Discussions of Different Deep Transfer Learning Models for Emotion Recognitions

CHIH-TA YEN (IEEE), AND KANG-HUA LI

1Department of Electrical Engineering, National Taiwan Ocean University, Keelung City 202301, Taiwan
2Department of Electrical Engineering, National Formosa University, Huwei, Yunlin 632, Taiwan

Corresponding author: Chih-Ta Yen (chihtayen@gmail.com)

This work was supported in part by the Ministry of Science and Technology—MOST under Grant MOST 108-2221-E-150-022-MY3, Grant MOST 110-2634-F-019-002, and Grant MOST 111-2221-E-019-074; and in part by the National Taiwan Ocean University.

ABSTRACT In recent years, facial emotion recognition (FER) has been a popular topic in affective computing. However, FER still faces many challenges in automatic recognition for several reasons, including quality control of sample data, extraction of effective features, creation of models, and multi-feature fusion, which have not been thoroughly researched and therefore are still hot topics in computer visualization. In view of the mature development of deep learning, deep learning methods are increasingly being used in FER. However, because deep learning requires a large amount of data to achieve effective training, many studies have employed transfer learning to compensate for this drawback. Nevertheless, there has been no universal approach for transfer learning in FER. Accordingly, this study used the five classic models in FER (i.e., ResNet-50, Xception, EfficientNet-B0, Inception, and DenseNet-121) to conduct a series of experiments: data preprocessing, training type, and the applicability of multi-stage pretraining. According to the results, class weight was the optimal technique for data balance. In addition, the freeze + fine-tuning training type can produce higher accuracy, regardless of the size of the dataset. Multi-stage training was also effective. Compared with the model accuracy in previous studies, the accuracy achieved in this study using the proposed transfer learning method was superior for both large and small datasets. Specifically, on AffectNet, the accuracy for the ResNet-50, Xception, EfficientNet-B0, Inception, and DenseNet-121 models increased by 8.37%, 10.45%, 10.45%, 8.55%, and 5.47%, respectively. On FER2013, the accuracy for these models increased by 5.72%, 2%, 10.45%, 5%, and 9%, respectively. These results proved the validity and advantages of the experiments in this study.

INDEX TERMS Transfer learning, convolutional neural network, facial emotion recognition (FER), affective computing, pretrained models, fine-tuning, FER2013, AffectNet.

I. INTRODUCTION

Facial expression is essential to non-verbal communication and is one of the most natural ways to convey our internal feelings in interpersonal interactions. In this study, we focused on facial expression recognition (FER) based on computer vision in the field of emotion recognition. Compared with the information from other non-verbal expressions, the facial expression information obtained from image processing enables a machine to operate in a way similar to how our brain processes and recognize information. The abundant information it provides then offers important clues for the machine to infer human intentions. Therefore, FER can be widely applied in many fields and is indispensable in affective computing.

Most of the early studies of FER used datasets developed in the laboratory, such as JAFFE [1] and CK+ [2]. However, the facial expression data obtained from these datasets were too similar; only positive expressions without any occlusion were included, and therefore cannot be applied to complex scenarios in real life. To solve this problem, many FER studies have established datasets in the unconstrained wild [3], [4], [5], [6]. Therefore, this study selected AffectNet and FER2013 as the
data sources. However, because the images in these datasets were taken in natural scenes, they often contain many factors irrelevant to facial expression, such as different backgrounds, ages, genders, races, appearances, localization errors, occlusion (hair, glasses, scarf, mask, hands, arms, food, and other objects that might be placed in front of the face), lights, head posture, and body movement. These factors all cause dataset shifts or feature-space shifts, limiting the accuracy and generalizability of the trained model.

In recent years, many methods have been used to explore the use of facial expression to recognize emotions [7]. Among these methods, using convolutional neural networks (CNNs) for model creation and improvement has been a trend [8]. Compared to the conventional method, which manually extracts texture features, CNNs can extract more effective texture features, generalize them to invisible data features, and automatically extract the geometric features of facial expressions and accurate attributes of facial appearances. However, although CNNs perform well in image recognition, they rely heavily on data and tags.

Transfer learning is a solution. Through the training of a pretrained model on a new task, the knowledge from the trained model is “transferred” to the pretrained model. This can lower the hardware requirements, reduce costs, and increase accuracy. Transfer learning can increase model efficacy by transferring the knowledge learned from other related data. It also relaxes the requirements for the training and testing data to be independent and identically distributed, thus enabling researchers to overcome the problem of lacking training data. In transfer learning, the training and testing data do not have to be independent and identically distributed, and there is no need for the model in the target domain to be retrained. This can substantially reduce the demand for the training and testing data in the target domain [9]. However, at the same time, whether to apply transfer learning to model training has been debated [10].

Model accuracy can be improved through, not only transfer learning, but also an integration of different deep learning models or ensemble learning. For example, Srinivasu et al. used MobileNetV2 and long short-term memory to categorize skin disorders [11]. Their study involved harnessing deep learning to classify images and the GLCM to perform model training with the HAM10000 dataset, thus achieving an accuracy of 93.89%. Sahoo et al. used ensemble learning methods (GooLeNet, ResNet18, and Inception_V3) to detect emotions in voices. They began by performing training through transfer learning, yielded their final training results through fuzzy rank-based ensemble learning, and obtained the optimal results in the existing literature using the databases of SAVEE, EmoDB, and RAVDESS [12].

Basak et al. proposed a deep learning-based method for HAR [13]. This method involved using a suitable coding approach to encode the features of 3D skeletal data into images, inputting these images into models to extract features, employing antlion optimization (ALO), a feature selection algorithm, to perform categorization, and conducting training and evaluation with the databases of UTD-MHAD, HDM 05, and NTU RGB+D 60 to achieve an accuracy of up to 98%. Bhattacharya et al. presented the Ensem-HAR, a deep learning ensemble model for HAR, which was designed using four CNN and LSTM models to perform feature extraction, integrating these features and inputting them into a random forest for categorization [14]. When accessed for its accuracy with different databases, this model achieved 98.70% with WISDM, 97.45% with PAMAP2, and 95.05% with UCI-HAR.

The studies cited above indicate that, regardless of the line of research, deep learning models can be integrated, or trained with ensemble learning, to improve their accuracy. This is why the present study conducted experiments using five typical deep learning models. Indeed, when a transfer learning method appropriate for training each of the models to perform FER is determined, these models can have further application.

This study examined the importance of transfer learning to FER and the influence of two decisive factors (source domain and training type) of transfer learning. Many studies on transfer learning with respect to FER have improved model accuracy by using multi-source transfer learning or training certain models, but no researchers have discussed the influence of different fine-tuning methods executed through transfer learning or the generalizability of transfer learning methods to different models. This study compared different transfer learning methods performed on several models using datasets of different sizes, and analyzed, from different perspectives, transfer learning methods deemed most appropriate for FER.

In this paper, Section 2 reviews relevant studies on transfer learning, Section 3 describes the experiments conducted in this study, and Section 4 discusses the research outcomes and delineates the contribution of this study to FER. Conclusions are made in Section 5.

II. RELATED WORK

Thanks to open source deep learning software, such as Keras, Torch, and TensorFlow, which provides many famous ImageNet-pretrained models (e.g., ResNet50, Xception, and EfficientNet), the threshold for using transfer learning has been lowered, leading to a considerable increase in the application of pretrained models to training in FER.

A technique of transfer learning is to transfer the pretrained model to a new model for task training. For example, Sekaran et al. [15] used the AlexNet model, which was pretrained as ImageNet, for FER transfer learning. The source domain was ImageNet (the source task was object recognition, and the target domains are FER2013 and CK+ (the target task was emotion classification). The RAdam optimizer and image data augmentation were used for model training. Eventually, the researchers obtained accuracies of 70.52% and 99.44% on FER2013 and CK+, respectively. Other datasets have also been used for domain training models. For example, Savchenko et al. [16], [17] pretrained the
MobileNet model on the VGGFACE dataset (a large human face dataset) and then transferred it to the AffectNet dataset for model training. The results showed that using the pretrained model of VGGFACE, which is related to human faces, produced better transfer learning results than using ImageNet.

Some studies have performed transfer learning through fine-tuning with multiple training iterations. Nguyen et al. [18] used multiple datasets for the training of a pretrained model for transfer learning with multiple source domains. The AlexNet model was firstly trained on the ImageNet dataset and then transferred to two FER datasets (i.e., FER2013 and EmotiW) for model training through a two-stage process. According to the results, the accuracy achieved using the pretrained model through the transfer learning with multiple source domains was 15% higher than the accuracy obtained using the baseline model. Gupta et al. [19] selected ResNet50 and ResNet152 as the transfer learning models and pretrained them on two datasets using cross-databases. The results showed that using the VGGFACE dataset for second-time model pretraining gave an accuracy that is 26% higher than that obtained using the model pretrained on ImageNet. Kong et al. [6] used the ResNet50 model with the VGGFACE2 dataset for six-stage pretraining to test FER domain adaptation. When the source and target domains both belong to the FER database, the researchers trained a combination of the datasets in the laboratory (CK+, IASLab, and GEMEP) and outdoor space (AffectNet, Aff-Wild2, and FER-2013). They argued that multi-stage pretraining can enhance the effect of generalization and further increase the accuracy of the target datasets. In addition, they also examined the effects of using different training approaches and found that performing fine-tuning by freezing the convolution layer gave better results than fully performing fine-tuning.

The literature review shows that transfer learning studies mainly comprise two aspects: the dataset and the training method. Regarding the dataset, transfer learning can be performed using datasets with a single source domain, ImageNet, or face-related datasets. Transfer learning can also be performed through multi-stage pretraining, which can enhance the transferability of the knowledge from the source model. As for the training method, most studies used different numbers of freezing layers for fine-tuning according to the model characteristics or directly fine-tuned all layers for training.

Studies on transfer learning rarely use multiple models, instead using a single model, for universal transfer learning. Thus, this study used five mainstream models to evaluate the applicability of transfer learning on FER. The pretrained model of ImageNet was transferred to the large dataset AffectNet and the small dataset FER2013 to explore the influence of transfer learning on the size of the target data. In addition, this study compared the training types of fine-tuning and freeze + fine-tuning to determine which is most suitable for FER transfer learning. Many studies have neglected the influence of data preprocessing on transfer learning outcomes, especially in the FER field where data classes are extremely imbalanced. Therefore, this study also compared the effects of data balancing and data augmentation on AffectNet and FER2013. Data recorded in the real world usually constitute small datasets. Therefore, in this study, knowledge is transferred from ImageNet firstly to AffectNet and then to FER2013 to verify the effect of multi-stage transfer learning, thereby identifying the transfer learning method most suitable for FER.

III. EMPIRICAL EVALUATION

A. FER DATASETS

Because many FER datasets are characterized by wide changes, many researchers have been investigating these datasets [20], [21]. The most important distinction is whether the targets in the FER datasets contain specific or natural facial expressions. The datasets comprising specific facial expressions were recorded in a controlled laboratory, with the same posture, lighting, background, and camera. Because the facial expressions were formed according to actors’ stereotypes of emotions, they are usually too exaggerated. On the contrary, the natural facial expressions were made when the actors were in a natural, relaxed situation; they are more representative of the emotions in real life. These natural facial expressions are often collected online to form the datasets, which are more suitable for real-life tasks than the datasets made in a laboratory. Thus, the datasets adopted in this study comprised natural facial expressions. To ensure a fair comparison, all the experiments were conducted on open datasets, namely AffectNet and FER2013.

1) AffectNet

AffectNet [4] is the largest facial expression dataset in the world, all the image data in which were collected from the Internet. The subsets comprise eight classes of emotion, namely happy, sad, anger, fear, surprise, disgust, contempt, and neutral. The dataset consists of 287,651 images in the training set and 4,000 images in the validation set (each class containing 500 images); the data distribution is presented in Fig. 1. The accuracy for face recognition is 65±5% [4]. Because the test set is not open, the majority of the studies on AffectNet [17], [22] used the validation set as the test set. The original data are shown in Fig. 2. To remove the noise caused by the non-face areas, this study used the bounding boxes data converted by the OpenCV HaarCascade Classifier provided by AffectNet to crop the faces. The results after cropping are shown in Fig. 3. The size of each image was set to be 224 × 224.

2) FER2013

The FER2013 dataset was introduced to the facial expression recognition challenge in the 30th International Conference on Machine Learning in 2013 [23]. The dataset consists of face images that match emotion keywords, which are collected using Google Image Search APIs. All the images have been cropped, with only face areas remaining. The dataset contains
seven emotions: happy, sad, anger, fear, surprise, disgust, contempt, and neutral. The data were distributed according to previous research [22], [24], [25], namely, 28,709 images in the training set and 3,589 images in each of the validation and test sets (PublicTest and PrivateTest, respectively), as shown in Fig. 4. The accuracy of face recognition is 64±5% [26]. To perform transfer learning, the size of the images was adjusted to 224 × 224 grayscale, as shown in Fig. 5.

**B. FER MODEL AND TRAINING PROCEDURE**

This study adopted the mainstream models of ResNet50, EfficientNet, Xception, Inception, and EfficientNet, provided by Keras and Pytorch, as the transfer learning models. The models are introduced in the following paragraphs and the model frameworks are presented in Fig. 6.

- **ResNet50** [21] is one of the most common CNN models, which can effectively solve the problem of degradation caused by the increase of the number of network layers. This model won the championship in the ImageNet Challenge 2015. The major concept of ResNet is adding a skip connection to the network, which allows the original input signal to be directly passed to the next layer. Each residual block comprises a series of layers: convolutional layer, batch normalization layer, pooling layer, and ReLU. Through the residual blocks, the gradient degradation and overfitting problems can be effectively solved.

- **Xception** [27] is a trendy CNN model. It combines the concepts of GoogleNet and ResNet but the inception module is replaced with a separable convolutional layer. The separable convolutional layer assumes that the spatial and cross-channel patterns can be separately simulated. Therefore, its number of parameters, memory usage, and computational complexity are lower than those of the convolutional layer; it also performs better than the convolutional layer.

- **EfficientNet** [28] is a new network model developed by Google that has become popular due to its features of being light, fast, and accurate. The model design was inspired by MnasNet. It not only used the mobile inverted bottleneck convolution (Macon) but also introduced the attention mechanism of Squeeze-and-Excitation Network (SENet). The multi-objective neural architecture search (NAS) was then used to design a high-efficiency model EfficientNet-B0, and EfficientNet B1–B7 were developed by setting different coefficients. Because the structures are similar, this study only used EfficientNet-B0 to serve as the EfficientNet.

- **Inception** [29], developed by Google in 2014, is also called GooLeNet. The Goole team used a subnetwork called an inception module to use parameters more efficiently, that is, the number of parameters was minimized while the accuracy of the network was ensured. Some variants were subsequently proposed. In this study, Inception-v3, which has been most frequently used in FER, was adopted.

- **DenseNet-121** [30] directly connects all the layers that generate feature maps, which enables each convolution layer to know the feature maps output from the previous convolution layer, thereby realizing the dense connections between layers. In addition, DenseNet
FIGURE 6. Model framework.
adopts many methods to solve the gradient vanishing problem.

In a CNN, the convolution layers in the hidden layers perform convolution computation through the kernel made up of neurons to extract features and generate feature maps. The weights on the kernels can be used as the method for identifying the features related to the target tasks. In the CNN training process, each neuron uses different kernels to identify different feature maps and pass them to the next neuron, so that the later neurons can search for higher-level features in the feature maps. Through training, the features of the training data are stored in the weights of kernels. The feature map in the previous layer catches the basic features of an image (such as edges and corners), and the one in the next layer detects more complex features (such as shapes and patterns). The later layers can extract higher-level features (such as information related to the target task), and finally all the features are passed to the dense layer for subsequent classification.

Transfer learning is a process of fine-tuning the pretrained network model to create a new model in a specified domain. Because the model is pretrained on a huge dataset, it has sufficient spatial hierarchy features and therefore can serve as a universal model for the visual world. In terms of feature transfer learning, the knowledge learned after training will be stored in the weights. Through the transfer between layers, the weights that have been trained can be repeatedly used, which enables the newly created model to possess some knowledge in advance. The new model can therefore be used to perform new tasks. Hence, most of the studies on transfer learning in FER chose to transplant the pretrained model into the new model, that is, replacing the classifier layer or some layers to make the model suitable for the tasks in the target dataset, as shown in Fig. 7. This study also adopted this model design, generating the transfer learning model by replacing all the hidden layers.

The common problems in FER are class imbalance and data insufficiency. Therefore, this study conducted experiments involving class weight and data augmentation by retraining the model that had not undergone transfer learning. The model that was confirmed to be optimal would then be used to test different training types for transfer learning.

The training types in this study were divided into fine-tuning and freeze + fine-tuning. To highlight the effects and applicability of freeze training and the versatility of different models, this study used the most original training method in the training of freeze + fine-tuning. Two-stage training was adopted. In the first stage, all layers, except for the classifier layer, were frozen until they converged. The second stage unfroze the frozen layers to fine tune all the layers until convergence. In the fine-tuning training, all the layers were fine-tuned for training.

To ensure the comparability and fairness of different training types, all the experiments were set to have 60 training iterations. In the freeze + fine-tuning training experiment, to ensure the convergence of all frozen layers, the number of training iterations at both the first and second stages was set at 30. In all the experiments, the model that showed the best performance in the validation set was selected.

As for the equipment, Keras and TensorFlow were adopted for model training. The experiments were conducted on a personal computer with an Intel i7-8700 CPU, NVIDIA GeForce RTX 2060 GPU, and 32.0 GB RAM, and the environment was Anaconda on Windows 10. The learning rates were 0.01 and 0.001 for freeze training and fine-tuning training, respectively. The optimizer was Adam; the batch was 48; the attenuation rate for fine-tuning training was 0.000001. The learning rate was reduced by 0.01 every three times when the accuracy of the validation set cannot be improved. For the evaluation criteria, the accuracy of top-1 was reported, which is the proportion of samples being accurately predicted. Because most of the datasets are imbalanced, this study also reported the weighted F1 score, as shown in Eq. 1, where \( N_l \) is the number of samples in the \( l \) th class, and \( N_{total} \) is the total number of samples being evaluated [6]. To accurately evaluate the model applicability in the real world, all the aforementioned accuracy values are results predicted by the new model generated after model training.

\[
Weighted \ F1 = 2 \sum \frac{N_l \cdot \text{precision}_l \times \text{recall}_l}{N_{total} \cdot \text{precision}_l + \text{recall}_l} \tag{1}
\]

C. CLASS WEIGHT AND DATA AUGMENTATION

The problem of class imbalance can be observed in the existing datasets. Some expressions, such as smiles, are very easily elicited and annotated, while disgust, anger, and other uncommon expressions are very difficult to detect. Thus, the FER datasets in real-life scenarios contain a higher proportion of smiles. If a dataset is dominated by one class but lacks the samples of another class, this imbalance may mislead the
model’s judgment of the relevance of the target task. This class imbalance problem, which has been neglected by many previous studies, is the major challenge in FER. To that end, we applied class weight balancing. Class weight balancing involves focusing on classes with fewer data during training, and assigning different weights (i.e., the so-called “weighted data spaces”) to different classes of data to ensure the same total weight for each of the classes, and thus eliminate the influence of the sample size on datasets as Eq. (2) [31].

$$class\_weight_i = \frac{\#samples_i}{\#classes \times \#samples}$$

(2)

where \#samples is the number of measurements in a dataset, \#classes is the number of classes, \#samples\_i is the number of measurements for class i.

Another problem in FER is the lack of data. Because the labeling of FER data requires professional skills, and facial expressions are subjective and vary across different regions and cultures, it is extremely costly, difficult, and time-consuming to label and classify facial expressions of individuals with different skin colors, educational levels, and emotional levels. This is the reason why there are far fewer FER datasets compared to other fields. To examine the influence of data augmentation on the FER dataset more effectively, we performed augmentation on datasets of different sizes. The augmentation procedure involved using the ImageDataGenerator function of TensorFlow to read each sample image and perform random transformations (zooming, rotation, flipping, and displacement) to generate new sample images; thus, the sizes of the datasets were left unchanged. Table 1 shows the ImageDataGenerator parameters for data augmentation.

The results of data augmentation are shown in Fig. 8, the upper-left side of which presents the original images. The other images were sample images generated through the augmentation procedure; an analysis of these images suggested that, although they underwent several augmentations, they were randomly combined taking into account augmentation parameters and did not overlap with each other.

The models we used in data processing experiments had been retrained to eliminate the influence of the weight of transfer learning. An analysis of the F1 score showed that, because of data imbalances, the F1 score was slightly lower than the accuracy value most of the time. The results of model training with the AffectNet dataset are presented in Table 2. On the basis of the results, the accuracy and F1 score of models without class weight balancing showed greater differences, ranging from 1% to 6%, whereas those of models with class weight balancing were similar. This indicated that class weight balancing allowed the models to be trained with significantly fewer emotion classes and exhibit greater performance. Such differences were not found when the models were trained with FER2013. This suggested that the models could not effectively learn the features extracted from small datasets (because of the limited sizes of the data) thus yielding similar accuracy values and F1 scores (Table 3).

An observation of the confusion matrices showed that, when class weight balancing was not performed, none of the models could be trained to predict the Contempt or Disgust class when they were fed with AffectNet as shown in Table 4. All the emotion classes of were limited in size, and they predicted many of the classes as Neutral. However, after class weight balancing, the models all became more accurate in predicting Contempt and Disgust classes when trained with AffectNet as shown in Table 4.

As for the results of being trained with FER2013 as shown in Table 5, none of the models could be trained to predict the Disgust class when no class weight balancing was performed, but they all became more accurate in predicting this class if class weight balancing was performed. In addition, accuracy across all the models declined after class weight balancing, because the models focused less on the classes they had accurately predicted, thereby becoming less accurate in predicting them (e.g., Happy).

After determining how class weight balancing improved model training with datasets of different sizes, further data augmentation was performed to ascertain whether it could improve accuracy—and we obtained surprising results. The accuracy of models trained with FER2013 decreased, rather than increased. This was probably because images in all classes were highly similar, and the classes showed negligible differences. For these reasons, the accuracy errors became larger after data augmentation, preventing the models from effectively distinguishing between the classes. This assumption was validated by examining the patterns in the confusion

---

**TABLE 1. ImageDataGenerator parameters for data augmentation.**

| Parameter       | Value           |
|-----------------|-----------------|
| zca_epsilon     | 0.0000001       |
| rotation_range  | 0.1             |
| width_shift_range | -0.1          |
| height_shift_range | -0.1          |
| shear_range     | 0.0             |
| zoom_range      | 0.1             |
| channel_shift_range | 1.05          |
| fill_mode       | nearest         |
| horizontal_flip | True            |

**FIGURE 8. Results of data augmentation on FER2013.**
### TABLE 2. Comparison between different data processing with AffectNet.

| Type                        | ResNet-50 | Xception | EfficientNet-B0 | Inception | DenseNet121 |
|-----------------------------|-----------|----------|-----------------|-----------|-------------|
|                             | ACC       | F1       | ACC             | F1        | ACC         | F1       | ACC | F1       | ACC | F1 | ACC | F1 |
| None                        | 47%       | 43%      | 49%             | 45%       | 50%         | 47%      | 48% | 43%      | 47% | 43% |
| Class weight                | 55%       | 55%      | 58%             | 58%       | 55%         | 55%      | 56% | 56%      | 55% | 55% |
| Class weight and Data       | 54%       | 54%      | 58%             | 58%       | 54%         | 54%      | 58% | 58%      | 58% | 58% |
| augmentation                |           |          |                 |           |             |          |     |           |     |     |

### TABLE 3. Comparison between different data processing with FER2013.

| Type                                | ResNet-50 | Xception | EfficientNet-B0 | Inception | DenseNet121 |
|-------------------------------------|-----------|----------|-----------------|-----------|-------------|
|                                     | ACC       | F1       | ACC             | F1        | ACC         | F1       | ACC | F1       | ACC | F1 | ACC | F1 |
| None                                | 60%       | 55%      | 47%             | 37%       | 63%         | 60%      | 65% | 60%      | 59% | 53% |
| Class weight                        | 47%       | 43%      | 56%             | 51%       | 61%         | 59%      | 54% | 49%      | 49% | 43% |
| Class weight and Data               | 24%       | 19%      | 50%             | 44%       | 52%         | 45%      | 54% | 47%      | 48% | 41% |
| augmentation                        |           |          |                 |           |             |          |     |           |     |     |

### TABLE 4. Comparison between models tested with affectnet regarding their accuracy of identifying emotion classes.

| Model          | Type                        | Anger | Contempt | Disgust | Fear | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Surprise | Neutral | Fear | Happiness | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Anger | Contempt | Disgust | Fear | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Anger | Contempt | Disgust | Fear | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Anger | Contempt | Disgust | Fear | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Anger | Contempt | Disgust | Fear | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Anger | Contempt | Disgust | Fear | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Anger | Contempt | Disgust | Fear | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Neutral | Sad | Disgust | Neutral | Sad | Happiness | Neutral | Sad | Surprise | Experimental matrices of all models (Figs. 29–33); the influence of class weight balancing was diminished after data augmentation (for example, none of the models could predict the Anger, Disgust, or Fear class, as Figs. 20–29 suggested), thus causing the models to make incorrect classifications and undermining their accuracy. The accuracy of models trained with...
TABLE 6. Different training types with different models in AffectNet.

| Source Dataset      | Type            | Target dataset (AffectNet) |
|---------------------|-----------------|----------------------------|
|                     |                 | ResNet50 | Xception | EfficientNet-B0 | Inception | DenseNet121 |
|                     |                 | ACC     | F1       | ACC     | F1       | ACC     | F1       | ACC     | F1       |
| ImageNet            | fine-tuning     | 54%     | 54%      | 59%     | 59%      | 59%     | 58%      | 57%     | 57%      |
|                     | Freeze + fine-tuning | 58%   | 58%      | 61%     | 61%      | 59%     | 59%      | 60%     | 60%      | 53%     | 50%      |

TABLE 7. Different training types with different models in FER2013.

| Source Dataset & AffectNet | Type            | Target dataset (FER2013) |
|----------------------------|-----------------|---------------------------|
|                            |                 | ResNet50 | Xception | EfficientNet-B0 | Inception | DenseNet121 |
|                            |                 | ACC     | F1       | ACC     | F1       | ACC     | F1       | ACC     | F1       |
| ImageNet                   | fine-tuning     | 59%     | 56%      | 44%     | 42%      | 68%     | 67%      | 50%     | 44%      | 69%     | 66%      |
|                            | Freeze + fine-tuning | 70%   | 69%      | 65%     | 65%      | 68%     | 67%      | 67%     | 66%      | 71%     | 71%      |
| ImageNet                   | fine-tuning     | 67%     | 64%      | 70%     | 67%      | 69%     | 66%      | 68%     | 67%      | 67%     | 63%      |
|                            | Freeze + fine-tuning | 67%   | 65%      | 68%     | 68%      | 70%     | 68%      | 67%     | 66%      | 63%     | 57%      |

AffectNet, which contained massive data and thus allowed class weight balancing to exert its influence, showed limited improvement even after data augmentation: Of the five models, two improved by 2%–3% in accuracy, two decreased by 1% in accuracy, and only the Xception model’s accuracy remained constant.

According to the aforementioned results, the technique of class weight balance was effective. However, in FER, data augmentation was ineffective because it caused the model to have difficulty distinguishing between different expressions. We also observed that the accuracy of the model trained using class weight in small datasets (FER2013) was unable to exceed the accuracy of the models without data preprocessing. This is because the datasets were too small to allow every class to learn enough features. Therefore, the overall accuracy was low. However, this shortcoming can be compensated through transfer learning. Therefore, all the subsequent experiments on training type difference were based on transfer learning, and class weight balance was employed in the training.

D. TRAINING TYPE

In this study, an experiment was conducted to test the effect of different training types in transfer learning. In most of the studies, the training type is usually divided into fine-tuning and freeze + fine-tuning. According to the results of the data preprocessing experiments and the consistency, in this experiment, class weight balance was used for data preprocessing in the training of all the models. The concepts of different training types are introduced as follows. Before a specific task began, the CNN models, including ResNet-50, Xception, EfficientNet-B0, Inception, and DenseNet121, were pretrained in the source domain (ImageNet or AffectNet) according to the demands of the target domain (AffectNet or FER2013), in order to learn the universal features. In addition, the model on AffectNet that showed the highest accuracy was selected for the multi-stage pretraining.

1) FINE-TUNING

All the layers were fine-tuned to adapt to the tasks in the target dataset. This technique can also serve as the regularizer to prevent the effect exerted by overfitting. In the fine-tuning training of this study, all layers of the model were directly trained, to enable the model to be applicable to the task classes in the target domain.

2) FREEZE + FINE-TUNING

The backpropagation algorithm in CNN passes loss (i.e., the gap between predicted value and real value) backward and computes the gradient of each layer according to the error sent back, thereby updating the parameters of the designated layer. In freeze training, the designated layer was not involved in the model training, that is, the parameter updated by backpropagation after each iteration will not enter the frozen layer, and therefore the weight parameter value in the kernel composed of neurons will not be updated. Thus, the feature extraction method of the designated layer remains unchanged before the model training finishes. This is to prevent the learned knowledge (weight) from being destroyed when the knowledge is transferred, which may result in failing to extract effective features. Although not many FER studies have adopted this technique, it has been widely applied in other fields, such as medicine and malware detection.

The model training conducted in this study consisted of two parts. The first part involved freezing until convergence; all transferred layers were frozen and only the classification layer was trained. The second part entailed fine-tuning all layers of the models until they converged. For model training with AffectNet, accuracy not only remained constant for EfficientNet-B0 and improved for the fine-tuned
DenseNet-121, but it also increased for models that underwent freezing and fine-tuning (Table 6), suggesting that it is necessary to train a classifier without undermining the weight of its source domain. Moreover, for model training with FER2013 using ImageNet & AffectNet as the multi-source domain, accuracy results were similar to those found in model training with AffectNet. Such similarity was attributed to the fact that the AffectNet source domain shared similar data with the target domain, and thus the models underwent less fine-tuning when ImageNet was used as the source domain.

A comparison between the AffectNet and ImageNet source domains regarding their influence on the training of models with FER2013 indicated that, with transfer learning implemented using ImageNet & AffectNet as the multi-source domain, the models yielded better results, except for ResNet-50 and DenseNet-121 (Table 7). In addition, negligible differences were observed between accuracy and the F1 score across different source domains and transfer learning methods, because the model learned features effectively. At the same time, DenseNet-121 sustained the most influence of imbalanced data, and its accuracy and F1 score showed the largest differences, whereas EfficientNet-B0 sustained the least influence of the data, and it was subject to less influence than the other models across different source domains and transfer learning methods.

**IV. DISCUSSION**

This study explored the effects and verified the necessity of using transfer learning in FER. This section discusses the training process and training methods (training types, data preprocessing, and multiple source domain training) and compares the results of this study with those of other studies.

First, this study concluded that it is necessary to use transfer learning in FER. Compared with the model trained from scratch, when the target domain was AffectNet, all the models showed increased accuracy, by 3% for ResNet-50, 3% for Xception, 4% for EfficientNet-B0, 4% for Inception, and 2% for DenseNet-121. When the target source was FER2013, two baselines were adopted for comparison. The first one is the highest accuracy achieved by the model trained from scratch (i.e., the result of model training without class...
In this case, the largest increase of accuracy in ResNet-50, Xception, EfficientNet-B0, Inception, and DenseNet-121 was 10%, 23%, 7%, 3%, and 12%, respectively. The second baseline was used to compare the degree of accuracy increase of the models that used the same data preprocessing technique. Therefore, the training results of the models using class weight balance were used as the baselines for the models trained from scratch. In this case, the largest accuracy increases in ResNet-50, Xception, EfficientNet-B0, Inception, and DenseNet-121 were 23%, 14%, 9%, 14%, and 22%, respectively. The comparison indicates that the models with transfer learning produced higher accuracy, and the increase in accuracy in the small dataset (FER2013) was higher than that in the large dataset (AffectNet).

As for data preprocessing, because the FER data are difficult to collect and define, class imbalance is a serious problem. Therefore, we used the techniques of class weight balance and data augmentation in the experiment to observe the effect of data preprocessing technique on model training. The results showed that class weight balance significantly improved the model accuracy in the large dataset (AffectNet), with the accuracy for ResNet-50, Xception, EfficientNet-B0, Inception, and DenseNet-121 increasing by 8%, 9%, 5%, 8%, and 8%, respectively. In the small dataset (FER2013), because the amount of data that can be trained is low and the model paid the same attention to all classes, the overall accuracy was lower. This study also found that because of FER’s high requirement for data quality, data augmentation would not increase model accuracy; instead, the accuracy decreased in most of the models. Accordingly, this study suggested only using class weight balance for transfer learning in FER model training.

Regarding multiple source domain training, most of the models pretrained twice in similar domains achieved higher accuracy than those pretrained in a single source domain. As for the training types for transfer learning, with the large dataset (AffectNet), the accuracy of the models trained through freeze + fine-tuning was higher than those of the models trained through only fine-tuning. For the small dataset (FER2013), the models trained through freeze + fine-tuning also exhibited better accuracy than those trained through only fine-tuning, despite the difference in the source domains. However, we also noticed that when the source domains were similar, using a different training type would not generate distinct results. This may be because the high-level features of the source domain (AffectNet) were similar to the tasks in the target domain. Therefore, conducting freeze training would not make much difference. Accordingly, this study suggested conducting freeze + fine-tuning training for FER model training on large datasets. On small datasets, although we also suggested using freeze + fine-tuning training, more effort should be placed on pretraining in multiple similar domains to enhance accuracy.

In terms of training speed, in the small dataset (FER2013), the transfer learning speed of both the training types can be accelerated, while the speed increase in the large dataset (AffectNet) was not significant. Moreover, the models...
undergoing the freeze + fine-tuning training showed higher speed and accuracy than those trained from scratch or fine-tuned (Figs. 9-28). Notably, to ensure that all the models could be trained to converge in the freeze training, in the first freeze training of the freeze + fine-tuning training, all the iterations were set to be 30.

Regarding the differences discussed between the models in terms of classification performed on different datasets, only the model that achieved the highest accuracy, regardless of the training type, was in discussion. For models that achieved the same accuracy level in each emotion class regardless of the training type, only one of the classes was examined. When trained with AffectNet, all models became the least accurate in classifying the images in the Contempt class; these images were generally classified as Happy and Neutral. The images in the Happy class were the easiest to classify; EfficientNet-B0 was the most successful in classifying them, probably because emotions happiness were the most distinguishable from those associated with the other

effects (Figs. 13-14). Moreover, when the models were trained with FER2013, they became the most accurate in identifying Happy and least accurate in identifying Fear (Figs. 19-28). Both types of transfer learning facilitated the training process where the models were trained with the small dataset FER2013, but their facilitation became less noticeable when the large dataset AffectNet was used.

As for which model was the most accurate in identifying which emotion class when AffectNet was used for training, Xception achieved the highest identification accuracy in Disgust and Fear; Inception did so in Anger, Neutral, and Surprise; ResNet50 did so in Sadness; EfficientNet-B0 did so in Happy; and DenseNet121 did so in Contempt as show in Table 8.

When all models were trained with FER2013, they became the most accurate in identifying Happy but the least accurate in identifying Fear. It is also observed in Table 9 that ResNet50 achieved the highest identification accuracy in Anger and Fear, EfficientNet-B0 did so in Sadness and Surprise, and Inception did so in Neutral. Overall, Xception was more accurate than the other models when trained with
TABLE 8. Comparison between best models tested with AffectNet regarding their accuracy of identifying emotion classes.

| Emotion Model | Anger | Contempt | Disgust | Fear | Happiness | Neutral | Sadness | Surprise |
|---------------|-------|----------|---------|------|-----------|---------|---------|----------|
| ResNet50      | 57%   | 41%      | 46%     | 57%  | 81%       | 62%     | 64%     | 59%      |
| Xception      | 58%   | 49%      | 58%     | 66%  | 77%       | 58%     | 60%     | 59%      |
| EfficientNet-B0 | 58% | 41%      | 46%     | 59%  | 82%       | 59%     | 63%     | 59%      |
| Inception     | 60%   | 46%      | 49%     | 59%  | 80%       | 65%     | 62%     | 62%      |
| DenseNet121   | 53%   | 51%      | 56%     | 60%  | 75%       | 51%     | 56%     | 58%      |

TABLE 9. Comparison between best models tested with FER2013 regarding their accuracy of identifying emotion classes.

| Emotion Model      | Anger | Disgust | Fear | Happy | Sad | Surprise | Neutral |
|--------------------|-------|---------|------|-------|-----|----------|---------|
| ResNet50           | 63%   | 78%     | 52%  | 88%   | 58% | 81%      | 68%     |
| Xception           | 37%   | 67%     | 37%  | 49%   | 32% | 80%      | 35%     |
| EfficientNet-B0    | 62%   | 91%     | 49%  | 88%   | 58% | 83%      | 70%     |
| Inception          | 61%   | 73%     | 46%  | 87%   | 54% | 80%      | 69%     |
| DenseNet121        | 58%   | 53%     | 43%  | 90%   | 59% | 80%      | 71%     |

![FIGURE 20](image1.png) Training results for ResNet50 on FER2013 (Loss).

![FIGURE 21](image2.png) Training results for Xception on FER2013 (Loss).

![FIGURE 22](image3.png) Training results for EfficientNet-B0 on FER2013 (Loss).

AffectNet, and DenseNet121 was more accurate than the other models when trained FER2013. However, yielding the optimal testing results involved different source domains and training types in transfer learning, depending on the model (Tables 10 and 11).
To prove the effectiveness of our experiments, we compared the results of transfer learning implemented for the same models, datasets, and source domains with those reported in existing studies (Table 12). The comparison showed that our recommended transfer learning methods were effective in FER. This was explained by the following accuracy improvements: ResNet-50 improved by 8.37%, Xception and EfficientNet-B0 by 10.45%, Inception by 8.55%, and DenseNet-121 by 5.47% when trained with AffectNet; ResNet-50 by 5.72%, Xception by 2%, EfficientNet-B0 by 10.45%, Inception by 5%, and

| Model     | Type            |
|-----------|-----------------|
| ResNet-50 | Freeze + fine-tuning |
| Xception  | Freeze + fine-tuning |
| EfficientNet-B0 | Freeze + fine-tuning |
| Inception | Freeze + fine-tuning |
| DenseNet-121 | fine-tuning     |
Different sizes, using different data preprocessing techniques, different training methods, and different models. We first compared these data preprocessing techniques and decided to use class weight training after taking into account the considerable differences in sample size between all emotion classes of FER2013 and the generalizability of datasets large and small. Next, we examined the influence of different training types in transfer learning and concluded that, with ImageNet used as the source domain, all models achieved the highest accuracy when they underwent freeze + fine-tuning whether with the large dataset AffectNet or the small dataset FER2013 (except for EfficientNet-B0, whose accuracy remained constant, and DenseNet-121, which became less accurate than the other models). Regarding the speed of model training, the models were trained faster with freeze + fine-tuning than with re-training or fine-tuning, regardless of the size of the dataset used.

Moreover, an observation of a multi-source transfer learning experiment conducted using FER2013 on source domains relating to the target domain, indicated that the transfer learning methods exerted less influence and there were no significant differences in training results when these methods were performed using AffectNet as the source domain, compared to ImageNet. This finding suggested that using the right source domain can lead to more significant improvement for transfer learning models than using the right training method. However, yielding the optimal training results involving different source domains and training types in transfer learning, depending on the model.

We also determined the best models with respect to different datasets. Specifically, while all models averaged 59% accuracy when trained using AffectNet, Xception was the most accurate (61%), followed sequentially by Inception (60%), EfficientNet-B0 (59%), ResNet-50 (58%), and DenseNet-121 (57%). All models averaged 69.8% in accuracy when trained using FER2013; DenseNet-121 was the most accurate (71%), followed sequentially by ResNet-50, Xception, and EfficientNet-B0 (all achieving 70%), and Inception (68%).

Regarding the future direction of our research, we may attempt to further improve model accuracy by taking into account existing ensemble learning methods. We may also focus on the practical application of the models by building datasets, training the models with the data through transfer learning, and using them for real-time identification.

V. CONCLUSION
This study examined the transfer learning of five CNN models and presented straightforward and thorough comparisons of the models. Researchers with hardware limitations, or a lack of software programs, can draw on this study to seek appropriate models and transfer learning methods. We analyzed these CNN models, which differ in the framework. They are transfer learning models commonly used across different fields: ResNet-50, Xception, EfficientNet-B0, Inception, and DenseNet-121—all of which differ in the framework.

We conducted exhaustive experiments on datasets of different sizes, using different data preprocessing techniques, different training methods, and different models. We first compared these data preprocessing techniques and decided to use class weight training after taking into account the considerable differences in sample size between all emotion classes of FER2013 and the generalizability of datasets large and small. Next, we examined the influence of different training types in transfer learning and concluded that, with ImageNet used as the source domain, all models achieved the highest accuracy when they underwent freeze + fine-tuning whether with the large dataset AffectNet or the small dataset FER2013 (except for EfficientNet-B0, whose accuracy remained constant, and DenseNet-121, which became less accurate than the other models). Regarding the speed of model training, the models were trained faster with freeze + fine-tuning than with re-training or fine-tuning, regardless of the size of the dataset used.

Moreover, an observation of a multi-source transfer learning experiment conducted using FER2013 on source domains relating to the target domain, indicated that the transfer learning methods exerted less influence and there were no significant differences in training results when these methods were performed using AffectNet as the source domain, compared to ImageNet. This finding suggested that using the right source domain can lead to more significant improvement for transfer learning models than using the right training method. However, yielding the optimal training results involving different source domains and training types in transfer learning, depending on the model.

We also determined the best models with respect to different datasets. Specifically, while all models averaged 59% accuracy when trained using AffectNet, Xception was the most accurate (61%), followed sequentially by Inception (60%), EfficientNet-B0 (59%), ResNet-50 (58%), and DenseNet-121 (57%). All models averaged 69.8% in accuracy when trained using FER2013; DenseNet-121 was the most accurate (71%), followed sequentially by ResNet-50, Xception, and EfficientNet-B0 (all achieving 70%), and Inception (68%).

Regarding the future direction of our research, we may attempt to further improve model accuracy by taking into account existing ensemble learning methods. We may also focus on the practical application of the models by building datasets, training the models with the data through transfer learning, and using them for real-time identification.

ACKNOWLEDGMENT
The authors would like to thank the editor kind coordination and also would like to thank the reviewers for constructive suggestions.

REFERENCES
[1] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, “Coding facial expressions with Gabor wavelets,” in Proc. 3rd IEEE Int. Conf. Autom. Face Gesture Recognit., Apr. 1998, pp. 200–205, doi: 10.1109/AFGR.1998.670949.
[2] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, “The extended cohn-kanade dataset (Ck+): A complete dataset for action unit and emotion-specified expression,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.-Workshops, Jun. 2010, pp. 94–101, doi: 10.1109/CVPRW.2010.5534362.
[8] S. Farid, K. Hornik, and T. Frank, “Facial expression recognition based on temporal features,” in Proc. 37th Int. Conf. Mach. Learn. (ICML), 2020, pp. 230–239.

[9] E. H. Huang, W. H. Ko, and J. Chen, “A survey of facial expression recognition methods,” IEEE Trans. Cybern., vol. 50, no. 2, pp. 526–543, Feb. 2020.

[10] T. Wang, J. Chen, and Y. Sun, “A survey on facial expression recognition,” Image Process. On Line, vol. 9, no. 1, pp. 59–77, 2019.

[11] V. K. Gunjan, P. N. Suganthan, J. Haase, and A. Kumar, Eds. Singapore: Springer, 2021, pp. 273–279, doi: 10.1007/978-981-33-6691-6_30.

[12] D. S. H. Noor, M. S. A. Bakar, and M. S. Ismail, “Facial expression recognition on small datasets using transfer learning,” in Proc. IEEE Infotech. Soc. (IT Soc), Nov. 2020, pp. 878–883, doi: 10.1109/FG47880.2020.001143.

[13] R. H. K. Sahoo, I. Dutta, M. F. Ijaz, and P. K. Singh, “Deep learning approaches for facial emotion recognition: A case study on FER-2013,” in Advances in Hybridization of Intelligent Methods (Smart Innovation, Systems and Technologies), vol. 85, I. Hatizilergoudis and V. Palade, Eds. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-66790-4_1.

[14] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[15] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. 39th Int. Conf. Mach. Learn., 2019, pp. 6105–6114.

[16] J. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojnaw, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 2818–2828, doi: 10.1109/CVPR.2016.308.

[17] T. L. FuzzyNet: Fuzzy rank-based ensemble of transfer learning models for emotion recognition from human speech,” IEEE Access, vol. 9, pp. 166518–166530, 2021, doi: 10.1109/ACCESS.2021.3135658.

[18] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, “A systematic evaluation of domain adaptation in facial expression recognition,” 2021, arXiv:2106.15453.

[19] B. Ko, “A brief review of facial emotion recognition based on sequential analysis,” in Proc. Conf. Inf. Commun. Technol. (ICoICT), 2021, pp. 1–17, doi: 10.1109/ICoICT52021.2021.9527512.

[20] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, “A systematic evaluation of domain adaptation in facial expression recognition,” 2021, arXiv:2106.15453.

[21] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep learning in transfer learning,” in Proc. Int. Conf. Brain Inf. Process. (ICBIP), 2018, pp. 270–279, doi: 10.1109/ICBIP.2018.8512931.

[22] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, “Rethinking pre-training and self-training,” 2020, arXiv:2006.06802.

[23] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim, and J. J. Kang, “Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM,” Sensors, vol. 21, no. 8, pp. 2852, 2021.

[24] K. K. Sahoo, I. Dutta, M. F. Ijaz, M. Wozniak, and P. K. Singh, “TLEFuzzyNet: Fuzzy rank-based ensemble of transfer learning models for emotion recognition from human speech,” IEEE Access, vol. 9, pp. 166518–166530, 2021, doi: 10.1109/ACCESS.2021.3135658.

[25] H. Basak, R. Kundu, P. K. Singh, M. F. Ijaz, M. Wozniak, and R. Sarkar, “A union of deep learning and swarm-based optimization for 3D human action recognition,” Sci. Rep., vol. 12, no. 1, pp. 1–17, Dec. 2022, doi: 10.1038/s41598-022-09293-8.

[26] A. Chakraborty, D. Sharma, W. Kim, M. F. Ijaz, and P. K. Singh, “Ensemble-HAR: An ensemble deep learning model for smartphone sensor-based human activity recognition for measurement of elderly health monitoring,” Biosensors, vol. 12, no. 6, pp. 393, Jun. 2022, doi: 10.3390/bios12060393.

[27] S. A.-P. R. Sekaran, C. P. Lee, and K. M. Lim, “Facial emotion recognition using transfer learning of AlexNet,” in Proc. 9th Int. Conf. Intell. Commun. Technol. (ICoICT), Aug. 2021, pp. 170–174, doi: 10.1109/CoICT52021.2021.9527512.

[28] A. V. Savchenko, “Computationally efficient image classification algorithms based on sequential analysis,” in Proc. 12th Int. Conf. Signal Process. (ICSP), Nov. 2009, pp. 267–273, doi: 10.1109/SSPA.2009.5388910.

[29] A. V. Savchenko, “Facial expression and attributes recognition based on multi-task learning of lightweight neural networks,” in Proc. IEEE 19th Int. Symp. Intell. Syst. Informat. (ISIS), Sep. 2021, pp. 119–124, doi: 10.1109/ISIS52375.2021.9582508.

[30] H.-W. Ng, V. D. Nguyen, V. Vonkakis, and S. Winkler, “Deep learning for emotion recognition on small datasets using transfer learning,” in Proc. ACM Int. Conf. Multimed. Interact., Nov. 2015, pp. 443–449, doi: 10.1145/281346.2830593.

[31] A. Gupta and R. Balakrishnan, “Supervised cross-database transfer learning-based facial expression recognition,” in Cybernetics, Cognition and Machine Learning Applications (Algorithms for Intelligent Systems), V. K. Gunjan, P. N. Suganthan, J. Haase, and A. Kumar, Eds. Singapore: Springer, 2021, pp. 273–279, doi: 10.1007/978-981-33-6691-6_30.

[32] B. Ko, “A brief review of facial emotion recognition based on visual information,” Sensors, vol. 18, no. 2, p. 401, Jan. 2018, doi: 10.3390/s18020401.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[34] W. Xie, L. Shen, and J. Duan, “Adaptive weighting of handcrafted feature losses for facial expression recognition,” IEEE Trans. Cybern., vol. 51, no. 5, pp. 2787–2800, May 2021, doi: 10.1109/TCYB.2019.2925005.

[35] C. Huang, K. Hirota, J. Ma, Z. Iia, and Y. Dai, “Facial expression recognition using hybrid features of pixel and geometry,” IEEE Access, vol. 9, pp. 18876–18889, 2021, doi: 10.1109/ACCESS.2021.3054332.

[36] J. F. Goodfellow et al., “Challenges in representation learning: A report on three machine learning contests,” in Natural Information Processing, ICONIP 2013 (Lecture Notes in Computer Science), vol. 8228, M. Lee, A. Hirose, Z. G. Hou, and R. M. Kil, Eds. Berlin, Germany: Springer, 2013, doi: 10.1007/978-3-642-42051-1_16.