Applied Soft Computing 109 (2021) 107491

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Multi-level transfer learning for improving the performance of deep neural networks: Theory and practice from the tasks of facial emotion recognition and named entity recognition

Jason C. Hung, Jia-Wei Chang *

Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taiwan

ARTICLE INFO

Article history:
Received 12 December 2020
Received in revised form 17 April 2021
Accepted 5 May 2021
Available online 10 May 2021

Keywords:
Multilevel transfer learning
Computer vision
Natural language processing
Facial emotion recognition
Named entity recognition

ABSTRACT

Transfer learning has become a promising field in machine learning owing to its wide application prospects. Its effectiveness has spawned various methodologies and practices. Transfer learning refers to improving the performance of target learners in the target domain by transferring the knowledge contained in different yet related source domains. In other words, we can use data from additional domains or tasks to train a model with superior generalization. Using transfer learning, the dependence on considerable target-domain data can be reduced, thereby constructing target learners. Recently, the fields of computer vision (CV) and natural language processing (NLP) have witnessed the emergence of transfer learning, which has significantly improved the most advanced technology on a wide range of CV and NLP tasks. A typical approach of applying transfer learning to deep neural networks is to fine-tune a pretrained model of the source domain with data obtained from the target domain. This paper proposes a novel framework, based on the fine-tuning approach, called multilevel transfer learning (mLTL). Under this framework, we concluded the crucial findings and principles regarding the training sequence of related domain datasets and demonstrated its effectiveness by performing facial emotion and named entity recognition tasks. According to the experimental results, the deep neural network models using mLTL outperformed the original models on the target tasks.

1. Introduction

Pan et al. [1] proposed the concept of transfer learning, in which the existing domain knowledge is used to improve the ability of the target domain to overcome problems related to the scarcity and overfitting of training data. In other words, the previously acquired knowledge is used to learn unknown knowledge to avoid a huge investment in learning an unknown domain, thus reducing excessive resource wastage. A comparison diagram of traditional machine learning versus transfer learning is presented in Fig. 1. The diagram indicates that transfer learning considerably reduces the training time by transferring the parameter weights from a previously established model to an untrained model, and only requires retraining to fine-tune the parameter weights with little data.

Transfer learning can be classified into the following two techniques: feature-based transfer learning and fine-tuning. In feature-based learning, a pretrained model, which refers to an initial model with random weights, is first trained by a representative dataset of a particular field, and once the training process is completed, the model structure and trained parameters are stored. Therefore, feature-based transfer learning uses a pretrained model as a powerful encoder and provides a suitable base for further training of similar tasks. In fine-tuning, the previously saved models and parameters are trained to identify a target dataset, and then the model parameters are updated to achieve the training goal. If the target is identified without following this process, the identification results will be poor, and various training and testing processes will have to be performed, such as adjusting the hyperparameters and fine-tuning the layers, before an appropriate fine-tuning method can be identified.

Therefore, transfer learning has been extensively applied to computer vision tasks. For example, Happy et al. [2] proposed the use of a pretrained VGG-face model [3] for supervised learning with label-smooth models. Four emotion datasets were trained individually by using both labeled and unlabeled data to adjust the model weights, to adjust the hyperparameters during the training process, and finally to fine-tune the last convolution layer and fully connected layer; thus, the model could learn different changes in the expression strength. Ahmed et al. [4] performed face recognition in an uncontrolled environment using

* Corresponding author.
E-mail addresses: jiaweichang.gary@gmail.com (J.-W. Chang), jhung@gm.nutc.edu.tw (J.C. Hung).

https://doi.org/10.1016/j.asoc.2021.107491
1568-4946/

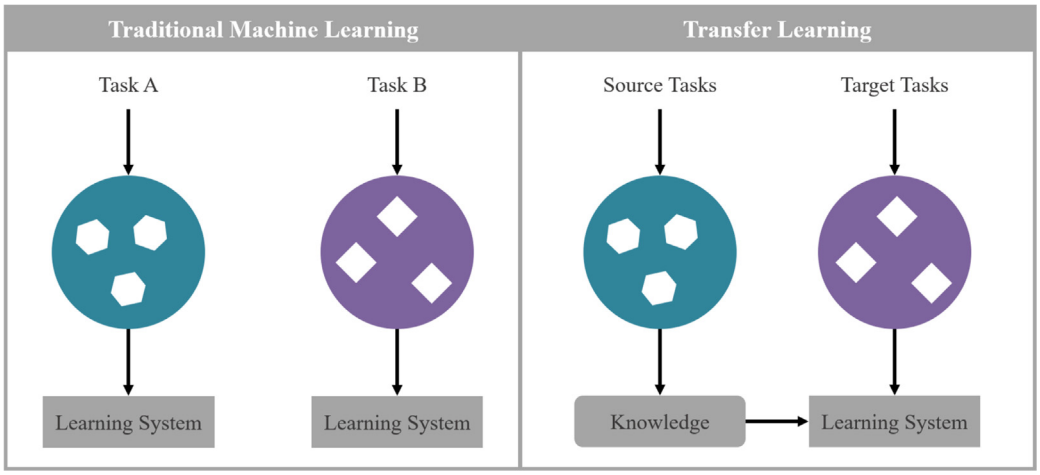

Fig. 1. Comparison of traditional machine learning and transfer learning.

the VGG16 network architecture [5] and pretraining using the ImageNet dataset [6]. The training was performed using the CK+ dataset and the dataset they collected, namely the ITLab dataset. During the training, their proposed incremental active learning method was used for model training and fine-tuning, and the experimental results indicated that both datasets performed well under different light sources, environments, and distances.

In addition, transfer learning is a common and useful technique in the field of natural language processing (NLP). The self-attention mechanism in the transformer allows the Bidirectional Encoder Representations from Transformers (BERT) to model many downstream tasks, such as named entity recognition (NER), sequence or sentence pair classification, question-answering, and sentence tagging. In addition, ELMo is structured to extract context-sensitive functionality from left-to-right and right-to-left and covers the following six baseline tasks: question-answering, textual entailment, semantic role labeling, coreference resolution, named entity extraction, and sentiment analysis.

Hung et al. [7] proposed the Dense_FaceLiveNet model, which is based on a conventional convolutional neural network (CNN) architecture, to offer a solution for learning emotion recognition, which improved the precision and accuracy of the FaceLiveNet network in basic emotion recognition. In this model, transfer learning was applied at two levels. At the first level, transfer learning aimed to learn the relatively simple data of the Japanese Female Facial Expression database and Karolinska Directed Emotional Faces database to model the FER2013 basic emotion dataset. At the second level, transfer learning aimed to transfer the FER2013 basic emotion recognition model to the learning emotion recognition model. The results indicated that the learning emotions model driven by Dense_FaceLiveNet can accurately retain the key action units essential for learning emotions. This demonstrates the effectiveness of deep neural network models using two-level transfer learning in the recognition of learned emotions.

According to Hung [7], increased levels of transfer learning are a promising skill for achieving excellent performance for more downstream tasks in the fields of computer vision (CV) and NLP with limited data. Therefore, this paper proposes a framework based on a fine-tuning approach called multilevel transfer learning (mLTL). The study objectives were as follows: (1) to develop a data analysis method for the datasets with the same task; (2) to discover the principles of using transfer learning with multiple levels; and (3) to validate the effectiveness of the mLTL framework in facial emotion recognition (FER) and NER tasks.

2. Related work

According to Pan et al. [1], transfer learning algorithms can be grouped into the following four categories based on the representation of knowledge to be transferred (i.e., "what to transfer").

2.1. Instance-based transfer learning

As presented in Fig. 2, instance-based transfer learning refers to identifying data in the source domain that are similar to those in the target domain and adjusting their weights, ensuring the new data match the target domain data. Through training learning, a model applicable to the target domain is obtained. Although this method is simple and easy to implement, the choice of weights and the measurement of similarity depend on experience, and the data distributions of the source and target domains often differ.

Tan et al. [8] proposed a transitive transfer learning framework to transfer knowledge from the source domain to an indirectly related target domain by using some intermediate domains. The first step was to identify a suitable domain to connect the given source and target domains. The second step involved an effective knowledge transfer between both domains. The experimental results indicated that the framework yields state-of-the-art classification accuracy on several classification datasets.

2.2. Feature-representation transfer learning

When the source and target domains contain some common crossover features, these features can be transformed to the same space through feature transformation; thus, the source and target domain data in this space exhibit the same data distribution and then perform traditional machine learning. Although this method works well in most scenarios, it is difficult to address and prone to overfitting. This concept is presented in Fig. 3.

While instance-based transfer learning only searches the actual data to obtain data similar to those in the target domain and then learns directly, feature-based transfer learning requires feature transformation to transform the source and target domain data into the same feature space. Long et al. [9] proposed a novel deep adaptation network architecture, which sets the fully connected layers of a deep network to an adaptive layer and focuses on the multiple kernel variant of maximum mean discrepancies. The experimental results indicated that deep learning-based methods are much more effective than traditional shallow transferring methods.

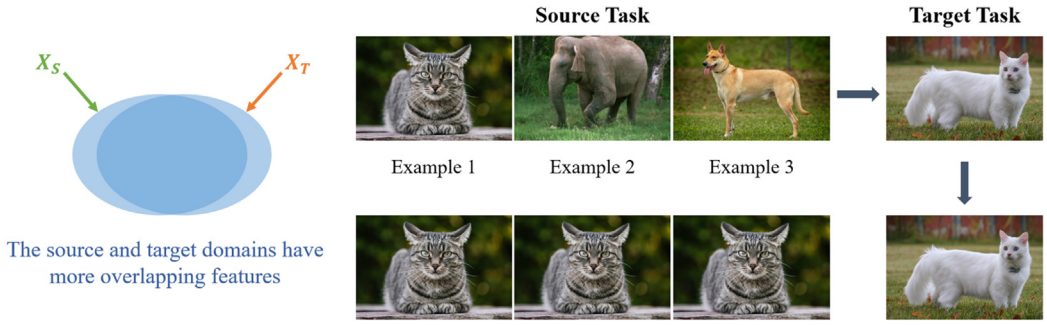

Fig. 2. Concept diagram of instance-based transfer learning.

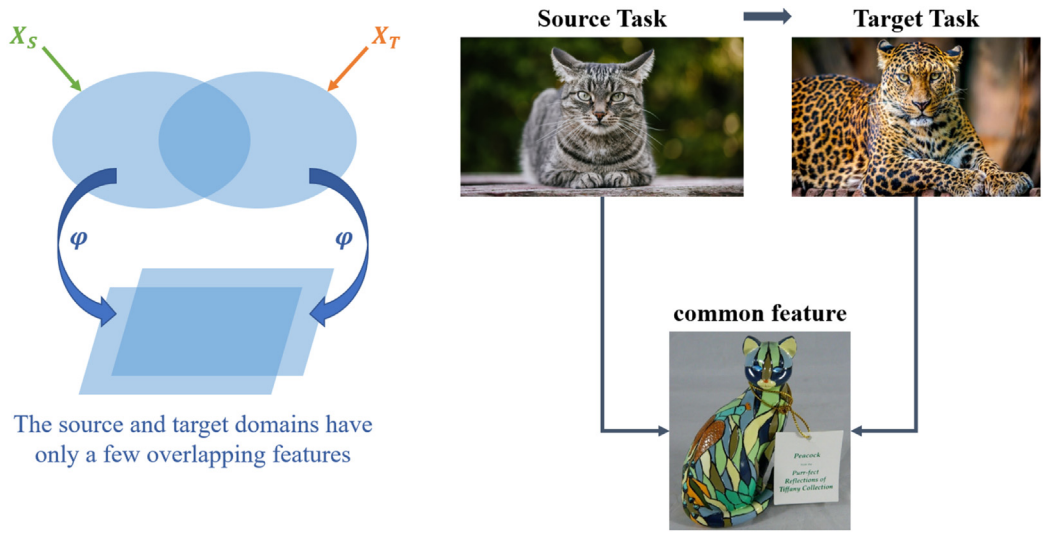

Fig. 3. Concept diagram of feature-representation transfer learning.

2.3. Parameter transfer learning

As indicated in Fig. 4, in parameter transfer learning, the target and source domains share the same model parameters or follow the same prior distribution. In other words, the model previously trained in the source domain with considerable data is applied to the target domain for prediction. Although it is relatively straightforward to exploit the similarities that exist between models, the model parameters do not easily converge.

Ge et al. [10] proposed a new framework of online multiple source transfer learning (MSTL). An offline MSTL method combines the knowledge obtained from the source and target domains using convex optimization. By contrast, an online MSTL method is developed based on the optimization framework, and performs better than the offline method. The results of the experiments conducted on the datasets of CAD Prediction, Email Spam Filtering and Intrusion Detection indicated that online MSTL is a fast and scalable algorithm.

2.4. Relational-knowledge transfer learning

Assuming that the data of the source and target domains exhibit the same logical network relations, these domains share some similar relations. The logical network relations learned in the source domain can be applied to the target domain to perform the transformation. A typical method of relational-knowledge transfer learning is the mapping method, such as the transformation from a biological virus propagation pattern to a computer virus propagation pattern and that from the teacher–student relation to the supervisor–subordinate relation. The concept diagram of this type of transfer learning is presented in Fig. 5. Mihalkova et al. [11] presented the SR2LR algorithm, which is designed for situations in which the target-domain data can only be provided as a single-entity-centered example. This algorithm evaluates the possible source-to-target predicate correspondences based on short-range clauses so that the knowledge captured in long-range clauses can also be transferred.

Table 1
The four methods of transfer learning.

Method	Inductive transfer learning	Transudative transfer learning	Unsupervised transfer learning
Instance-based	✓	✓	
Feature-representation	✓	✓	✓
Parameter	✓		
Relational-knowledge	✓		

In the first three types of transfer learning, the data are required to be independent of the distribution assumptions. Moreover, all four types of transfer learning require the selected source domain to be related to the target domain. Table 1. summarizes the four methods.

In the literature [1,12], several categorization criteria have been proposed for transfer learning. Transfer learning problems can be classified into the following three categories: transudative, inductive, and unsupervised. Zhuang et al. [12] approached the topic from data and model perspectives and illustrated more than 40 representative approaches to transfer learning. This survey interprets transfer learning approaches from the data and

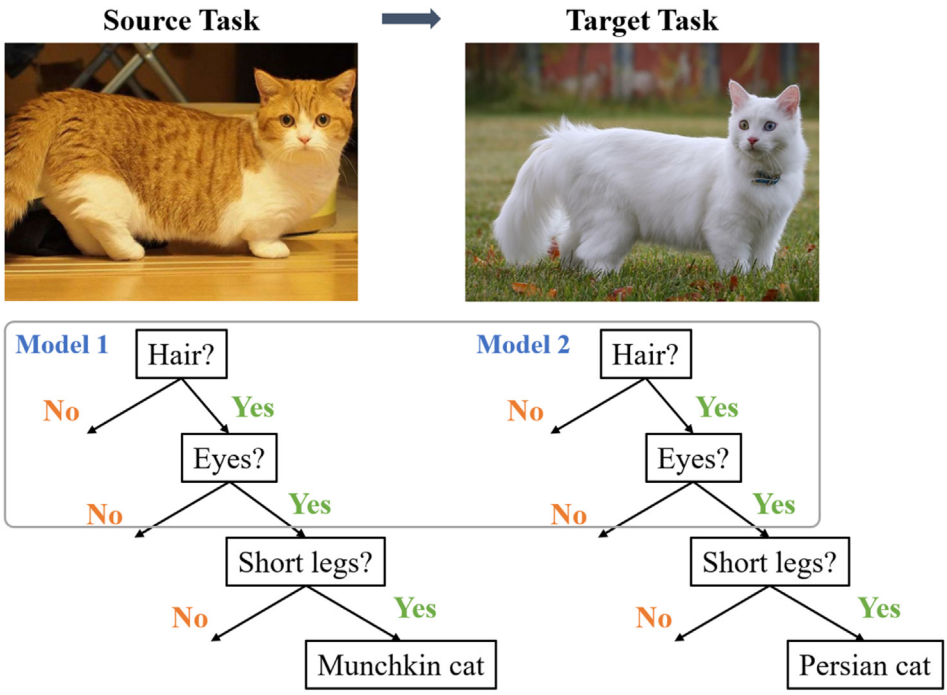

Fig. 4. Concept diagram of parameter transfer learning.

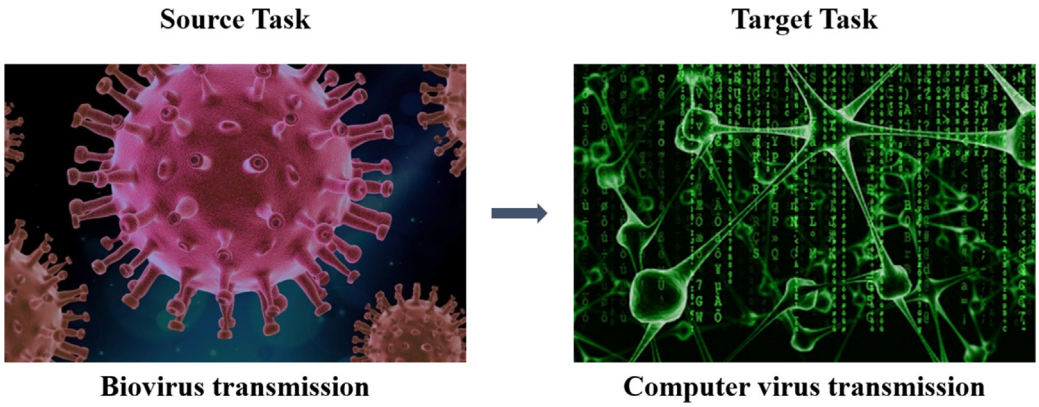

Fig. 5. Concept diagram of relational-knowledge transfer learning.

model perspectives [12], which refer to data- and model-based interpretation, respectively. Data-based interpretation focuses on transferring the knowledge through the adjustment and transformation of data, including instance- and feature-based transfer learning. By contrast, model-based interpretation aims to obtain accurate prediction results on the target domain. In addition, transfer learning approaches were introduced not only to focus on general text- and image-related applications but also to be applied to specific areas, such as medicine, bioinformatics, transportation, and recommender systems. To present the performance of different migration learning models, they conducted experiments using 20 representative migration learning models, which were executed on three datasets. The experimental results indicated the importance of selecting the right migration learning model for practical applications.

In this era of large pretrained models, fine-tuning can be considered a normal operation. Fine-tuning is a very costly operation, although it generally produces accurate results. As mentioned above, two main techniques exist for performing transfer learning in NLP, namely feature-based and fine-tuning. Houlsby et al. [13] proposed an alternative approach called an adapter module. Both feature-based and fine-tuning approaches require a new set of weights to be trained for each task, whereas the adapter approach can use parameters more efficiently by simply training the parameters within the module. The adapter tuning approach for NLP exhibits three main features. It achieves suitable performance, there is no requirement for handling different datasets simultaneously, and only a few additional parameters are required for each new task. When using deep learning models, fine-tuning is the most commonly applied transfer learning method. However, there are two major drawbacks to the commonly used fine-tuning approach: (1) overfitting may occur when the target dataset is small and the parameters of the pretrained network are excessive; and (2) the initial number of frozen layers must be manually selected, and thus, the optimization efficiency cannot be improved.

3. Methodology

The mLTL framework is illustrated in Fig. 6. Transfer learning is used to resolve the problem of insufficient training data. In the figure, N is the total number of datasets with the same

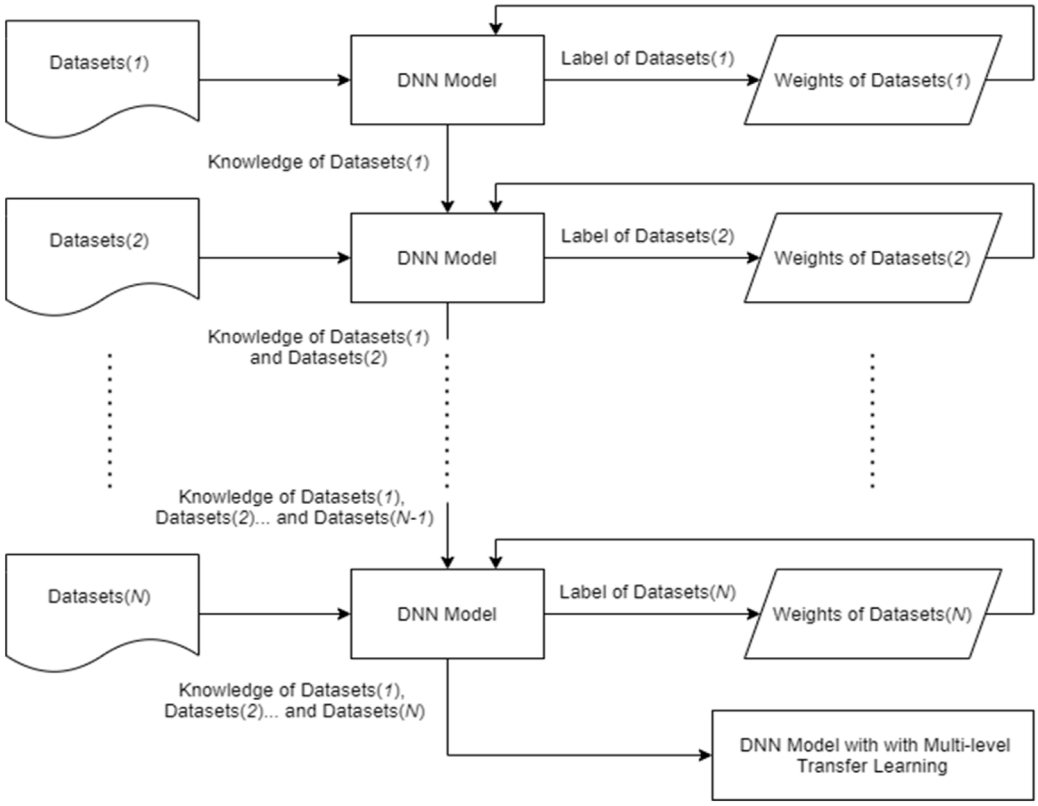

Fig. 6. Framework of multi-level transfer learning.

task and is also defined as the total number of corresponding transfer learning levels, because each dataset is learned at the corresponding transfer learning level. When transferring a trained neural network model to the Nth dataset, its output layer must be replaced by the output labels corresponding to the Nth dataset.

After achieving the optimal performance in each dataset (i.e., transfer learning level), the trained model parameter weights are saved for transfer learning of the next dataset. Because this model does not use the fully connected layer as its last layer, features learned from the previous dataset can be relocated to the next dataset and the number of nodes in the final output layer is replaced with the number of datasets that contain features, allowing more features to be collected.

Definition 1. In formula (1), T_m^n is a certain combination of transfer learning, n is the number of datasets used for feature transferring, and m is the level of transfer learning:

$$T_m^n = \begin{cases} m! \times \binom{n}{m}, \text{if } n > m > 0 \\ (m-1)! \times \binom{n}{m-1}, \text{if } n = m \end{cases} \quad (1)$$

Example 1. T_1^3 indicates that THREE datasets are trained for feature transfer and performing one-level feature transfer. $T_1^3 = 1! \times \binom{3}{1} = 3$ means that we can perform THREE combinations for this transfer learning.

Example 2. T_2^3 indicates that THREE datasets are trained for feature transfer and performing two-level feature transfer. $T_2^3 = 2! \times \binom{3}{2} = 6$ means that we can perform SIX combinations for this transfer learning.

Example 3. T_2^2 indicates that TWO of datasets are trained for feature transfer and performing two-level feature transfer. $T_2^2 = (2-1)! \times \binom{2}{2-1} = 2$ means that we can perform TWO combinations for this transfer learning.

To validate the proposed mLTL framework, we applied it to address the CV and NLP tasks. For the CV task, we used Dense_FaceLiveNet for FER, and for the NLP task, we used DistiBERT for NER. In the following subsections, we present Dense_FaceLiveNet and DistiBERT in detail.

3.1. Using Dense_FaceLiveNet model for FER

Fig. 7 illustrates Dense_FaceLiveNet, which is a convolutional neural network structure of the gaming FER model introduced by Hung et al. [7]. Its architecture combines the architecture of DenseNet proposed by Huang et al. [14] and that of FaceLiveNet proposed by Ming et al. [15]. The design concept of DenseNet connects each inception layer. Based on FaceLiveNet, the following three improvements were made in Dense_FaceLiveNet.

3.1.1. Replacing fully connected layers with global average pooling

Fully connected layers aim to use the results of the convolutional and pooling layers to classify the image into a label. The main problem with fully connected layers is the excessive number of parameters, which can not only easily lead to overfitting but also prevent the generalization ability of the overall neural network. Dense_FaceLiveNet refers to the concept proposed in [16], according to which, instead of adopting fully connected layers for classification, the average of each feature map is taken and the resulting vector is fed into the softmax layer. One advantage of global average pooling over the fully connected layers is that there are no optimized parameters in global average pooling, which prevents overfitting in this layer.

3.1.2. Replacing residual blocks with dense blocks

Dense_FaceLiveNet uses two layers of dense blocks, where the interconnected structure is called dense inception blocks. Finally, a translate layer is added to reduce the number of feature dimensions. Dense blocks are used instead of residual blocks because DenseNet exhibits greater recognition accuracy than ResNet on ImageNet [14]. All layers are connected; thus, each layer obtains additional inputs from all previous layers and passes on its feature maps to all subsequent layers.

3.1.3. Replacing ReLU with Swish as model activation functions

Swish, proposed by Google Brain [17], is an activation function similar to the rectified linear unit (ReLU), as both are smooth and nonmonotonic functions. On ImageNet, by simply replacing ReLUs with Swish units, the accuracy is improved by 0.9% and 0.6% for Mobile NASNet-A [18] and Inception-ResNet-v2 [19], respectively.

3.2. Using DistilBERT for NER

Transfer learning is becoming increasingly common with large-scale pretrained models. BERT, a language model proposed by Google AI Language [20], is the most advanced model developed in NLP to date. However, under constrained computational training or inference budgets, operating these large models remains challenging. For this reason, much research has been conducted on how to compress these large-scale pretrained models. Using DistilBERT [21], Sanh et al. indicated that a much smaller language model pretrained with knowledge distillation can achieve similar performance on numerous downstream tasks. As DistilBERT reduces the BERT model by 40%, retains 97% of the language understanding capabilities, and is 60% faster than BERT, it is considered an ideal alternative for edge applications.

Knowledge distillation [22] is a compression technique that can be understood as a student–teacher relation, where the purpose is to allow the student to achieve the same abilities as those of the teacher to the maximum possible extent. The teacher and student refer to BERT and DistilBERT, respectively, whose structures are presented in Fig. 8. According to the figure, the two structures appear to be similar, with the only difference lying in the number of layers and the hidden size.

4. Experiments

4.1. FER and its experimental datasets

In 1995, Picard introduced the concept of affective computing [23], which uses computer technology to recognize human emotions and explore the implicit emotion recognition techniques and applications. This method analyzes the information obtained from sensors, such as skin temperature [24], electroencephalogram data [25], and facial expressions, and understands the corresponding emotions through machine learning. According to the rule of Mehrabian [26], visual communication forms the most crucial part of emotional communication. This is why facial expressions are considered the main approach to understanding emotions. Previous studies [27,28] have divided facial expressions into two groups, namely basic and complex emotions. Basic emotions include sadness, anger, fear, disgust, joy, and surprise, which are associated with commonly recognizable facial expressions. By contrast, complex emotions are socialized emotions learned through acquired learning; for example, different people may react differently in the same situation, such as grief, regret, flow, and jealousy.

The traditional face recognition classification algorithms for machine learning include support vector machine (SVM) [29], rule-based methods [30], principal component analysis [31], and template matching methods [32]. Features are represented as action units by the facial action coding system (FACS) [33]. However, it is difficult to mark facial features because high image quality is required; thus, FER faces a bottleneck. A characteristic of the deep learning approach is that it does not require feature engineering and can learn the features on its own from the available data. CNNs, such as DenseNet [14] and FaceLiveNet [15], exhibit suitable recognition accuracy in the image processing of basic emotions.

Table 2
Emotional labels and quantity of each label of the four FER datasets.

	CK+	FER2013	LE	GFE2019
Angry (An)	135	4953		42
Disgust(Di)	177	547		
Fear(Fe)	75	5121		
Delightful (De)	207	8989	36	363
Sad (Sa)	84	6077		19
Surprised (Su)	249	4002	8	46
Neutral or Flow(Fl)		6198	1570	563
Frustration (Fr)			7	89
Confused (Co)			82	58
Boredom (Bo)			1477	
Excitation (Ex)				95
Total	927	35,887	3180	1275

In the mLTL experiment, the following four FER datasets were used: CK+ [34], FER2013 [35], learning emotion (LE) [7], and Gaming Facial Emotion 2019 (GFE2019). Table 2 presents the emotional labels and quantity of each label for the four FER datasets.

1. **CK+** is a basic emotion dataset and an extension of the Cohn–Kanade (CK) dataset. Compared with CK, CK+ exhibits increases of 22% and 27% in the numbers of sequences and subjects, respectively. The target expressions for each sequence are encoded with FACS and the emotion labels are revised and validated.
2. **FER2013** is a basic emotion dataset, which comprises grayscale images of faces with different angles, lighting, genders, and ethnicities. These images conform to real-life situations—where the faces have more subtle expressions, some of which are difficult to distinguish with the human eyes.
3. **LE** is a dataset of original video data collected by four students from the Department of Information Management at National Chung Hsing University, Taiwan. These students captured the images by watching free videos from YouTube, VoiceTube, and other famous online video platforms and recorded the original video in 3 s intervals. The categories of the collected data included frustration, confused, boredom, delightful, surprised, and neutral/flow.
4. **GFE2019** The data collected for this experiment contain 1275 images of consulting professional e-sports coaches and gamers before defining the game emotions. The dataset contains eight main game facial expressions, with four positive and four negative emotions. The positive emotions include delightful, excitation, surprised, and neutral/flow, whereas negative emotions include angry, confused, sad, and frustration.

4.2. NER and its experimental datasets

NER aims to identify specific types of object names from unstructured text (e.g., persons, locations, organizations, times, and dates). It forms the basis for NLP tasks, such as knowledge graphs, machine translation, and question-answering systems. Recent

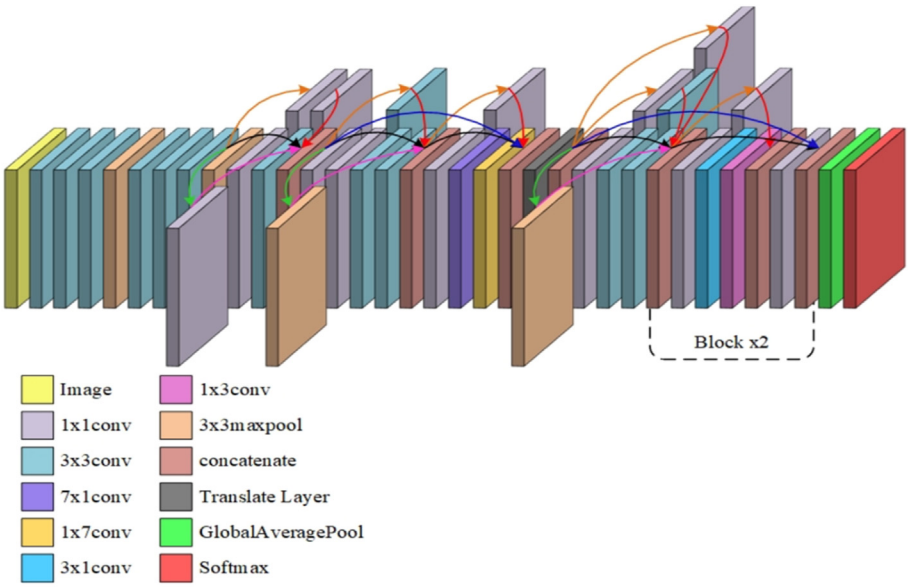

Fig. 7. Architecture of Dense_FaceLiveNet.

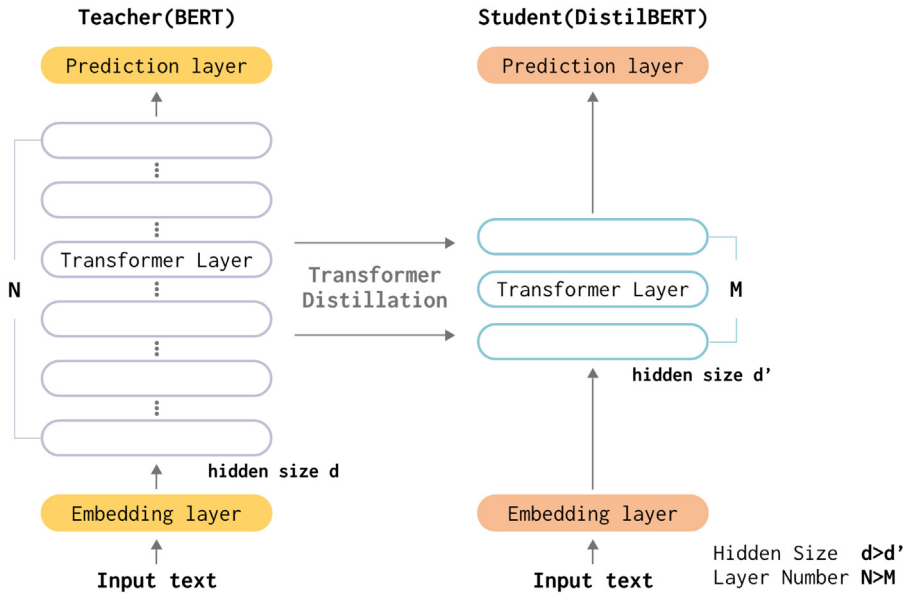

Fig. 8. Architectures of BERT and DistilBERT.

studies have used machine learning methods, such as conditional random fields [36], maximum entropy [37], SVM [38], and recurrent neural network (RNN) [39] language models for NER tasks. In 2018, the Google AI Language Lab proposed BERT [20], which can handle multiple NLP tasks simultaneously. Instead of RNNs, BERT uses an approach that is entirely based on the attention mechanism [40] and addresses the fundamental constraint of sequential computation.

In the mLTL experiment, the following four Chinese NER datasets were used: MSRA [41], WeiboNER [42], OntoNotes4.0 [43], and Resume [44]. Table 3 presents the statistics of the entity labels and informative tokens of these datasets, and Table 4 introduces examples and further information of each of their entity labels.

1. **Weibo NER**: 1890 texts were selected from Weibo from November 2013 to December 2014 and tagged with four terms: person, location, organization, and geopolitical.
2. **OntoNotes 4.0** is a large language corpus dataset, which supports the named entity dataset in Chinese, English, and Arabic from a wide range of sources, including news, radio conversations, phone conversations, and weblogs.
3. **MSRA** is a Chinese named entity corpus dataset provided by Microsoft Research Asia in 2006, which includes named entity tags such as personal name, country and region, and organization and has been extensively cited and extended.
4. **Resume** is a dataset of 1,027 resume summaries of senior executives from Sina Finance provided by Zhang et al. [44] in the ACL 2018 conference presentation. This dataset contains eight types of named entities: country, location, personal name, organization, major, educational institutions, ethnicity, and job title.

Table 3
Statistics of entity labels and informative tokens of the four datasets.

	Labels	Number of tokens	Number of none labels	Number of defined labels	Ratio of none labels (%)	Ratio of defined labels (%)
OntoNote 4.0	4	491,903	450,700	41,203	91.62	8.38
MSRA	3	1,955,827	1,727,930	227,897	88.35	11.65
Resume	8	124,099	45,085	79,014	36.33	63.67
Weibo NER	8	73,728	68,777	4951	93.28	6.72

Table 4
Examples and information of entity labels for the four NER datasets.

Name	Label abbreviation	Label name	Examples
MSRA	NR	Personal Name	梁實秋 (Liang Shiqiu)
	NS	Country and Region	歐美 (Europe and America)
	NT	Organization	北大 (Peking University)
OntoNote 4	GPE	Country	菲律賓 (Philippines)
	LOC	Location	中東 (Middle East)
	ORG	Organization	委員會 (Council)
	PER	Personal Name	梁實秋 (Liang Shiqiu)
Weibo NER	GPE.NAM	Country (Specific)	美國 (USA)
	GPE.NOM	Country (Substitute)	國家 (Country)
	LOC.NAM	Location (Specific)	辦公室 (offices)
	LOC.NOM	Location (Substitute)	街道 (Street)
	ORG.NAM	Organization (Specific)	高通 (Qualcomm)
	ORG.NOM	Organization (Substitute)	學校 (School)
	PER.NAM	Personal Name (Specific)	宋院長 (President Song)
	PER.NOM	Personal Name (Substitute)	粉絲 (Fans)
Resume	CONT	Country	美國 (USA)
	EDU	Educational Institutions	大學學歷 (University Degree)
	LOC	Location	廣東 (Guangdong Province)
	NAME	Personal Name	梁實秋 (Liang Shiqiu)
	ORG	Organization	公司 (Company)
	PRO	Major	經濟管理學 (Economics and Management)
	RACE	Ethnicity	漢族 (Han ethnicity)
	TITLE	Job Title	高級工程師 (Senior Engineer)

5. Results

In the experimental results, the performance measures are defined as follows:

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \tag{2}$$

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{3}$$

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{TN} + \text{FN}) \tag{4}$$

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall}) \tag{5}$$

TP, TN, FP, and FN stand for True Positive (the number of pairs correctly labeled as paraphrases), True Negative (the number of pairs correctly labeled as non-paraphrases), False Positive (the number of pairs incorrectly labeled as paraphrases), and False Negative (the number of pairs incorrectly labeled as non-paraphrases), respectively.

5.1. Experimental results of FER

To address the problem of training data scarcity, the experiment conducted in this study used data augmentation to rotate an image randomly by -5° to 5°, flip it horizontally, and zoom randomly into it by 1–1.5 times, which considerably increases the amount of training data in the original dataset. K-Folds cross-validation was used as a training model to avoid overfitting of the training model, and the result was obtained by averaging the results of each training session. As the base model of transfer learning, we set a learning rate of 0.01, a batch of 64, an epoch of 50 times, and a patience value of 10 for early stopping skill to Dense_FaceLiveNet. As GFE2019 has the largest number of categories and a high application value, it is used as the target dataset for mLTL, and CK+, FER2013, and LE are arranged in different combinations. Each dataset is abbreviated to an uppercase letter as follows:

"C" for the CK+ dataset,
"F" for the FER2013 dataset,
"L" for the LE dataset, and
"G" for the GFE2019 dataset.

As presented in Table 5, the accuracy of the GFE2019 emotion recognition model without transfer learning was 71.53% and training was performed using mLTL to further improve the accuracy. In addition, Tables 6–8 present one-, two-, and three-level transfer learning training combinations and the experimental results, respectively.

Compared with the model without transfer learning, that with transfer learning exhibited improved accuracy, as indicated in Table 3, with the largest improvement of 10.09% observed for F→G. As presented in Table 4, the combination exhibiting the largest improvement in accuracy (13.96%) after two-level transfer learning was F→L→G. The worst result was obtained for the F→C→G combination; however, the accuracy was improved compared with that achieved without transfer learning. The results obtained after the third transfer learning are presented in Table 5, and the best model was built in the migration order of

Table 5
Dense_FaceLiveNet [7] without transfer learning (GFE2019 only).

	Average accuracy (%)	Convergence epochs	Average training time (s)
G	**71.53**	30	105

Table 6
Combinations for one-level transfer learning in FER.

	Average accuracy (%)	Convergence epochs	Average training time (s)
C→G	77.57	23	100
F→G	**82.43**	21	86
L→G	76.39	17	94

Table 7
Combinations for two-level transfer learning in FER.

	Average accuracy (%)	Convergence epochs	Average training time (s)
C→F→G	82.98	17	86
C→L→G	76.61	19	95
F→C→G	74.90	4	62
F→L→G	**85.49**	24	105
L→C→G	77.65	11	79
L→F→G	83.14	16	90

Table 8
Combinations for three-level transfer learning in FER.

	Average accuracy (%)	Convergence epochs	Average training time (s)
C→F→L→G	**87.84**	19	97
C→L→F→G	86.67	22	102
F→C→L→G	83.92	15	112
F→L→C→G	83.53	9	74
L→C→F→G	87.45	22	103
L→F→C→G	85.88	15	87

C→F→L→G. Compared with the best models in Tables 5–8, the overall accuracy improved by 16.31% (G, 71.53% to C→F→L→G, 87.84%), 5.41% (F→G, 82.43% to C→F→L→G, 87.84%), and 2.35% (C→F→G, 85.49% to C→F→L→G, 87.84%), respectively. In Table 8, the accuracy of all six combinations was above 80% by three-level transfer learning.

5.2. Experimental results of NER

In this study, we trained NER on the MSRA, Weibo NER, OntoNotes4.0, and Resume datasets and learned their implicit knowledge using the mLTL approach. Because the Resume dataset has more entity tags from a broad perspective and is more useful for applications, and therefore, it was used as the target dataset for transfer learning. In addition, different combinations of MSRA, Weibo NER, and OntoNotes 4.0 were arranged for training, where each dataset was abbreviated to an uppercase letter as follows:

"M" for the MSRA dataset,
"W" for the Weibo NER dataset,
"O" for the OntoNotes 4.0 dataset, and
"R" for the Resume dataset.

In the experiments, we adopt DistilBERT model as the base of transfer learning. Because DistilBERT model is similar to BERT model, we used a learning rate of 3e-5, a batch of 10, and an epoch of 10 times, as suggested in the original BERT [20,21]. In addition, we adopted the early stopping skill, whose value of patience is 3. Table 9 indicates that all combinations of the three-level transfer learning exhibited accuracy superior to the original DistilBERT without transfer learning. In particular, DistilBERT (O→M→W→R) achieved the best performance with an F1-Score of 93.49%.

Table 9
Results of DistilBERT [21] with/without mLTL for NER.

	Precision (%)	Recall (%)	F1-Score (%)
R	66.38	64.49	65.42
M→W→O→R	92.29	93.88	93.07
M→O→W→R	92.66	94.24	93.44
W→M→O→R	71.11	67.89	69.46
W→O→M→R	92.10	92.30	92.19
O→M→W→R	92.28	94.75	**93.49**
O→W→M→R	76.53	69.64	72.92

Table 10
Comparisons of NER models on resume dataset.

Model	Precision (%)	Recall (%)	F1-Score (%)
Lattice LSTM [44] (R)	94.81	94.11	94.46
Glyce+BERT [45] (R)	96.62	96.48	96.54
DistilBERT [21] (R)	66.38	64.49	65.42
DistilBERT by mLTL framework (O→M→W→R)	92.28	94.75	93.49

Table 11
Comparison of prediction speeds of NER models.

	Glyce+BERT	Lattice LSTM	DistilBERT
Seconds per token	0.001788	0.010463	0.000598
Total seconds	27	158	9

As indicated in Table 10, the metrics for the Resume dataset were low before performing mLTL, and those for DistilBERT (O→M→W→R) were close to the performance of the lattice LSTM model. The results of the Glyce+BERT method proposed by Meng et al. [45] indicated that the F1-Score of the Resume dataset was 96.54%, which is a state-of-the-art approach. However, Glyce+BERT was a model trained with several parameters, and it thus had a slower execution. In addition, the DistilBERT model outperformed the BERT model in terms of the number of training parameters, and thus, it had the disadvantage of slower execution. Furthermore, the DistilBERT model outperformed the BERT model in terms of speed. As indicated in Table 11, the DistilBERT model's prediction time of 9 s was superior to the Glyce+BERT model's prediction time of 27 s and significantly superior to the lattice LSTM model's prediction time of 158 s. In addition, the DistilBERT model predicted a token 0.000596 s faster than the Glyce+BERT model, and the lattice LSTM model exhibited the worst prediction performance. Therefore, DistilBERT outperformed Glyce+BERT in the real-world application field.

6. Discussion

6.1. Similarity of labels between datasets in FER and NER tasks

In this study, the number of datasets used for both NER and FER tasks in mLTL was four. The FER datasets were CK+, FER2013, LE, and GFE2019, as indicated in Table 2, and the NER datasets were MSRA, OntoNote 4, Weibo NER, and Resume, as indicated in Table 4.

Of the four FER datasets, CK+ and FER2013 are similar and LE and GFE2019 are similar. The CK+ and FER2013 datasets are basic emotion datasets, with the only difference being that CK+ has six categories and FER2013 has seven categories. Meanwhile, both the LE and GFE2019 datasets belong to more complex mood categories and have more overlapping categories; however, the GEF2019 dataset contains more categories. Therefore, this study used the GFE2019 dataset as the target dataset in mLTL.

Of the four NER datasets, the labels in Weibo NER, MSRA, and OntoNote 4.0 are highly similar in concept but the granularity of

Table 12
Further analysis of O→M→W→R and M→O→W→R in terms of two-level transfer learning.

	Precision (%)	Recall (%)	F1-Score (%)
O→M→W	49.35%	51.25%	50.28%
M→O→W	45.25%	45.98%	45.61%

the labels is quite different. There are additional location labels on OntoNote 4 and Weibo NER. Both Weibo NER and Resume are datasets with numerous labels and high granularity. Compared with other datasets, Resume has up to eight tag types – CONT, EDU, LOC, NAME, ORG, PRO, RACE, and TITLE – which are much broader in terms of categories and more meaningful for practical applications. Therefore, this study used the Resume dataset as the target dataset in mLTL.

6.2. Similar findings and usage principles from NER and FER tasks

Transfer learning has been proven effective in improving the learning performance of target learners in the target domain. This paper presents the mLTL framework, which is based on fine-tuning. Based on this framework, individual experiments were conducted for the FER and NER tasks. In the two experiments, we obtained two findings and derived the following common principles of mLTL.

1. **Results obtained using mLTL were superior to those obtained without transfer learning:**
   According to the results obtained for both FER and NER experiments, the accuracy obtained with mLTL was superior to that obtained without transfer learning. As indicated in Table 5, the average accuracy of GFE2019 without mLTL was 71.53% for the FER task. Compared with the results of mLTL experiment (Table 8), the worst combination was F→L→C→G, which had an average accuracy of 83.53%, which was 12% higher than that of the GFE2019 dataset without transfer learning. For the NER task, as indicated in Table 9, the F1-Score of the Resume dataset without mLTL was 65.42%, and the worst combination, W→M→O→R, had a higher F1-Score of 69.46% than that of the Resume dataset without transfer learning.

2. **In mLTL transition, the different sequential combinations of datasets might affect the final result of the target dataset:**
   - On the NER task, as indicated in Table 9, the combinations that performed well were O→M→W→R and M→O→W→R. Although their F1-Scores were similar, O→M→W→R obtained slightly better results than M→O→W→R. As indicated in Table 12, comparing the results of O→M→W and M→O→W, the F1-Score of O→M→W was 50.28%, which is better than the 45.61% of M→O→W. We speculate that O→M→W→R outperformed M→O→W→R in the subsequent transfer to the Resume dataset because of the superior performance of O→M→W. On the FER task, as indicated in Table 8, the combinations that performed well were C→F→L→G and L→C→F→G with accuracies of 87.84% and 87.45%, respectively. Although the two accuracies were similar, C→F→L→G performed slightly better than L→C→F→G. Comparing the results of F→L→G and C→F→G experiments, as indicated in Table 7, F→L→G's accuracy was 85.49%, which is higher than C→F→G's accuracy of 82.98%. We speculate that the superior performance of F→L→G enabled C→F→L→G to outperform L→C→F→G.
   - For the NER task, as indicated in Table 9, the two poor combinations were W→M→O→R and O→W→M→R, with F1-Scores of 69.46% and 72.92%, respectively. Their common ground is the inclusion of W→M, presumably because the poor learning in W→M affected the final results. For the FER task, as indicated in Table 8, the two poor combinations were F→C→L→G and F→L→C→G. The accuracies of L→C→G and C→L→G were 77.65% and 79.61%, respectively. Presumably, the poor performance of L→C→G made the F→L→C→G results even worse than the F→C→L→G results.

Based on the aforementioned two findings, we deduced the following principles of use for improving the results of mLTL.

1. **A dataset with considerable data can help learn better features in mLTL:**
   - For the FER task, the results of one-level transfer learning (listed in Table 6) indicate that the average accuracy of the FER2013 transfer to GFE2019 was 82.43%, which was the best among the three migration combinations. Table 2 indicates that the FER2013 dataset is the largest of all datasets. We speculate that a large dataset helps to learn a wider variety of features in transfer learning training. By contrast, C→G and L→G had lower accuracies of 77.57% and 76.39%, respectively. We speculated that, for both combinations, the scarcity of data results in insufficient features being learned during migration to the GFE2019 dataset. For the NER task, similar findings are presented in Table 12. First, the OntoNotes 4.0 dataset is smaller than the MSRA dataset (Table 3). For the Weibo NER dataset, MSRA contributed more than OntoNotes 4.0 from the O→M→W sequence than from the M→O→W sequence.

2. **Placing a large dataset in the later order can yield superior results:**
   - For the NER task, as indicated in Table 9, O→M→W→R and M→O→W→R performed better, and they share the inclusion of the WR order in common. Table 3 indicates the transfer from the smallest of the four datasets, Weibo NER, to a larger one, the Resume dataset, where Weibo NER has only 73,728 tokens and 4,951 non-O tags, whereas the Resume dataset has 124,099 tokens and 79,014 non-O tags. For the FER task, F→C→L→G and F→L→C→G combinations with a large dataset trained first were less effective than the other combinations, as indicated in Table 8. The common ground here was that FER2013, which has considerable data, was placed first for training (Table 2). Therefore, in mLTL, larger datasets are not suitable for training at the forefront.

3. **Placing similar datasets closer can improve the training results:**
   - For the FER task, as indicated in Table 8, the combinations C→F→L→G and L→C→F→G obtained superior accuracies of 87.84% and 87.45%, respectively. Based on Table 2, we assumed the reason to be that both CK+ and FER2013 are basic mood datasets and have the closest number of categories. For the NER task, as indicated in Table 9, superior results were achieved by O→M→W→R and M→O→W→R, with F1-Scores of 93.49% and 93.44%, respectively. According to Table 3, a possible reason was that the Weibo NER and Resume datasets have the same numbers of categories and labels. Therefore, we inferred that a similar number of categories can improve the results.
   - For the FER task, as indicated in Table 7, the C→L→G and L→C→G results were worse than those for other combinations. According to Table 2, CK+ and LE were the basic

and complex emotion datasets, respectively. In both cases, the category similarity between the two datasets was low and the amount of data was small; in the LE dataset in particular, the problem of category imbalance was evident. For the NER task, as indicated in Table 9, the F1-Scores of W→M→O→R and O→W→M→R were worse than those of the other combinations. According to Table 3, both combinations contain the order of W→M, and the Weibo NER dataset contains eight labels, whereas the MSRA dataset only has three labels. Therefore, we conjectured that when the Weibo NER dataset is transferred to the MSRA dataset, it has too many labels for MSRA to learn effectively.

In the previous sections, we discussed the size of the datasets and the similarity of the labels. For example, as indicated in Table 8, C→F→L→G and L→C→F→G obtained better results than the other combinations. This is consistent with Principles 2 and 3. However, if the conflict between the two principles is ambiguous, such as satisfying Principle 2 but not fitting a category similar to the requirement to put it together (Principle 3), we performed further analysis and obtained Principle 4.

4. **Placing the larger dataset in the front order if the two datasets are not similar:**

- For the FER task, a conflict existed between Principles 2 and 3. As indicated in Table 8, when FER2013 was transferred to CK+ within the sequence of F→C→L→G, the results were unsatisfactory. This sequence conformed to Principle 3 and placed similar categories together, but did not conform to Principle 2. For the NER task as well, a conflict existed between Principles 2 and 3, as indicated in Table 9, where W→M→O→R and O→W→M→R yielded poor results even though the order of W→M satisfied Principle 2. According to Table 3, the Weibo NER dataset contains eight labels, whereas the MSRA dataset has only three labels; moreover, even the label of the Weibo NER dataset has a higher granularity, and therefore, it does not meet Principle 3.
- For the NER task, according to Table 9, M→W→O→R and O→M→W→R performed better than the other combinations, which neither satisfied Principle 2 nor Principle 3. For the FER task, according to Table 7, the best combination was F→L→G, which did not satisfy Principles 2 and 3, and even FER2013 is a basic emotion dataset and LE is a complex emotion dataset. From the aforementioned conflict, we instead determined that if the two datasets are not similar, the larger dataset should be placed in front for transfer learning.

Based on the aforementioned observations, for dataset sorting combinations, one should first put together those with high category similarity and then adjust them according to the dataset size, placing those with large datasets behind. However, if the two datasets are not similar, placing the larger dataset in front will help the later dataset to learn more features in the transfer learning process.

7. Conclusion

This paper proposed the mLTL framework based on the fine-tuning method, this framework can be used when several pre-trained models or datasets related to the target task are earned. To validate the effectiveness of mLTL framework, we conducted separate experiments for FER and NER tasks and obtained the following results.

- For the FER task, the original accuracy of Dense_FaceLiveNet on GFE2019 was 71.53%, and the model with three-level transfer learning achieved the optimal accuracy of 87.84%.
- For the NER task, the original F1-Score of DistilBERT on Resume was 65.42% only, and the model with three-level transfer learning achieved the optimal F1-Score of 93.49%.

The results of the two experiments indicated that Dense_FaceLiveNet and DistilBERT with mLTL performed better than the original databases without mLTL, proving that mLTL is highly beneficial for deep neural network models. In addition, we discussed the results of the mLTL framework for the FER and NER tasks and further concluded two important findings and four principles for mLTL usage. Based on the findings of this study, we believe that mLTL can benefit more practical applications in the future.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was supported by a grant from the Ministry of Science and Technology of Taiwan (No. MOST-108-2637-E-025-002). We thank the Ministry of Science and Technology for funding this study.

References

[1] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1345–1359.
[2] S.L. Happy, A. Dantcheva, F. Bremond, A weakly supervised learning technique for classifying facial expressions, Pattern Recognit. Lett. 128 (2019) 162–168.
[3] O.M. Parkhi, A. Vedaldi, A. Zisserman, Deep face recognition, 2015.
[4] M.U. Ahmed, K.J. Woo, K.Y. Hyeon, M.R. Bashar, P.K. Rhee, Wild facial expression recognition based on incremental active learning, Cogn. Syst. Res. 25 (2018) 212–222.
[5] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of the International Conference on Learning Representations, ICLR, 2015.
[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings of IEEE conference on computer vision and pattern recognition, CVPR, 2009, pp. 248–255.
[7] J.C. Hung, K.C. Lin, N.X. Lai, Recognizing learning emotion based on convolutional neural networks and transfer learning, Appl. Soft Comput. 84 (2019) 105724.
[8] Ben Tan, Yangqiu Song, Erheng Zhong, Qiang Yang, Transitive transfer learning, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, Association for Computing Machinery, 2015, pp. 1155–1164.
[9] M. Long, Y. Cao, J. Wang, M. Jordan, Learning transferable features with deep adaptation networks, in: Proceedings of the 32nd International conference on machine learning, in Proceedings of Machine Learning Research, PMLR, 37, 2015, June, pp. 97–105.
[10] L. Ge, J. Gao, A. Zhang, OMS-TL: A framework of online multiple source transfer learning, in: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013, October, pp. 2423–2428.
[11] L. Mihalkova, R.J. Mooney, Transfer learning by mapping with minimal target data, in: Proceedings of the AAAI-08 Workshop on Transfer Learning for Complex Tasks, 2008, July.
[12] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, et al., A comprehensive survey on transfer learning, Proc. IEEE (2020).
[13] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, et al., Parameter-efficient transfer learning for NLP. in: Proceedings of the 36th International Conference on Machine Learning (ICML), Proceedings of Machine Learning Research, PMLR, 97, 2019.
[14] G. Huang, Z. Liu, L.v.d. Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 21–26 July 2017, 2017, pp. 2261–2269, http://dx.doi.org/10.1109/CVPR.2017.243.
[15] Z. Ming, J. Chazalon, M.M. Luqman, M. Visani, J.-C. Burie, Facelivenet: End-to-end networks combining face verification with interactive facial expression-based liveness detection, in: 2018 24th International Conference on Pattern Recognition, ICPR, IEEE, 2018, pp. 3507–3512.

[16] Min Lin, Qiang Chen, Shuicheng Yan, Network in network, 2013.
[17] Prajit Ramachandran, Barret Zoph, Quoc V. Le, Searching for activation functions, 2017.
[18] Irwan Bello, Barret Zoph, Vijay Vasudevan, Quoc V. Le, Neural optimizer search with reinforcement learning, in: International Conference on Machine Learning, 2017, pp. 459–468.
[19] Sergey Ioffe, Christian Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
[20] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[21] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019, arXiv preprint arXiv:1910.01108.
[22] Cristian Bucila, Rich Caruana, Alexandru Niculescu-Mizil, Model compression, in: KDD, 2006.
[23] R.W. Picard, Massachusetts Institute of Technology, Affective Computing, MITMedia Laboratory Perceptual Computing Section Technical Report No. 321, Perceptual Computing Section Media Laboratory, Massachusetts Institute of Technology, 1995, p. 26.
[24] BELA M.D. MITTELMANN, HAROLD G.M.D. WOLFF, Emotions and skin temperature: Observations on patients during psychotherapeutic (psychoanalytic) interviews1, 1943.
[25] Paul Salvador Inventado, Roberto Legaspi, The Duy Bui, Merlin Suare, Predicting student's appraisal of feedbask in an ITS using previous affective states and continuous affect labels from EEG data.
[26] A. Mehrabian, Silent Messages, first ed., Wadsworth, Belmont, CA, 1971.
[27] Ekman Paul, J. Davidson, The nature of emotion: Fundamental questions, 1994.
[28] Rafael A. Calvo, Sidney D'Mello, Affect detection: An interdisciplinary review of models, methods, and their applications, IEEE Trans. Affect. Comput. (2010) 18–37, 23 7.
[29] J.M. Sun, X.S. Pei, S.S. Zhou, Facial emotion recognition in modern distant education system using SVM, in: 2008 International Conference on Machine Learning and Cybernetics, Vol. 6, IEEE, 2008, pp. 3545–3548, July.
[30] G. Yang, T.S. Huang, Human face detection in complex background, Pattern Recognit. 27 (1994) 53–63, Jan.
[31] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3 (1991) 71–86.
[32] R. Brunelli, T. Poggio, Face recognition: Features versus templates, IEEE Trans. Pattern Anal. Mach. Intell. 15 (1993) 1042–1052.
[33] P. Ekman, W.V. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologists Press, San Francisco, 1978.
[34] The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression.
[35] Ian J. Goodfellow, Dumitru Erhan, Pierre Luc Carrier, Aaron Courville, Mehdi Mirza, Ben Hamner, Will Cukierski, Yichuan Tang, David Thaler, Dong-Hyun Lee, Yingbo Zhou, Chetan Ramaiah, Fangxiang Feng, Ruifan Li, Xiaojie Wang, Dimitris Athanasakis, John Shawe-Taylor, Maxim Milakov, John Park, Radu Ionescu, Marius Popescu, Cristian Grozea, James Bergstra, Jingjing Xie, Lukasz Romaszko, Bing Xu, Zhang Chuang, Yoshua Bengio, Challenges in representation learning: A report on three machine learning contests, in: ICONIP 2013: Neural Information Processing, pp. 117–124.
[36] A. Ekbal, R. Haque, S. Bandyopadhyay, Named entity recognition in Bengali: a conditional random field approach, in: Proceedings of the 3rd International Joint Conference on Natural Language Processing, IJCNLP 2008, 2008, pp. 589–594.
[37] Borthwick, A Maximum Entropy Approach to Named Entity Recognition (Ph.D. thesis), New York University, 1999.
[38] Asif Ekbal, Sivaji Bandyopadhyay, Named entity recognition using support vector machine: A language independent approach, Int. J. Comput. Syst. Sci. Eng. 4 (2) (2008) 155–170.
[39] I.J. Unanue, E.Z. Borzeshi, M. Piccardi, Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition, J. Biomed. Inf. 76 (2017) 102–109.
[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, et al., Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[41] G.A. Levow, The third international Chinese language processing bakeoff: Word segmentation and named entity recognition, in: Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing, 2006, pp. 108–117.
[42] N. Peng, M. Dredze, Named entity recognition for chinese social media with jointly trained embeddings, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 548–554.
[43] R. Weischedel, S. Pradhan, L. Ramshaw, M. Palmer, N. Xue, M. Marcus, et al., OntoNotes release 4.0. LDC2011T03, 2011, Philadelphia, Penn., Linguistic Data Consortium.
[44] Y. Zhang, J. Yang, Chinese NER using lattice LSTM, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 1554–1564, Retrieved from https://github.com/jiesutd/LatticeLSTM.
[45] Y. Meng, W. Wu, F. Wang, X. Li, P. Nie, F. Yin, et al., Glyce: Glyph-vectors for chinese character representations, in: Advances in Neural Information Processing Systems, 2019, pp. 2742–2753.

Jason C. Hung is an Associate Professor with the Department of Computer Science and Information Engineering at National Taichung University of Science and Technology, Taiwan, ROC. His research interests include multimedia systems, e-learning, affective computing, artificial intelligence, and social computing. Dr. Hung received his BS and MS degrees in Computer Science and Information Engineering from Tamkang University in 1996 and 1998, respectively. He received his Ph.D. in Computer Science and Information Engineering from Tamkang University in 2001. Dr. Hung has participated in many international academic activities, including the organization of many international conferences. He is the founder of the International Conference on Frontier Computing. He served as Vice Chair of IET Taipei LN. In April 2014, he was elected Fellow of the Institution of Engineering and Technology (FIET). He was elected vice chair of IET Taipei LN in November 2014. Since June 2015, he has been the Editor-in-Chief of the *International Journal of Cognitive Performance Support*.

Jia-Wei Chang is an assistant professor with the Department of Computer Science and Information Engineering at National Taichung University of Science and Technology. Since January 2019, he has been a Young Professionals Chair of the Institution of Engineering and Technology (IET) – Taipei Network. Since 2017, he has been a consultant at the NEXCOM Industry 4.0 Innovation Center. From February to July 2018, he was an adjunct assistant professor with the Department of Engineering Science at National Cheng Kung University. He was a data scientist and project manager at IoT BU, Nexcom, during 2016–2017. He received his Ph.D. degree from the Department of Engineering Science, National Cheng Kung University, in 2017. His research interests include natural language processing, Internet of Things, artificial intelligence, data mining, and e-learning technologies.