accuracy (%)	F1 (%)	Parameter(M)	Loss
SNN_C	85.00	84.95	7.396	0.1180
SGN_C	88.33	88.29	6.807	0.1067
SCN_C	88.00	87.50	7.937	0.1103
SGCCN_C	**90.00**	**89.58**	**3.361**	**0.0986**

4. SGCCN_C denotes a Siamese capsule network that incorporates grouped convolution and GC blocks.

Comparative experiments were conducted on a bovine face dataset constructed in our laboratory. Table 1 presents the validation results. In comparison to SNN_C, the parameter size of SGCCN_C is reduced by 54.56%. Relative to SGN_C, it sees a reduction of 50.62%, and when compared to SCN_C, there's a decrease of 57.65%. Nevertheless, the accuracy of SGCCN_C on the validation set surpasses that of SNN_C by 5%. Moreover, SGCCN_C's accuracy exceeds SGN_C's by 1.67% and outperforms SCN_C's by 2%. Due to the substantial number of parameters in the aforementioned models, we employed Dropout to prevent overfitting. We attempted to enhance the feature extraction by increasing the depth of grouped convolutions. However, results indicated that as the depth of the network increased, the recognition rate declined. Moreover, in recent literature related to capsule networks, it was observed that with an increase in model depth, capsules tend to become inactive. Such inactive capsules are no longer activated [40].

Experimental results demonstrate that SGCCN, by employing grouped convolutions and GC blocks, excels in extracting low-level facial features of cattle, such as color, texture, and edges. Subsequently, the model leverages capsules to capture instantiated spatial vector information. As a result, SGCCN extracts more comprehensive features compared to other capsule-based networks.

B. A COMPARATIVE ANALYSIS OF RESULTS WAS CONDUCTED BETWEEN SGCCN AND OTHER SIAMESE NETWORKS UTILIZING THE ADJUSTED COSINE SIMILARITY METRIC

In this study, various network architectures were assessed utilizing the adjusted cosine similarity, denoted as _AC. Since cosine similarity is non-parametric, the parameter count remains consistent between SGCCN_AC and SGCCN_C. Leveraging the adjusted cosine similarity, SGCCN_AC achieved an accuracy rate of 92.67%.

Table 2 presents a comparative analysis of validation results between the adjusted cosine similarity and the traditional cosine similarity across different Siamese networks. The experiments reveal that the SNN_AC network, which employs the adjusted cosine similarity, achieves a 0.67% accuracy improvement over the SNN_C network that uses the traditional cosine similarity. Similarly, the SCN_AC network

TABLE 2. Validation results of SGCCN were compared with other Siamese networks using the adjusted cosine similarity metric.

Model	Accuracy (%)	F1 (%)	Loss
SNN_AC	85.67	84.75	0.1156
SGN_AC	88.33	88.22	0.1072
SCN_AC	88.33	88.14	0.1077
SGCCN_AC	**92.67**	**92.36**	**0.0908**

witnesses a 0.33% enhancement in accuracy compared to SCN. Notably, the SGCCN_AC network, when leveraging the adjusted cosine similarity, sees a substantial accuracy boost of 2.67% over the SGCCN_C that relies on the traditional cosine similarity.

Both SNN and SGN exhibited a decline in performance upon adopting the adjusted cosine similarity. Given that both SNN and SGN are convolutional networks, the adjusted cosine similarity measures the likeness of two vectors by subtracting the overall mean of the two vectors. This can pose an issue when two images of the same class, once processed through the model, output vectors with small angular differences but significant magnitude discrepancies. The subtraction induced by the adjusted cosine similarity can lead to an increased angle between the two vectors, causing the similarity to approach values below zero. This phenomenon can result in mispredictions for pairs of images from the same class, subsequently leading to a decline in overall performance.

Both SCN and SGCCN are convolutional capsule networks. Within the realm of capsule networks, dynamic routing is implemented based on the dot product between input and output capsules, aligning the direction of identical entity features within the vectors. Moreover, capsule networks deploy the squash function. This function serves to constrain the vector magnitudes within a specific range while preserving their original directionality. When outputting vectors with small angular differences and pronounced magnitude disparities, the adjusted cosine similarity experiences only a marginal decrement. This reduction is less pronounced than in SNN and SGN. Therefore, when employing adjusted cosine similarity, SCN and SGCCN exhibit superior robustness in their models.

For the one-shot learning challenge, we evaluated 300 image pairs generated from 30 cattle individuals that had not previously undergone model training, with the results detailed in Table 3. The experimental findings suggest that the SGCCN_AC model, through the mechanism of grouped convolutions and GC blocks, carries out channel group convolution. Each filter group distinctively identifies facial features of cattle. The integration of the capsule network enables the model to detect subtle pattern shifts in images, such as the lateral movements observed in cattle faces. Furthermore, by utilizing the adjusted cosine similarity, the model amplifies the sensitivity of vector magnitudes to cosine similarity, effectively diminishing the impact of background noise on cattle face images. SGCCN_AC achieved a notable accuracy rate of 89.33% on the test set. This outcome robustly attests to the prowess of the SGCCN_AC network in adeptly recognizing cattle faces that were not part of the training dataset.

TABLE 3. Test results from 30 bovines without prior model training.

Model	Accuracy (%)	F1 (%)
SNN_C	83.00	83.39
SNN_AC	83.67	84.24
SGN_C	84.00	84.71
SGN_AC	84.00	85.00
SCN_C	82.33	82.85
SCN_AC	83.00	83.28
SGCCN_C	86.33	86.47
SGCCN_AC	**89.33**	**88.97**

C. COMPARING THE NUMBER OF DIFFERENT GROUPINGS FOR GROUPED CONVOLUTIONS IN PRIMARY CAPS

Although employing depthwise separable convolutions in capsule networks [35], [36] can reduce the number of parameters and computational time, our experiments revealed that they fail to comprehensively and effectively extract facial features from the channel information of cattle. The division of channels through this method is overly isolated. Even when channel mixing was implemented following the depthwise separable convolutions, it did not sufficiently aggregate the facial features of cattle. Consequently, we refrained from segmenting each channel into separate groups. Instead, we prioritized keeping a larger number of channels within each group, ensuring the filters adeptly capture the intrinsic facial features of cattle.

To investigate the influence of the number of groups on extracting bovine facial features, we conducted various groupings for the 9×9 convolutions in the primary capsule. As shown in Table 4, the network attained its peak recognition accuracy when the grouping was set to eight.

D. COMPARING SGCCN WITH SIAMESE DENSE BLOCK CAPSULE NETWORK (SDBCN) WHEN M=1.7 IN THE COMPARISON LOSS FUNCTION

In our study, we juxtaposed the SGCCN methodology with the SDBCN technique as delineated in [11]. For the sake of experimental fairness, the hyperparameters within the networks were maintained consistent. Specifically, the hyperparameter M in the contrastive loss was set at 1.7, and m in the squeeze function was established at 0.5. As evidenced by Table 5, SGCCN_C surpasses SDBCN_C by 4% on the validation set. Furthermore, SGCCN_AC employing the adjusted cosine similarity outstrips SDBCN_P that utilizes the Pearson correlation coefficient margin of 5.34%. Intriguingly, the SGCCN_AC network exhibits consistent accuracy outcomes for both $M = 1$ and $M = 1.7$. Notably, when set at $M = 1.7$,

TABLE 4. Main capsule convolution test results with different number of groups.

Groups	Accuracy (%)	F1 (%)	Parameter(M)
G=2	87.33	87.16	5.352
G=4	88.33	87.84	4.025
G=8	**89.33**	**88.97**	3.361
G=16	86.00	85.11	3.029

TABLE 5. M=1.7 Validation results.

Model	Accuracy (%)	F1 (%)	Loss
SDBCN_C	86.00	86.79	0.4319
SDBCN_P	87.33	87.50	0.4319
SGCCN_C	90.00	89.73	0.4371
SGCCN_AC	**92.67**	**92.47**	**0.4284**

the convergence speed of the loss is slower compared to that at $M = 1$. This substantiates the robustness of the SGCCN model, emphasizing its adaptability to varying M values.

In Figure 3, the image pairs on the left depict the test results from SCN, while those on the right represent the outcomes from SGCCN_AC. Within these pairs, when the label is 0, the images on the left and right are from different individuals; conversely, when the label is 1, both images are from the same individual. Predictions align similarly: a prediction of 0 suggests the model anticipates the images as being from different individuals, while a prediction of 1 indicates the same individual. The final numeric value presented signifies the cosine similarity between the image pairs. When the cosine similarity is less than 0.5, the prediction is 0, and when it exceeds 0.5, the prediction becomes 1.

From the presented images, it's discernible that SGCCN_AC demonstrates enhanced recognition capabilities for unique color patterns and facial features of the cattle. Furthermore, it can effectively detect rudimentary pattern shifts. The incorporation of the adjusted cosine similarity heightens the sensitivity to vector magnitudes. In culmination, SGCCN_AC manifests considerable robustness in recognizing cattle individuals that it hasn't been trained on.

E. STABILITY EXPERIMENTS FOR SGCCN

At present, the majority of face verification techniques attain high performance by leveraging extensive training data. We employed the CASIA-WebFace dataset for training and subsequently tested on the LFW dataset. As can be inferred from the results in Table 6, our method outperforms 3DMM [41] by a margin of +0.52%. However, the accuracy of SGCCN falls short when compared to Deepfacehe [42] and PSI [43]. This discrepancy arises primarily because human facial skin tones are relatively uniform, making it easy for models to detect and extract pertinent features from images.

In contrast, while cattle faces are conspicuously colored, capturing them often results in the bovine face and body

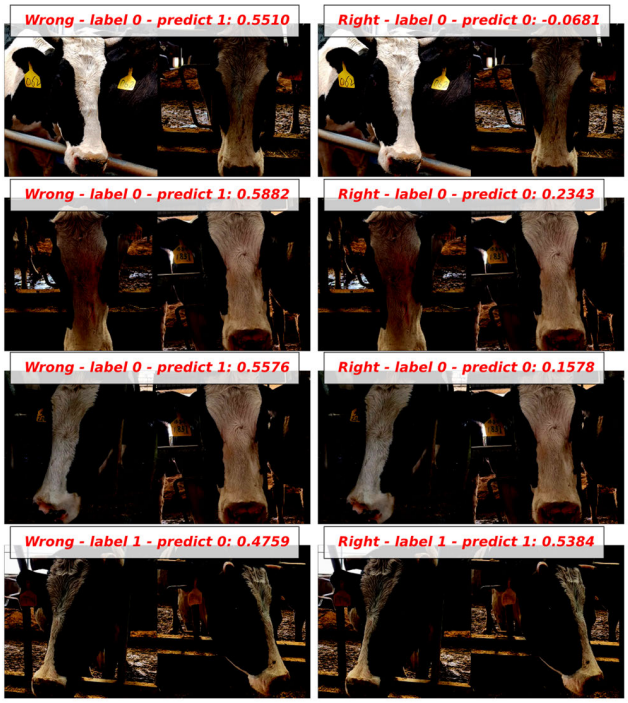


FIGURE 3. SCN model and SGCCN model cow face recognition results.

TABLE 6. LFW test results.

Model	Accuracy (%)
3DMM[41]	92.35
DeepFace[42]	95.92
DeepFace-Siamese[42]	96.17
PSI[43]	98.87
SGCCN(Proposed Network)	92.87

being oriented in the same direction. Consequently, images of cattle faces inadvertently feature portions of their bodies, the color of which closely resembles that of the face. This introduces significant background noise in the images. Such occurrences are inevitable, given they mirror real-world photography scenarios.

The SGCCN, however, is purposefully tailored for cattle face recognition. The GC block adeptly extracts color, texture, and contour features from images, shifting the model's focus predominantly onto the cattle face. The adjusted cosine similarity not only reveals directional variations but also differentiates based on absolute magnitudes. Thus, this model excels in recognizing cattle faces. However, given that human faces lack vibrant colors and distinctive textures, its performance in human face recognition remains unremarkable.

VI. SUMMARY

Cattle face recognition presents numerous challenges such as limited individual data availability, varied facial positions and angles, interference from image background noise, and reduced robustness in recognizing untrained individuals. To address these issues, this paper introduces the SGCCN,

designed specifically to tackle the challenges of cattle face recognition. The SGCCN integrates grouped convolution and the GC block to serve as a shallow feature extractor within the capsule network framework.

Grouped convolution and the GC block enable differentiated feature extraction from various channels of cattle face feature maps. Each filter group learns a unique representation of the cattle face, significantly reducing the number of parameters in the network while preventing overfitting. By integrating the GC block's shallow feature extractor, the distinct representations of cattle faces are fed into higher-level capsules. This enhances the expressive power of vector neurons with respect to cattle face feature attributes, addressing issues of limited individual cattle data and varied cattle face positions and angles.

The SGCCN employs a adjusted cosine similarity metric to gauge the likeness between two vectors. This not only accounts for the directional discrepancy between the two vectors but also emphasizes the numerical differences in their similarity. The experiments were conducted on a small-scale laboratory cattle face dataset, which comprises 130 cattle with 15 images per individual. In the tests, the SGCCN was juxtaposed with the classical Siamese capsule network and its variations. The SGCCN achieved an accuracy rate of 92.67% for recognizing trained cattle individuals and 89.33% for untrained ones. Compared to other Siamese capsule networks, it demonstrated significant enhancements in recognition accuracy and robustness, all while using fewer parameters. Additionally, the SGCCN exhibits strong robustness in recognizing untrained individuals, effectively addressing the one-shot learning challenge in cattle face recognition.

ACKNOWLEDGMENT

The authors appreciate the two modern farms in Inner Mongolia Autonomous Region of China for their courteous help in data collection.

REFERENCES

[1] H. Wang, J. Qin, Q. Hou, and S. Gong, "Cattle face recognition method based on parameter transfer and deep learning," *J. Phys., Conf.*, vol. 1453, no. 1, Jan. 2020, Art. no. 012054, doi: 10.1088/1742-6596/1453/1/012054.

[2] S. Kumar, S. Tiwari, and S. K. Singh, "Face recognition of cattle: Can it be done?" *Proc. Nat. Acad. Sci., India Sect. A, Phys. Sci.*, vol. 86, no. 2, pp. 137–148, Jun. 2016, doi: 10.1007/s40010-016-0264-2.

[3] H. T. Kim, Y. Ikeda, and H. L. Choi, "The identification of Japanese black cattle by their faces," *Asian-Australas. J. Animal Sci.*, vol. 18, no. 6, pp. 868–872, Jun. 2005, doi: 10.5713/ajas.2005.868.

[4] C. Cai and J. Li, "Cattle face recognition using local binary pattern descriptor," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, Kaohsiung, Taiwan, 2013, pp. 1–4, doi: 10.1109/APSIPA.2013.6694369.

[5] S. Kumar, S. Tiwari, and S. Kumar Singh, "Face recognition for cattle," in *Proc. 3rd Int. Conf. Image Inf. Process. (ICIIP)*, Dec. 2015, pp. 65–72, doi: 10.1109/ICIIP.2015.7414742.

[6] S. Kumar, S. K. Singh, T. Dutta, and H. P. Gupta, "A fast cattle recognition system using smart devices," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 742–743.

[7] Y. Qiao, D. Su, H. Kong, S. Sukkarieh, S. Lomax, and C. Clark, "Individual cattle identification using a deep learning based framework," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 318–323, 2019, doi: 10.1016/j.ifacol.2019.12.558.

[8] M. Zhu, L. Zhao, and J. Shou, "Research and implementation of a model of cow face recognition system combining CNN with SVM and ResNet," *J. Chongqing Univ. Technol.*, vol. 36, no. 7, pp. 155–161, 2022.

[9] B. Xie, "Research on cow face recognition technology based on convolutional neural network," M.S. thesis, School Softw., Yunnan Univ., Kunming, China, 2020, doi: 10.27456/d.cnki.gyndu.2020.000644.

[10] B. Xu, W. Wang, L. Guo, G. Chen, Y. Li, Z. Cao, and S. Wu, "CattleFaceNet: A cattle face identification approach based on RetinaFace and ArcFace loss," *Comput. Electron. Agricult.*, vol. 193, Feb. 2022, Art. no. 106675, doi: 10.1016/j.compag.2021.106675.

[11] F. Xu, J. Gao, and X. Pan, "Cow face recognition for a small sample based on Siamese DB capsule network," *IEEE Access*, vol. 10, pp. 63189–63198, 2022, doi: 10.1109/ACCESS.2022.3182806.

[12] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. ICML*, vol. 2, 2015, pp. 1–30.

[13] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," 2016, *arXiv:1606.04080*.

[14] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[15] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 1199–1208, doi: 10.1109/CVPR.2018.00131.

[16] A. Srinivasan, A. Bharadwaj, M. Sathyan, and S. Natarajan, "Optimization of image embeddings for few shot learning," in *Proc. 10th Int. Conf. Pattern Recognit. Appl. Methods*, 2021, pp. 1–6.

[17] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[18] W. Jiang, K. Huang, J. Geng, and X. Deng, "Multi-scale metric learning for few-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1091–1102, Mar. 2021, doi: 10.1109/TCSVT.2020.2995754.

[19] X.-F. Xu, L. Zhang, C.-D. Duan, and Y. Lu, "Research on inception module incorporated Siamese convolutional neural networks to realize face recognition," *IEEE Access*, vol. 8, pp. 12168–12178, 2020, doi: 10.1109/ACCESS.2019.2963211.

[20] S. K. Roy, P. Kar, M. E. Paoletti, J. M. Haut, R. Pastor-Vargas, and A. Robles-Gomez, "SiCoDeF2net: Siamese convolution deconvolution feature fusion network for one-shot classification," *IEEE Access*, vol. 9, pp. 118419–118434, 2021, doi: 10.1109/access.2021.3107626.

[21] M. Zhang, H. Li, S. Pan, X. Chang, C. Zhou, Z. Ge, and S. Su, "One-shot neural architecture search: Maximising diversity to overcome catastrophic forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 2921–2935, Sep. 2021, doi: 10.1109/TPAMI.2020.3035351.

[22] A. Devi T. R., K. J. Sathick, A. A. A. Khan, and L. A. Raj, "A novel framework using zero shot learning technique for a non-factoid question answering system," *Int. J. Web-Based Learn. Teaching Technol.*, vol. 16, no. 6, pp. 1–13, Jun. 2021.

[23] J. Liu, C. Shi, D. Tu, Z. Shi, and Y. Liu, "Zero-shot image classification based on a learnable deep metric," *Sensors*, vol. 21, no. 9, p. 3241, May 2021, doi: 10.3390/s21093241.

[24] M. Chandrashekar and Y. Lee, "Class representative learning for zero-shot learning using purely visual data," *Social Netw. Comput. Sci.*, vol. 2, no. 4, p. 313, Jul. 2021, doi: 10.1007/s42979-021-00648-y.

[25] S. Zhou, Y. Zhou, and B. Liu, "Using Siamese capsule networks for remote sensing scene classification," *Remote Sens. Lett.*, vol. 11, no. 8, pp. 757–766, Aug. 2020.

[26] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR05)*, San Diego, CA, USA, Jun. 2005, pp. 539–546, doi: 10.1109/CVPR.2005.202.

[27] P. Sun, "The application of factor analysis in the study on cultural industry competitiveness evaluation index system," *Adv. Mater. Res.*, vols. 989–994, pp. 5132–5135, Jul. 2014.

[28] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 13728–13737, doi: 10.1109/CVPR46437.2021.01352.

[29] Y. Wu, J. Li, J. Wu, and J. Chang, "Siamese capsule networks with global and local features for text classification," *Neurocomputing*, vol. 390, pp. 88–98, May 2020, doi: 10.1016/j.neucom.2020.01.064.

[30] P. Demotte, K. Wijegunarathna, D. Meedeniya, and I. Perera, "Enhanced sentiment extraction architecture for social media content analysis using capsule networks," *Multimedia Tools Appl.*, vol. 82, no. 6, pp. 1–26, Sep. 2021, doi: 10.1007/s11042-021-11471-1.
[31] Q.-L. Zhang and Y.-B. Yang, "SA-Net: Shuffle attention for deep convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2235–2239.
[32] W. Rong, Z. Yang, and L. Leng, "Channel group-wise drop network with global and fine-grained-aware representation learning for palm recognition," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2022, pp. 1–9.
[33] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Artificial Neural Networks and Machine Learning—ICANN* (Lecture Notes in Computer Science), vol. 6791, T. Honkela, W. Duch, M. Girolami, and S. Kaski, Eds. Berlin, Germany: Springer, 2011, doi: 10.1007/978-3-642-21735-7_6.
[34] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3856–3866.
[35] S. Javadinia and A. Baniasadi, "PDR-CapsNet: An energy-efficient parallel approach to dynamic routing in capsule networks," 2023, *arXiv:2310.03212*.
[36] V. Mazzia, F. Salvetti, and M. Chiaberge, "Efficient-CapsNet: Capsule network with self-attention routing," *Sci. Rep.*, vol. 11, no. 1, p. 14634, Jul. 2021.
[37] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014, *arXiv:1411.7923*.
[38] 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," in *Advances in Face Detection and Facial Image Analysis*, M. Kawulok, M. Celebi, and B. Smolka, Eds. Cham, Switzerland: Springer, 2016.
[39] 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, pp. 5962–5979, Oct. 2022, doi: 10.1109/TPAMI.2021.3087709.
[40] M. Mitterreiter, M. Koch, J. Giesen, and S. Laue, "Why capsule neural networks do not scale: Challenging the dynamic parse-tree assumption," 2023, *arXiv:2301.01583*.
[41] A. T. Tran, T. Hassner, I. Masi, and G. Medioni, "Regressing robust and discriminative 3D morphable models with a very deep neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 21–26, doi: 10.1109/CVPR.2017.163.
[42] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 1701–1708, doi: 10.1109/CVPR.2014.220.
[43] G. Nam, H. Choi, J. Cho, and I.-J. Kim, "PSI-CNN: A pyramid-based scale-invariant CNN architecture for face recognition robust to various image resolutions," *Appl. Sci.*, vol. 8, no. 9, p. 1561, Sep. 2018, doi: 10.3390/APP8091561.

ZIHAN ZHANG was born in Hohhot, Inner Mongolia, China, in 1998. He received the bachelor's degree in computer science and technology from Inner Mongolia Agricultural University, China, in 2021, where he is currently pursuing the master's degree in computer application technology with the School of Computer and Information Engineering.

His research interest includes computer vision.

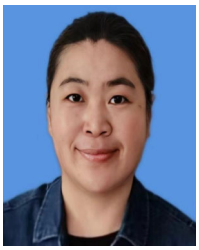
JING GAO received the master's degree in computer technology and the Ph.D. degree in computer software and theory from the Beijing University of Aeronautics and Astronautics (BUAA), in 2003 and 2009, respectively.

From 2011 to 2013, she was the Vice Dean of the School of Computer and Information Engineering, Inner Mongolia Agricultural University, where she was the Director of the Information and Network Centre, from 2014 to 2019. From 2019 to 2021, she was the Dean of the School of Computer and Information Engineering, Inner Mongolia Agricultural University, where she is currently a Ph.D. and Master's Supervisor with the School of Computer and Information Engineering. She has been working on big data intelligence and knowledge discovery, plant and animal phenotyping and histology big data analysis, and intelligent systems for agriculture and livestock. She has presided over or acted as the technical leader to complete the sub-topics of the National Natural Science Foundation of China and the National Nuclear High-Altitude Base Science and Technology Major Special Project. She has won the first prize and the second prize of Inner Mongolia Autonomous Region Scientific and Technological Progress. She was selected as "the first level of 321 talents project of the new century," and the Vice President of Inner Mongolia Autonomous Region Data Science and Big Data Society.

Prof. Gao is also a member of the Academic Committee of the Inner Mongolia Autonomous Region Discipline Inspection and Supervision Big Data Laboratory.

FENG XU received the bachelor's degree in computer science and technology from the Inner Mongolia University of Science and Technology, Baotou, China, in 2004, the master's degree in agricultural economy management from Inner Mongolia Agricultural University, Hohhot, China, in 2011, and the Ph.D. degree in agricultural information technical from Inner Mongolia Agricultural University, Inner Mongolia, China, in 2022.

He is currently a Lecturer with the Inner Mongolia Agricultural University. His research interests include machine learning and computer vision.



JUNJIE CHEN received the M.S. and Ph.D. degrees in computer application technology from Inner Mongolia University, in 2004 and 2020, respectively.

She is currently a Master's Tutor with the School of Computer and Information Engineering, Inner Mongolia Agricultural University. She has been engaged in the research and application of intelligent information processing

Dr. Chen was a Reviewer for the International Conference on Neural Information Processing (ICOIP), from 2019 to 2021, a top international conference on the CCF conference list, and The North American Chapter of the Association for Computational Linguistics (NAACL), in 2020. She is an invited Guest Editor of the Third International Conference on Computer Information Science and Artificial Intelligence (CISAI 2020) International Conference.