IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 19, 2022, accepted June 2, 2022, date of publication June 13, 2022, date of current version June 20, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3182806

Cow Face Recognition for a Small Sample Based on Siamese DB Capsule Network

FENG XU[1,2], JING GAO[1,2], AND XIN PAN[1]

[1]College of Computer and Information Engineering, Inner Mongolia Agricultural University, Hohhot 010011, China
[2]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, 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 Dairy cow face recognition using Neural Networks has several hurdles. For example, there are only a few instances of each individual. The positions and angles of the individuals in the image fluctuate considerably, the differences between individuals are not apparent, and the number of individuals that the network has not been trained on is enormous, etc. In this paper, an enhanced Siamese Neural Network is used to overcome these barriers. First, a combination of Dense Block (DB) and Capsule Network is employed as a feature extractor to keep the spatial information of features while expanding the feature extraction capabilities of the Convolutional Neural Network. Second, image pairings are processed through the Siamese Neural Network to obtain bivariate features. Finally, image recognition is achieved via the correlation analysis of bivariate features. We conduct comparison experiments with different networks on a small cow face dataset. The experimental results demonstrate that Siamese DB Capsule Network can learn abstract knowledge about distinct individuals and can be extended to unfamiliar cows for zero-shot learning.

INDEX TERMS Capsule network, cow face recognition, individual recognition, one-shot learning, Pearson correlation coefficient, Siamese neural network.

I. INTRODUCTION

Intelligent agriculture has been developed continuously and effectively through the promotion and application of artificial intelligence in agriculture. The basis of intelligent agriculture is large-scale farming. A prerequisite for large-scale farming is individual identification. A fundamental condition for monitoring animal safety and food production management is individual identification. The primary approach to identifying cows in large-scale farming is to deploy electronic tags based on RFID radio frequency technology [1], [2]. However, electronic tags have problems. For example, they are easy to lose and easy to tamper with. This can cause problems such as inaccurate identity recognition and individual identity replacement. A cow's face exhibits human-like facial features and rich textural elements such as patterns. Cows also have eyes, nose, mouth, and other parts of the face that are similar to humans. It provides unique biological characteristics for cow identification.

Convolutional Neural Network (CNN) has made many attempts in cow face recognition. Lu extracted features from 30,000 cow face images using Transfer Learning [3]. The redundancy property of the sparse representation dictionary was used to construct a sparse representation classification model. The accuracy of cow face recognition was compared for non-incremental and incremental cases. Gou *et al.* performed multiple cow face detection on 3,200 frames. Inception v2 was used as a predecessor network for Faster R-CNN, which was used to improve the model's accuracy. Non-Maximum Suppression was employed to optimize the visual scenes [4]. Yang *et al.* gathered 85,200 low-resolution pictures of the faces of 1,000 cows. For cattle facial identification, the super-resolution network was applied as a precursor network. While conducting image recognition, the image information was recovered [5]. Bisen used the K-means clustering algorithm to constantly deal with six prior frames and

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

then used YOLO3 to detect cow faces. 2,991 photos of cattle were used that contained multiple angles, such as the left face, the front face and the right face angles. The multi-angle cow face detection challenge was solved using the modified YOLO3 model [6]. All of the above studies were conducted with large-scale datasets. They all use Deep Feedforward Neural Networks that include convolutional computation for image recognition or target detection. The number of cows on large-scale farms ranges from 200 to 1,000. The number of cows on each farm is considerable. Obtaining a massive number of face photos for each cow is difficult. During data collection, the cow's head cannot be constrained. Cows are unable to cooperate like humans and take typical face pictures as required. The constant movement of the cow's head causes no uniformity in the data collected. It causes the cow faces in the photographs to have various orientations. In addition, the light, background, and other variables vary substantially [3]–[5]. Therefore, cow face data gathering is difficult, and identification is costly. This limits the application of CNN in the field of cow face recognition.

One-shot learning has been proposed to overcome the following problem: there are numerous categories, but the quantity of samples per category is minimal. LeCun *et al.* utilized the signatures that a bank had on file and handwritten signatures on checks as two inputs for a Neural Network [7]. The inputs were mapped to a new vector space to determine their similarity. Since then, one-shot learning has solved image classification problems, specifically facing recognition challenges. Taigman *et al.* used a 3D face model to align faces. The feature vectors of face images were extracted by a 9-layer neural network. The obtained feature vectors were used directly to predict whether two input face images belonged to the same person [8]. Hoffer *et al.* used three samples: one test sample, one sample from the same category as the test sample, and one sample from a different category [9]. These samples were fed through the same neural network to extract features independently. The model minimizes the distance between two similar features while increasing the distance between distinct categories. Lu *et al.* employed a context-aware module to improve the emphasis on the face by automatically ignoring the background of the image. Siamese Neural Network with center-classification was used to extract image features. The ''D-score'' was used to analyze the features [10]. Zong integrated depth features and RGB features. The features were recovered by Siamese Neural Network with privileged knowledge. This network was utilized to tackle the intra-class noise problem in face recognition [11]. Using the cyclical learning rate, Xu *et al.* integrated the Inception Module into Siamese Neural Network. This strategy improved the speed and accuracy of face recognition training. It was also suitable for small-size datasets [12]. When one-shot learning is deployed, each category has only a small number of labeled samples [13], [14]. When the categories are not annotated with samples, one-shot learning becomes zero-shot learning. Zero-shot learning performs Transfer Learning or even direct prediction through the commonality of image features and class attributes by utilizing already trained networks in selected semantic spaces [15]–[18].

Due to the lack of cooperation from cows, it is difficult to collect large amounts of data and verify the identity of individuals. This study employs one-shot learning to tackle the small sample size problem in individual cow face identification. For feature extraction, a Capsule Network with Dense Block module is proposed. Bivariate characteristics are obtained through Siamese Neural Network. The relationships of these characteristics are analyzed for cow face recognition.

The main contributions of this paper are the following:

- The Capsule Network is improved by incorporating the Dense Block module of DenseNet. This improves the ability of the classical Capsule Network to extract convolutional features and encode the spatial information of the object.
- A model of bivariate feature correlation analysis is utilized. Instead of the distance measure used in the classical Siamese neural network, the Pearson correlation coefficient is used. Thus, the convergence of the loss function is accelerated.
- The proposed network solves the problem of cow face recognition with small samples. Meanwhile, it is robust to unfamiliar cow faces and can be used for zero-shot learning.

The organizational structure of this paper is as follows. Section II introduces related techniques and algorithmic formulations. Section III describes the proposed Siamese Neural Network structure based on the enhanced Capsule Network. Section IV presents the experimental conditions. Section V discusses the experimental results. Section VI summarizes the conclusions.

II. RELATED TECHNOLOGIES

A. CNN AND DENSENET

In a classical CNN, an image is taken as input. It passes through the L-layer of the neural network, where the input of the layer i is denoted as X_{i-1}, and the nonlinear transformation is represented as $\mathrm{H_i}(*)$. $\mathrm{H_i}(*)$ is the accumulation of various functional operations, such as convolution, nonlinear activation, pooling, etc., used to obtain the output of layer I, denoted as X_i. The output features of layer i are obtained using (1).

$$X_i = \mathrm{H_i}\left(X_{i-1}\right) \tag{1}$$

DenseNet uses (2) to obtain layer i's output features [19]; $[X_0, X_1, \ldots, X_{i-1}]$ represents the merging of the output features of layers 0 to *i-1*. This tight connection exists only in each Dense Block. Dense Blocks connect all the layers directly and make more efficient use of feature information. At the same time, they ensure maximum information transfer between layers in the network.

$$X_i = \mathrm{H_i}\left(\left[X_0, X_1, \ldots, X_{i-1}\right]\right) \tag{2}$$

B. CAPSULE NETWORK

The classical Capsule Network consists of a shallow neural network [20], [21]. The first layer is a regular convolutional layer. The second layer collects 6 × 6 × 8 × 32-dimensional features following convolutional processes and defines the 8-dimensional vector of these features as a capsule. The third layer performs the convolution operation in the capsule and then obtains 10 capsules, each consisting of a 16-dimensional vector. The Capsule Network uses (3) as the nonlinear activation function, which is called Squash. v_j is the output vector of Capsule j, and s_j is the vector-weighted sum of all Capsule outputs from the previous layer to the current layer of Capsule j. It means that s_j is the input vector of Capsule j.

$$v_j = \frac{||\mathrm{s_j}||^2}{1+||\mathrm{s_j}||^2} \cdot \frac{s_j}{||s_j||} \tag{3}$$

The first part of (3) maps the activation vector between 0 and 1, with longer vectors nearer to 1. The second part ensures that the direction of the activation vector is not altered. The Capsule Network instantiates parameters by encapsulating convolutional neurons into neuronal feature vectors representing specific entity types. The neuronal feature vectors are called Capsule. Each Capsule contains spatial information, such as the position, texture, orientation, and probability of occurrence of a particular entity. There have been various attempts to improve the capsule network's characteristics by strengthening capsule extraction features and dynamic routing algorithms [22], [23].

C. SIAMESE NEURAL NETWORK

In Siamese Neural Network (SNN), two images are simultaneously feds into an embedding function $\mathrm{f}_\theta\ (*)$ consisting of multiple convolutional layers for feature extraction [24]. The Euclidean distance between the characteristics of the two images is measured [25]–[27]. The distance is converted into a probability and then classified using (4), where σ is the sigmoid activation function and $\propto$ represents the other parameters learned by the model during training. The probabilities are used to determine whether the two images belong to the same category.

$$p\left(\mathrm{x_i}, \mathrm{x_j}\right) = \sigma(\propto |\mathrm{f}_\theta(\mathrm{x_i}) - \mathrm{f}_\theta(\mathrm{x_j})|) \tag{4}$$

The SNN uses two duplicate networks with distinct images as inputs. During the computations, parameters are exchanged between networks. This network architecture executes the same feature extraction process for diverse images, providing equivalent output features.

The SNN generally utilizes Contrastive Loss function [28], [29], and (5) is its mathematical formulation.

$$L = (1-\mathrm{Y})\frac{1}{2}(d)^2 + (Y)\frac{1}{2}\{max(0,m-d)\}^2 \tag{5}$$

In (5), d is the distance between two features, and Y is the label of the image pair. When Y = 1, the two images belong to the same category, and L minimizes the distance between the two features. When Y = 0, the two images belong to different varieties, and if the distance between the two features is less than m, then the distance between the two features is increased to m.

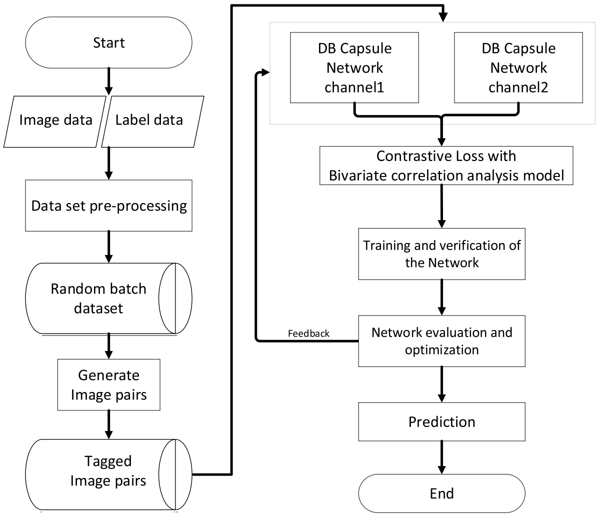

Start
Image data
Label data
Data set pre-processing
Random batch dataset
Generate Image pairs
Tagged Image pairs
DB Capsule Network channel1
DB Capsule Network channel2
Contrastive Loss with Bivariate correlation analysis model
Training and verification of the Network
Feedback
Network evaluation and optimization
Prediction
End

FIGURE 1. Flowchart of the siamese DB capsule network algorithm.

D. BIVARIATE CORRELATION ANALYSIS

The dependent relationships between variables can be analyzed with methods such as distance analysis and correlation analysis. Distance analysis commonly employs the Euclidean distance, cosine distance, and Hamming distance. Correlation analysis typically utilizes the Pearson correlation coefficient and Kendall's tau coefficient. The Pearson correlation coefficient takes variance as an assumption. It is used to characterize the degree of linear correlation between variables. The mathematical expression for the Pearson correlation coefficient is (6) [30].

$$\mathrm{COR}\left(\mathrm{x_i}, \mathrm{x_j}\right) = \frac{\sum_{1}^{n}(\mathrm{x_i} - \bar{\mathrm{x}}_\mathrm{i})(\mathrm{x_j} - \bar{\mathrm{x}}_\mathrm{j})}{\sqrt{\sum_{1}^{n}(\mathrm{x_i} - \bar{\mathrm{x}}_\mathrm{i})^2(\mathrm{x_j} - \bar{\mathrm{x}}_\mathrm{j})^2}} \tag{6}$$

$\mathrm{x_i}$, $\mathrm{x_j}$ are the actual variables; $\bar{\mathrm{x}}_\mathrm{i}$, $\bar{\mathrm{x}}_\mathrm{j}$ are the means of the corresponding variables; and $\mathrm{COR}\left(\mathrm{x_i}, \mathrm{x_j}\right)$ is the Pearson correlation coefficient of $\mathrm{x_i}, \mathrm{x_j}$. The value of $\mathrm{COR}\left(\mathrm{x_i}, \mathrm{x_j}\right)$ is between −1 and 1. A value of 1 shows a perfectly positive correlation between the two random variables. A value of −1 indicates an entirely negative correlation between the two random variables. A value of 0 indicates a linear correlation between the two random variables.

III. PROPOSED METHOD

A. SIAMESE DB CAPSULE NETWORK

In this paper, we propose a Siamese DB Capsule Network (SDBCN). SDBCN uses Dense Block and Capsule Network for feature extraction. The features of the image pairs are extracted by the Siamese Network architecture. Bivariate feature correlation analysis is deployed to determine the categories of two input images, thus solving the image recognition problem for small datasets. Figure 1 illustrates the flowchart of the SDBCN algorithm, which takes image pairs and labels as input. The label is equal to 1 when the two images belong to the same category and 0 when the two images belong to different classes. For cow identification, we can first generate a database of small samples of individual cow faces. Subsequently, new individual photos can be taken through SDBCN and compared with the database of individual cows. Therefore, the identity of each cow can be confirmed.

There are two deep neural sub-networks in SDBCN to extract features from two images simultaneously. SDBCN uses Capsule Network fused with Dense Block as a feature extractor. The Capsule Network fused with Dense Block is abbreviated as the DB Capsule Network. The DB Capsule Network consists of an input layer, a convolutional layer, a Dense Block layer, a Primary Caps layer, a CFace Caps layer, and an output layer. The weights are shared between the sub-networks so that the same features can be obtained for the same image from both sub-networks. Each layer is followed by batch normalization and dropout to reduce overfitting.

The DB Capsule Network extracts the features of image pairs in the form of two feature variables for correlation analysis. SDBCN uses the Pearson correlation coefficient as a bivariate correlation analysis metric. When the two images belong to the same category, the value of the Pearson correlation coefficient is nearer to 1. When the two images do not belong to the same category, the value of the Pearson correlation coefficient is closer to −1. SDBCN uses the Pearson Contrastive Loss, which is defined by (7).

$$L = \frac{1}{N}\sum_{n=1}^{N}[(1-Y)\times cor^2 + Y \times \max(margin - cor, 0)^2] \tag{7}$$

cor in (7) is the Pearson correlation coefficient, and Y represents the label of the image pair. When Y = 1, the two images belong to the same category, and L maximizes the correlation between the two image features. When Y = 0, the two images belong to different categories, and L minimizes the correlation between the two image features. The margin indicates the set threshold value. Although the loss functions defined by (7) and (5) look similar, the methods they use to calculate whether two images belong to the same category are completely distinct. When Y = 1, the more similar the two images are, the closer the value of similarity calculated by (7) is to 1, while the distance value calculated by (5) is closer to 0.

B. CONSTRUCTING BIVARIATE DATASETS

SDBCN requires two images to be fed into the DB Capsule Network simultaneously. Therefore, it is necessary to construct a bivariate dataset by randomly selecting two images at a time from the dataset. The pseudo-code of the algorithm for constructing the bivariate dataset is shown in Algorithm 1. The image containing the image category label is taken as the input, and the output is an image pair and a label. The algorithm constructs image pairs cyclically as needed. The label is represented by a randomly generated 0 or 1. Randomly generated labels ensure that the number of positive and negative image pairs remains consistent. When the image pair's label is 1, the two images belong to the same category, and when the label is 0, the two images belong to different categories. Specifically, an image img_0 and its category are randomly selected from the dataset X. When the image pair label is 1, an image img_1 is randomly filtered from the dataset until img_0 and img_1 belong to the same category. When the label is 0, a random image img_1 is selected from the dataset until img_0 and img_1 belong to different categories. Finally, img_0, img_1, and the label are added to the image pair dataset, which is the bivariate dataset.

Algorithm 1 Pseudocode for building a bivariate dataset

Algorithm I: Construct a bivariate dataset
Input: A batch of images X
Output: img_couple[]

```
Start:
Define empty lists img_couple[]
for all i = 1, 2, ..., n do
    img_couple_label = Random.randint(0, 1)
    img_0, img_0_label = Random.choice(X)
    if img_couple_label == 1
        While True
            img_1, img_1_label = Random.choice(X)
            if img_0_label == img_1_label
                break
    else
        While True
            img_1, img_1_label = Random.choice(X)
            if img_0_label != img_1_label
                break
    img_couple.add(img_0, img_1, img_couple_label)
end for
End.
```

C. DB CAPSULE NETWORK

In a classical Capsule Network, the first convolutional layer uses 9 × 9 kernels to extract features, but the large convolutional kernels are not capable of extracting deep convolutional features. Stacking the convolutional layers using small 3 × 3 kernels can improve the performance of feature extraction while speeding up the computations [27]. Therefore, the Dense Block module uses six layers of 3 × 3 and 1 × 1 convolutional kernels stacked interactively to extract deep features in multiple dimensions and improve the learning capability of the network. The 1 × 1 convolution process, for example, can fuse the features of each channel while also reducing the computational cost by adjusting the dimensionality. DenseNet's performance can be improved by using a lesser growth rate. After performing some experiments, we found that a growth rate of 32 works best.

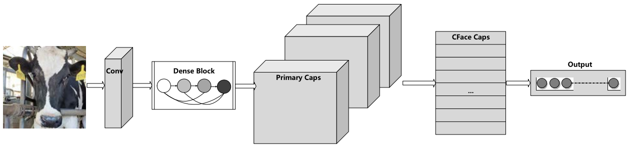


FIGURE 2. The structure of the DB capsule network.

SDBCN uses the DB Capsule Network as a facial feature extractor. The structure of the DB Capsule Network is shown in Fig. 2. It comprises six main parts: Input Layer, Convolutional Layer, Dense Block, Primary Caps, CFace Caps, and Output Layer. An image pair is fed into the network. First, 128 layers of shallow features are extracted by the convolutional network with two 3 × 3 kernels. Then, the features are extracted using Dense Blocks to get the 224-dimensional high-level features. The image features are passed through the Primary Cap to get 2,048 capsule cells, where each capsule cell is an 8-dimensional vector. After that, the CFace Caps layer is obtained using the Squash function. The CFace Caps layer uses a 3-time dynamic routing algorithm and a Softmax function to adjust the number and dimensionality of the capsules between two layers. Then, the final 16 capsules are obtained, where each capsule is a 10-dimensional vector. Each capsule represents the features of a class of visual entities, and their probabilities are predicted by the length of each capsule cell. A nonlinear activation function, Squash, is used between the two capsule layers; it is defined by (8), where s_j is the sum of the weights of all Primary Capsule outputs j, v_j is the value after squashing, and m is a constant. The capsules are flattened into a 160-dimensional vector. In the next step, the Pearson correlation coefficient is calculated by taking the output values from the two subnetworks.

$$v_j = \frac{||\mathrm{s_j}||^2}{\mathrm{m}+||\mathrm{s_j}||^2} \cdot \frac{s_j}{||s_j||} \tag{8}$$

D. BIVARIATE CORRELATION ANALYSIS MODEL

The classical Capsule Network determines whether two features belong to the same class by calculating the distance between them (e.g., the Euclidean distance) [24]. The intra-class samples correspond to features that are closer together, and the inter-class samples correspond to features that are farther apart. However, the capsule unit uses vectors rather than scalars to represent features. The distance between two capsule vectors can not be suitably measured using distance measures such as the Euclidean distance or cosine distance.

The DB Capsule Network obtains two 16 × 10 feature vectors through the Siamese Network. Each cow's face is distinct in each picture regarding its position, angle, and size. As a result, the feature vectors extracted by SDBCN for each image have different meanings in the vector space. The Pearson correlation coefficient does not change due to changes in the location and scale of the two variables; it is invariant to changes [26]. Therefore, a bivariate correlation analysis model is constructed using (4). Two 16 × 10 feature vectors are obtained through the DB Capsule Network. It calculates the bivariate Pearson correlation coefficient value. The closer this value is to 1, the more likely it is that the two images are in the same category.

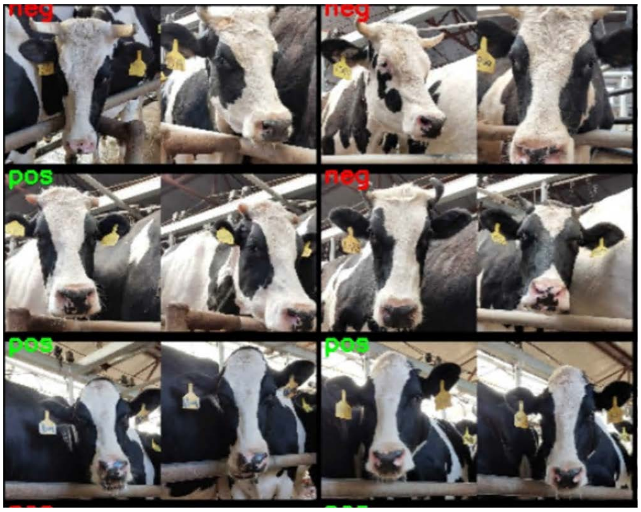


FIGURE 3. Examples of Image Pairs. When the image pair's label is 0, the two images belong to different categories (indicated as neg). When the label is 1, the two images belong to the same category (indicated as pos).

IV. EXPERIMENTS

A. DATA SETS AND PREPROCESSING

The data were taken from a contemporary farm with a breeding population of more than 300 cows. Face photos of 63 cows were obtained in a natural environment. The photographs contain various poses, such as glancing up, looking down, facing forward, facing sideways, chewing, etc. The 63 cows are numbered from 0 to 62, and each cow is considered one category. For one-shot learning, 15 RGB images are randomly selected for each category. A total of 945 cow face images constitute a small sample dataset.

For the sake of the experiment, all images were resized to 50 × 50. Fifty categories were randomly selected from the dataset for training, while the remaining 13 categories were used for testing. Ten photographs from each category for training were randomly picked to make the training dataset, and the remaining five images make up the validation dataset. There are no duplicate images in the training and validation datasets. Algorithm 1 was used to generate 300 image pairs from the training dataset, validation dataset, and test dataset, respectively. A small example of cow face recognition is performed. Sample image pairs are given in Fig. 3.

The LFW dataset [31], which is mainly collected from the Internet rather than the laboratory, contains more than 13,000 face images, each of which is identified by the name of the corresponding person, of which 1,680 people correspond to more than one image, i.e., about 1,680 people contain more than two faces.

B. EXPERIMENT

This experiment was done using a computer with a Tesla P40 23 GB GPU and the Pytorch 1.6.0 deep learning framework development platform on the Centos 7.9 operating system.

The experiments use accuracy, F1-score, and loss as evaluation metrics. The F1-score is the comprehensive evaluation of recall and precision, calculated according to (9). The higher the F1 value, the more robust the classification model is.

$$F1 = 2 \times Recall \times Precision \Big/ (Recall + Precision) \quad (9)$$

SDBCN is presented in Section III. C. SDBCN employs the Adam optimizer with a learning rate of 1e-4; there are 200 iterations of training per experiment, and a batch size of 32 is used. After parameter comparison experiments, it is clear that the network converges best with m = 0.5 in the Squash function and margin = 1.7 in the Contrastive Loss function.

In this paper, four sets of experiments are designed for cow face recognition on a small sample dataset. In the first set of studies, SDBCN is compared with other Siamese Networks under the same conditions, i.e., they all employ distance measures. The second set of comparison tests utilizes the distance metric and Pearson correlation coefficient as the input values of the Contrastive Loss function, respectively. The third series of tests was conducted, based on the second set of comparison tests, by adjusting the amount and size of the images that make up the training data. The final experiment applies SDBCN to zero-shot learning trials on a dataset that has not been trained.

V. RESULTS AND DISCUSSION

In this paper, comparison experiments are conducted using SDBCN and several deep convolutional networks. The experiments show from both a network architecture perspective and a feature metric perspective. To ensure the fairness of the comparison experiments, all of these networks use Siamese Network structures. In the following subsections, we present the details of each network separately.

A. COMPARISON OF DIFFERENT SIAMESE NETWORKS

Comparative experiments were conducted using five different networks under the same conditions, where all distance measurements were used. These networks all use a Siamese Network structure. To ensure fairness in the comparison trials, as much as possible, the same parameter settings are used for each network.

1. The Deep Neural Network used in [25] is denoted as SNN; the image size is 105 × 105. In this experiment, SNN is utilized as the baseline. SNN uses the Euclidean distance for image recognition, and it uses the Contrastive Loss function.

2. SNN_C is a network derived from SNN that uses the cosine distance instead of the Euclidean distance. It also uses the Contrastive Loss function.

TABLE 1. Experimental results and parameter sizes of the different siamese networks.

Models	Accuracy (%)	F1-score（%）	Loss	Parameter (M)
SNN	71.00	69.47	--	38.96
SNN_C	85.33	86.98	0.7762	38.96
SDN	88.00	89.77	**0.6796**	11.16
SCN	87.33	88.89	0.7886	20.09
SDBCN_C	**91.67**	**92.45**	0.7272	**10.77**

3. The classical DenseNet also uses the cosine distance and the Contrastive Loss function. It is denoted as SDN. The image size is 224 × 224.

4. The classical Siamese Capsule Network (SCN) also uses the cosine distance and Contrastive Loss function.

5. This experiment compares the feature extraction capabilities of SDBCN with those of the four networks listed above. As a result, the distance metric used by these other networks to calculate the distance between two features is likewise used by SDBCN. After testing, SDBCN fails to converge when the Euclidean distance is employed. Therefore, in this experiment, SDBCN uses the cosine distance instead of the Euclidean distance; this version of SDBCN is denoted as SDBCN_C.

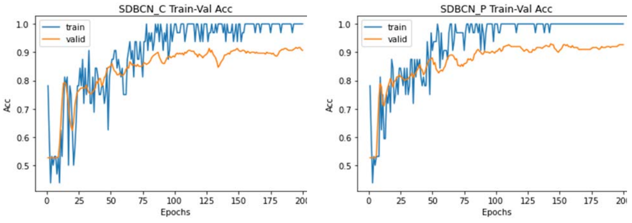
The experimental results and the number of parameters of different Siamese Networks are compared in Table 1. When comparing SDBCN_C to other Siamese Networks, it is clear that SDBCN_C outperforms them. SDN and SCN have similar accuracy rates, with SDN being 0.67% more accurate. SDBCN_C has a 91.67% accuracy rate, which is 3.67% higher than SDN. When compared to SNN with the same network configuration, SNN_C has a substantially higher accuracy rate. This Siamese Network maintains a steady F1-score and accuracy ranking. SNN uses the Euclidean distance, whereas the other Siamese Networks employ the cosine distance, and all of those networks employ the Contrastive Loss function. SDBCN_C has a more considerable loss function value than SDN by 0.0476. SNN_C has 3.62 times the number of parameters as SDBCN_C, while SCN has 1.87 times the number of parameters as SDBCN_C. SDBCN_C has approximately 0.39 million fewer parameters than SDN.

The efficiency of the network changes substantially when different deep convolutional neural networks are utilized as feature extractors in the same Siamese Network structure. Fig. 3 depicts the cow face data used in this investigation, including a complex natural background. While SDN and SCN extract features using unique deep convolutional neural networks, their performances are comparable. Compared to the other networks in this experiment, SDBCN_C has the best overall performance. It is primarily due to SDBCN_C's combination of Dense Block and Capsule Network, which boosts the network's ability to extract features.

The Dense Block concentrates all the previous layers' extracted features while extracting features from each layer.

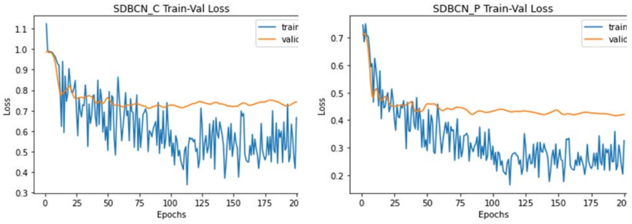
TABLE 2. Experimental results from the comparison of SCN and SDBCN.

Models	Accuracy (%)	F1-score （%）	Loss	Distance / Correlation coefficient
SCN_C	87.33	88.89	0.7886	0.3999
SCN_P	88.67	89.57	0.7632	**0.3810**
SDBCN_C	91.67	92.45	0.7272	0.3999
SDBCN_P	**93.00**	**93.54**	**0.4207**	0.4172



(a) (b)

FIGURE 4. Accuracy of SDBCN_C and SDBCN_P. (a) Accuracy of SDBCN_C on the training and validation datasets; (b) accuracy of SDBCN_P on the training and validation datasets.



(a) (b)

FIGURE 5. Loss values of SDBCN_C and SDBCN_P. (a) Loss values of SDBCN_C for the training and validation datasets; (b) loss values of SDBCN_P for the training and validation datasets.

This strategy improves the efficiency of feature reuse and also fuses the low-level features with the high-level features to improve the feature representation. In SDBCN_C, however, it is observed that stacking more Dense Block layers does not improve network performance. It implies that image-rich characteristics may be recovered by utilizing the proper quantity of Dense Block layers. A Capsule is a vehicle with many neurons. Each Capsule identifies a visual entity with a constrained observation condition and instantiates it as a spatial vector. Cows' facial features are readily visible, which facilitates feature extraction. SDBCN_C first utilizes Dense Block to extract low-level characteristics such as the color, texture, and edges. Then, it uses a Capsule to obtain instantiated spatial vector information. Therefore, SDBCN_C extracts richer features and expresses better than other Siamese Networks.

Although both SNN and SCN are shallow networks with up to 6 layers, the number of SNN parameters is 38.96M; SNN is 3.49 times bigger than SDN. It is since the use of fully connected layers in SNN considerably expands the number of parameters. The number of parameters in SDN and the number of parameters in SDBCN_C are nearer; these networks are markedly smaller than the other networks. This is due to the number of output channels being modified in each Dense Block, causing the number of parameters in the network to be limited.

B. COMPARISON OF THE USE OF DIFFERENT FEATURE METRICS IN SDBCN

In this experiment, the network performance is investigated when two different feature measurements are employed. SCN and SDBCN obtain similar spatial feature vectors. As a result, this experiment only compares SCN and SDBCN. SCN and SDBCN still preserve the image size of experiment A.

1.The SCN_C designation still refers to the classic Siamese Capsule Network (SCN), which uses the cosine distance.

2. SCN_P denotes the SCN created using the Pearson correlation coefficient to describe the two feature relations.

3. The version of SDBCN employing the cosine distance is still denoted as SDBCN_C.

4. The version of SDBCN utilizing the Pearson correlation coefficient to describe the two feature relations is denoted as SDBCN_P.

Since both feature metrics are non-parametric, the number of parameters for SDBCN_P and SDBCN_C is the same.

Table 2 compares the results of the SCN and SDBCN trials. SDBCN_P achieves an accuracy of 93.00%, which is 1.33% higher than SDBCN_C. The F1-score of SDBCN_P is 93.54%, which is 1.09 points higher than that of SDBCN_C. SDBCN_P performs better than SDBCN_C, SCN_C, and SCN_P.

The conventional Siamese Network utilizes the distance between two feature vectors to express the similarity of an image pair. SDBCN_P applies the Pearson correlation coefficient to describe the similarity of two feature vectors. When the Pearson correlation coefficient is close to 1, the image pairs have a high probability of belonging to the same category.

Fig. 4 demonstrates the accuracy of SDBCN_C and SDBCN_P in the training and validation experiments. It can be found that the overall performance of the two networks is consistent. SDBCN_P tends to stabilize at the 100th epoch, whereas SDBCN_C tends to stabilize at the 150th epoch. Both networks show the same trend in the accuracy in training and validation experiments, and the fit is robust. Fig. 5 shows the loss values of SDBCN_C and SDBCN_P in the training-validation experiments, and the overall performance of the two networks is likewise consistent. The trends of the accuracy and loss values of SDBCN_P are smoother than those of SDBCN_C. Thus, SDBCN_P better fits the data.

SDBCN extracts the spatial feature vectors of 16 groups of Capsules through the CFace Capsule layer. Each spatial feature vector portrays the visible entity, i.e., the instantiated spatial feature vector. The two image pairs at the bottom of Fig. 2 belong to the same class, but the cow's face has a different location and pose in each image. The states of the eyes, nose, and mouth of the cow's face are similarly varied. All of these facial features are retrieved by SDBCN as spatial

TABLE 3. Experimental results of SCN and SDBCN.

Models	Input scale (Pixels)	Number of Image Pairs	Accuracy (%)	F1-score （%）	Loss	Distance / Correlation coefficient
SCN_C	50×50	300	87.33	88.89	0.7886	0.3999
	50×50	900	87.67	88.33	0.6923	0.4031
	128×128	300	88.67	89.31	0.8078	0.3894
	128×128	900	90.23	91.67	0.7878	0.4325
SCN_P	50×50	300	88.67	89.57	0.7632	0.3810
	50×50	900	90.56	91.23	0.6573	0.3622
	128×128	300	89.33	89.98	0.8098	0.3878
	128×128	900	91.33	92.67	0.6596	0.4266
SDBCN_C	50×50	300	91.67	92.45	0.7272	0.3999
	50×50	900	92.00	91.95	0.5956	0.4388
	128×128	300	92.33	93.01	0.279	0.4185
	128×128	900	92.89	92.56	0.2310	0.4515
SDBCN_P (Proposed Network)	50×50	300	93.00	93.54	0.4207	0.4172
	50×50	900	93.67	94.69	0.1262	0.4124
	128×128	300	94.00	95.21	0.2640	0.4193
	128×128	900	94.56	95.57	0.1931	0.4177

feature vectors. For example, the position and movement of the mouth are different in each image. In addition, there are many poses: the cow may open its mouth, chew, stick its tongue out, etc. Hence, 16 Capsule groups are obtained using SDBCN. For example, the mouth feature is represented by the 7th capsule in the first image feature and by the 12th capsule in the second image feature. The identical visual components are positioned differently in the image pair's spatial feature vector. As a result, the use of distance to quantify the similarity of image pairs is constrained. The Pearson correlation coefficient in subsection II.D is invariant and does not change due to the change in position. The Pearson correlation coefficient is more applicable to SDBCN. According to the comparison mentioned above, SDBCN_P is the optimal version of SDBCN.

C. STABILITY EXPERIMENTS FOR SDBCN

The experiment in V. B. was conducted on a dataset of 50 × 50 pixels and 300 image pairs. Therefore, the experiment described in this section is meant to discover the effects of both increasing the image size and increasing the number of image pairs. The dataset is resized to 50 × 50 pixels with 900 image pairs; 128 × 128 pixels with 300 image pairs; and 128 × 128 pixels with 900 image pairs, respectively. Algorithm 1 was used to generate the image pairings, and no overfitting occurs. This experiment uses the four networks in Experiment V. B. Table 3 summarizes the experimental findings.

As shown in Table 3, when the number of 50 × 50 image pairs increases from 300 to 900, the accuracy of all four networks improves slightly. On the 50 × 50, 900-image-pair dataset, the accuracy of SDBCN_P, the version of SDBCN suggested by this study, improves by 0.67%. However, the size of the dataset was increased by a factor of three. The accuracy of the networks for the 128 × 128 dataset follows a similar trend. The accuracy of SCN and SDBCN increases with the amount of training data, but the difference is modest. For comparison, when the dimensions in pixels are increased from 50 × 50 to 128 × 128, the output capsule of the Primary Caps layer grows by 12.25 times. As a result, the number of SDBCN parameters is 40.26M, which increases by 3.74%. However, the accuracy rises only by approximately 1%.

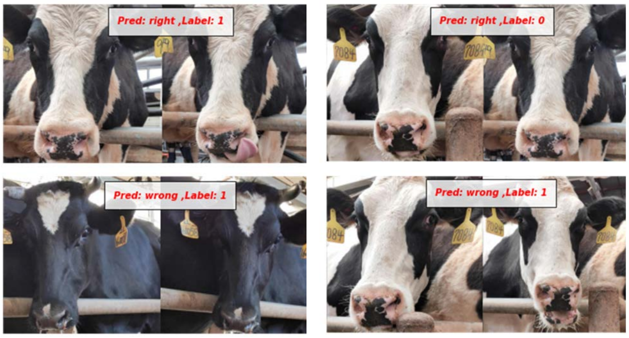


FIGURE 6. SDBCN test results on the unfamiliar dataset.

TABLE 4. Verification accuracies of different methodologies on the LFW dataset.

Models	Accuracy (%)
DeepFace[32]	95.92
DeepFace-Siamese[32]	96.17
3DMM[33]	92.35
PSI[34]	98.87
HSN[35]	98.95
SDBCN(Proposed Network)	96.87

All four networks in this experiment use Capsule units to represent cow face features, and the experiments demonstrate that the Capsule units can express image features stably. The DB Capsule Network in SDBCN has a better feature representation ability. The Pearson correlation coefficient fluctuates only slightly as the size of the dataset increases. SDBCN performs more stably on cow face recognition than other networks, regardless of the size of the dataset or the size of the images. In particular, it performs better on small datasets.

Currently, most face verification methods achieve high performance with vast amounts of training data. From the results in Table 4, the following points are noted. Our results are higher (+0.95%) than DeepFace[32], higher(+0.70%) than DeepFace-Siamese[32]. However, the accuracy of SDBCN is 2% lower than that of PSI[34], and HSN[35]. Since SDBCN is dedicated to cow face recognition rather than human face recognition such as the LFW dataset, this has been able to demonstrate its performance.

D. SDBCN FOR ZERO-SHOT LEARNING

The training procedure of SDBCN does not use the test dataset. Thus, the test dataset can be used as an unfamiliar dataset for SDBCN. This experiment employs the optimal, trained SDBCN. The network was tested using 300 randomly generated image pairs from the test dataset. The accuracy is 88.33%, and the F1-score is 88.89% when testing is performed with five cows. When testing with 13 cows, the accuracy is 86.27%, and the F1-score is 88.02%.

Fig. 6 displays the performance of SDBCN on the test set. The label indicates the tag of the image pair. If the label = 1, the images in the image pair belong to the same category, and if the label = 0, the images in the image pair belong to different categories. The correctness of the prediction is denoted by pred. If pred = right, the predicted label is different from the actual label. If pred = wrong, the predicted label is discordant with the actual label. For SDBCN, the test dataset is unfamiliar. The training dataset and the test dataset contain different individuals. The datasets both have pictures of cows' faces. Each image comprises comparable visual entities. Therefore, SDBCN can extract spatial feature vectors from images. The results reveal that SDBCN is robust to unknown cow face data. SDBCN can be utilized for zero-shot learning for cow facial recognition.

VI. CONCLUSION

Cow face recognition has numerous issues, such as a significant number of individuals and a small number of samples for each individual. In this paper, SDBCN is proposed to address the cow face recognition challenge. As a feature extractor, SDBCN combines Dense Block and Capsule Network. Through the Siamese Network structure, bivariate characteristics are then created. The correlation of the bivariate characteristics is determined to perform image recognition. The experiments are conducted on a small sample dataset of cow faces. The dataset contains 63 cows with 15 photos per individual, for 945 images. For experiments, SDBCN is compared with the classical Siamese Network and some of its variations. SDBCN can achieve an accuracy of 93%. It shows a significant improvement in recognition accuracy and robustness compared with other Siamese Networks. At the same time, SDBCN can classify individuals that have not been trained. SDBCN superimposes multiple layers of convolutional features while extracting spatial vectors. This feature extraction encodes the spatial information of visual entities while decreasing the influence of noisy features. To deal with the changes in spatial vector feature positions due to the different poses of cows, SDBCN uses correlation analysis instead of distance metrics to achieve positive results. SDBCN is a novel research idea for small-sample 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] C. Lu, "Research on pattern recognition and application on cattle face recognition based on deep learning and sparse representation," M.S. thesis, School Comput. Sci. Eng., North Minzu Univ., Yinchuan, China, 2018.

[4] X. T. Gou, W. Huang, and Q. F. Liu, "Gesture estimation of cattle face based on cascade structure," *Comput. Syst. Appl.*, vol. 28, no. 7, pp. 240–245, 2019.

[5] Z. Yang, H. Xiong, X. Chen, H. Liu, Y. Kuang, and Y. Gao, "Dairy cow tiny face recognition based on convolutional neural networks," in *Biometric Recognition*. Cham, Switzerland: Springer, 2019, pp. 216–222, doi: 10.1007/978-3-030-31456-9_24.

[6] W. Bisen, "Research on cow face recognition technology based on convolutional neural network," M.S. thesis, School Softw., Yunnan Univ., Kunming, China, 2020.

[7] J. Bromley, J. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a 'Siamese' time delay neural network," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 669–688, 1993, doi: 10.1142/S0218001493000339.

[8] 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.*, Jun. 2014, pp. 1701–1708, doi: 10.1109/CVPR.2014.220.

[9] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition*. Cham, Switzerland: Springer, 2015, pp. 84–92, doi: 10.1007/978-3-319-24261-3_7.

[10] T. Lu, Q. Zhou, W. Fang, and Y. Zhang, "Discriminative metric learning for face verification using enhanced Siamese neural network," *Multimedia Tools Appl.*, vol. 80, no. 6, pp. 8563–8580, Mar. 2021, doi: 10.1007/s11042-020-09784-8.

[11] R. Zong, "On privileged information driven robust face verification: A Siamese convolutional neural network approach," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 5874–5878, doi: 10.1109/bigdata50022.2020.9378074.

[12] 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.

[13] S. K. Roy, P. Kar, M. E. Paoletti, J. M. Haut, R. Pastor-Vargas, and A. Robles-Gomez, "SiCoDeF^2 Net: Siamese convolution deconvolution feature fusion network for one-shot classification," *IEEE Access*, vol. 9, pp. 118419–118434, 2021, doi: 10.1109/access.2021.3107626.

[14] 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.

[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.*, Jun. 2018, pp. 1199–1208, doi: 10.1109/CVPR.2018.00131.

[16] 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, Nov. 2021, doi: 10.4018/IJWLTT.20211101.OA12.

[17] 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.

[18] M. Chandrashekar and Y. Lee, "Class representative learning for zero-shot learning using purely visual data," *Social Netw. Comput. Sci.*, vol. 2, no. 4, pp. 1–21, Jul. 2021, doi: 10.1007/s42979-021-00648-y.

[19] R. Deng and S. Liu, "Relative depth order estimation using multi-scale densely connected convolutional networks," *IEEE Access*, vol. 7, pp. 38630–38643, 2019, doi: 10.1109/ACCESS.2019.2903354.

[20] S. K. Sahu, P. Kumar, and A. P. Singh, "Dynamic routing using inter capsule routing protocol between capsules," in *Proc. UKSim-AMSS 20th Int. Conf. Comput. Model. Simul. (UKSim)*, Mar. 2018, pp. 1–5, doi: 10.1109/UKSim.2018.00012.

[21] S. Sabour, N. Frosst, and G. Hinton, "Matrix capsules with EM routing," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, vol. 115, 2018, pp. 1–15.

[22] C. Pan and S. Velipasalar, "PT-CapsNet: A novel prediction-tuning capsule network suitable for deeper architectures," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11996–12005, doi: 10.1109/ICCV48922.2021.01178.

[23] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "DeepCaps: Going deeper with capsule networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10725–10733, doi: 10.1109/CVPR.2019.01098.

[24] 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, doi: 10.1080/2150704X.2020.1766722.

[25] 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. (CVPR)*, Jun. 2005, pp. 539–546, doi: 10.1109/CVPR.2005.202.

[26] 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, doi: 10.4028/WWW.SCIENTIFIC.NET/AMR.989-994.5132.

[27] 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)*, Jun. 2021, pp. 13733–13742, doi: 10.1109/CVPR46437.2021.01352.

[28] 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.

[29] P. Demotte, K. Wijegunarathna, D. Meedeniya, and I. Perera, "Enhanced sentiment extraction architecture for social media content analysis using capsule networks," *Multimedia Tools Appl.*, 2021. [Online]. Available: https://link.springer.com/article/10.1007/s11042-021-11471-1

[30] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2006, pp. 1735–1742, doi: 10.1109/CVPR.2006.100.

[31] 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.

[32] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 1988–1996.

[33] 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.

[34] 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.

[35] N. K. Ahmed, E. E. Hemayed, and M. B. Fayek, "Hybrid Siamese network for unconstrained face verification and clustering under limited resources," *Big Data Cognit. Comput.*, vol. 4, no. 3, p. 19, Aug. 2020, doi: 10.3390/bdcc4030019.