JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 691-709 (2023)
DOI: 10.6688/JISE.202305_39(3).0015

You Only Learn One Representation: Unified Network for Multiple Tasks

CHIEN-YAO WANG[1,3,+], I-HAU YEH[2,3] AND HONG-YUAN MARK LIAO[1,3]

[1] *Institute of Information Science*
Academia Sinica
Taipei, 115 Taiwan
[2] *Elan Microelectronics Corporation*
Hsinchu, 308 Taiwan
[3] *Frontier Institute of Research for Science and Technology*
National Taipei University of Technology
Taipei, 106 Taiwan
E-mail: kinyiu@iis.sinica.edu.tw[+]; ihyeh@emc.com.tw; liao@iis.sinica.edu.tw

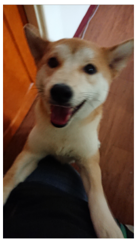
People "understand" the world via vision, hearing, tactile, and also the past experience. Human experience can be learned through normal learning (we call it explicit knowledge), or subconsciously (we call it implicit knowledge). These experiences learned through normal learning or subconsciously will be encoded and stored in the brain. Using these abundant experience, as a huge database, human beings can effectively process data, even they were unseen beforehand. In this paper, we propose a unified network to encode implicit knowledge and explicit knowledge together, just like the human brain can learn knowledge from normal learning as well as subconsciousness learning. The unified network can generate a unified representation to simultaneously serve various tasks. We can perform kernel space alignment, prediction refinement, and multi-task learning in a convolutional neural network. The results demonstrate that when implicit knowledge is introduced into the neural network, it benefits the performance of all tasks. We further analyze the implicit representation learnt from the proposed unified network, and it shows great capability on catching the physical meaning of different tasks. The source code of this work is at : https://github.com/WongKinYiu/yolor.

Keywords: unified network, representation learning, multiple task learning, image classification, object detection, multiple object tracking

1. INTRODUCTION

As shown in Fig. 1, humans can analyze the same piece of data from various angles. However, a trained convolutional neural network (CNN) model can usually only fulfill a single objective. Therefore, the features that can be extracted from a trained CNN are usually poorly adaptable to other types of objectives. The main cause for the above problem is that we only extract features from neurons. As to the implicit knowledge, which is abundant in CNN, is not adequately used. In fact, when a real human brain is operating, the aforementioned implicit knowledge can effectively assist the brain to execute various tasks.

Received February 7, 2023; accepted February 22, 2023.
Communicated by Tsan-sheng Hsu.
[+] Corresponding author.



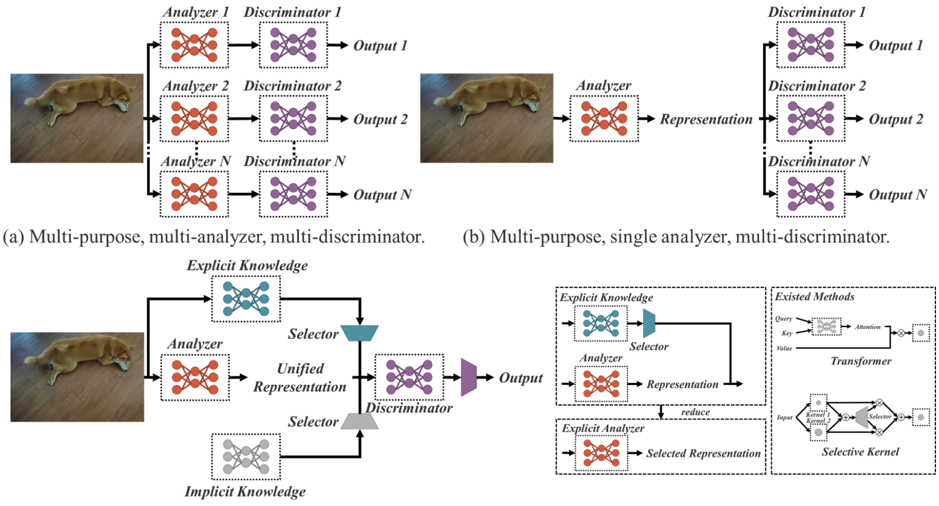
→What is this?	→A Shiba Inu.
→Where is the Shiba Inu?	→In a room.
→Where is she?	→In a room.
→What is she doing?	→LOL.
→What is her name?	→A I.
→Do you love her?	→Yes! Sure! Of course!

Fig. 1. Human beings can answer different questions from a same input. Our aim is to use a single deep neural network to execute multiple tasks.

Implicit knowledge refer to the knowledge learned in a subconscious manner. However, there is no systematic definition of how implicit learning operates and how to obtain implicit knowledge. As an example in Fig. 1, if the ultimate goal of a certain task is to ask "what kind of dog is in your photo?", then the system will use the feature organization mechanism (*e.g.*, by composing certain feature pyramid networks) to organize features, and then tell you "it is a Shibu Inu." But if the objective of another task is to ask you "where is this Shibu Inu?", then the system will seek the answer through another combination of feature pyramid networks and answer "the Shibu Inu is in a room." The backbone of this system actually encodes a wealth of implicit knowledge, and these implicit knowledge can support to organize different feature combinations for tasks with different objectives. For the objective of a specific task, the features integrated by the system to solve the problem are called explicit knowledge. Our system allows solving tasks with different goals by combining features already present in the backbone networks.

In this paper, we propose a unified network to integrate implicit knowledge and explicit knowledge. The unified network enables the learned model to contain a general representation, and this general representation can provide sub-representations suitable for various tasks. Fig. 2 (c) illustrates the proposed unified network architecture. The way to construct the unified networks is to combine compressive sensing and deep learning, and the main theoretical basis can be found in our previous work [16–18]. The proposed unified network can encode implicit knowledge and explicit knowledge together, just like how the human brain learns various things from the external world, and such a learning mechanism can be conscious learning with specific goals or subconscious learning without specific goals. The proposed network can generate a unified representation to simultaneously serve multiple objectives. For example, this unified representation can be used to execute kernel space alignment, prediction refinement, as well as multi-task learning. Experiment results demonstrate that when implicit knowledge is introduced into the unified network, it benefits the performance of all tasks. Besides, we found the implicit representation learnd from the proposed unified network shows great capability on catching the physical meaning of different tasks. The contribution of this work are summarized as follows:

1. We propose a unified network that can simultaneously execute various tasks. It learns a general representation by integrating implicit knowledge and explicit knowledge, and one can complete various tasks through this general representation. The proposed network effectively improves the performance of the model with a very small amount of additional cost. (less than one ten thousand of the amount of parameters and calculations.)



(a) Multi-purpose, multi-analyzer, multi-discriminator.

(b) Multi-purpose, single analyzer, multi-discriminator.

(c) Multi-purpose, single unified network.

Fig. 2. Multi-purpose NN architectures; (a) distinct models for distinct tasks; (b) shared backbone: different heads for different tasks; and (c) our proposed unified network: one representation with explicit knowledge and implicit knowledge for serving multiple tasks.

2. We introduce kernel space alignment, prediction refinement, and multi-task learning into the implicit knowledge learning process, and verified their effectiveness.
3. We discuss the ways of using vector, neural network, or matrix factorization as a tool to model implicit knowledge, and at the same time verified its effectiveness.
4. We confirm the learned implicit representation learned can accurately correspond to a specific physical characteristic, and we also present it in a visual way. Besides, we confirm that if operators that conform to the physical meaning of an objective can be used to integrate implicit knowledge and explicit knowledge, it will have a multiplier effect.
5. Combined with the state-of-the-art methods, our proposed unified network achieves comparable accuracy as Scaled-YOLOv4-P7 [15] on object detection and improves 88% inference speed.

2. RELATED WORK

We conduct a review of the literature that is related to this research topic. This literature review is mainly divided into three aspects: (1) explicit deep learning: it will cover some methods that can automatically adjust or select features based on the input data; (2) implicit deep learning: it will cover the related literature of implicit deep knowledge learning and implicit differential derivative; and (3) knowledge modeling: it will list several methods that can be used to integrate implicit knowledge and explicit knowledge. We briefly summarize them as follows.

2.1 Explicit Deep Learning

Explicit deep learning can be carried out in several ways. Transformer [5, 14, 20] is a commonly used one, and it mainly uses query, key, or value to obtain self-attention. Non-local networks [4, 21, 24] is another way to obtain attention, and it mainly extracts pair-wise attention in time and space. Another commonly used explicit deep learning method [7, 25] is to automatically select the appropriate kernel by input data.

2.2 Implicit Deep Learning

The methods that belong to the category of implicit deep learning are implicit neural representations [11] and deep equilibrium models [2, 3, 19]. The former is to obtain the parameterized continuous mapping representation of discrete inputs to perform different tasks, while the latter is to transform implicit learning into a residual form neural networks, and perform the equilibrium point calculation on it.

2.3 Knowledge Modeling

As for the methods belonging to the category of knowledge modeling, sparse representation [1, 23] and memory networks [12, 22] are included. The former uses exemplar, predefined over complete, or learned dictionary to perform modeling, while the latter relies on combining various forms of embedding to form memory, and enable memory to be dynamically added or changed.

3. HOW IMPLICIT KNOWLEDGE WORKS?

The main purpose of this research is to construct a unified network that can effectively train implicit knowledge, so first we will focus on how to train implicit knowledge and inference it quickly in the follow-up. Since implicit representation $\mathbf{z}_i$ is irrelevant to observation, we can think of it as a set of constant tensor $Z = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$. We will introduce how implicit knowledge as constant tensor can be applied to various tasks.

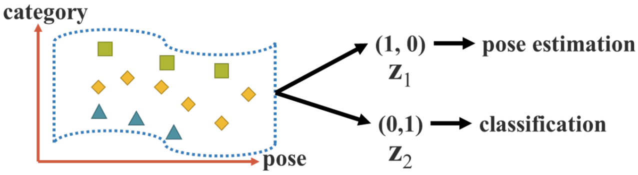
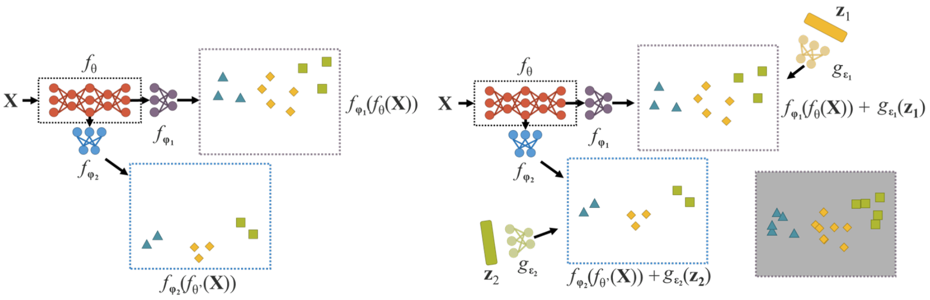


Fig. 3. Manifold space reduction.

3.1 Manifold Space Reduction

We believe that a good representation should be able to find an appropriate projection in the manifold space to which it belongs, and facilitate the subsequent objective tasks to succeed. For example, as shown in Fig. 3, if the target categories can be successfully classified by the hyperplane in the projection space, that will be the best outcome. In the above example, we can take the inner product of the projection vector and implicit representation to achieve the goal of reducing the dimensionality of manifold space and effectively achieving various tasks.

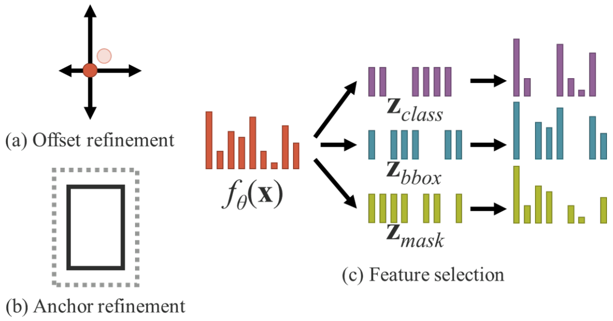

f_θ
$\mathbf{X}$
f_{φ_1}
f_{φ_2}
$f_{\varphi_1}(f_\theta(\mathbf{X}))$
$f_{\varphi_2}(f_{\theta'}(\mathbf{X}))$
$\mathbf{z}_1$
g_{ε_1}
$f_{\varphi_1}(f_\theta(\mathbf{X})) + g_{\varepsilon_1}(\mathbf{z}_1)$
$\mathbf{z}_2$
g_{ε_2}
$f_{\varphi_2}(f_{\theta'}(\mathbf{X})) + g_{\varepsilon_2}(\mathbf{z}_2)$

(a) Without alignment (b) With alignment

Fig. 4. Kernel space alignment.

3.2 Kernel Space Alignment

In multi-task and multi-head neural networks, kernel space misalignment is a frequent problem, Fig. 4 (a) illustrates an example of kernel space misalignment in multi-task and multi-head NN. To deal with this problem, we can perform addition and multiplication of output feature and implicit representation, so that Kernel space can be translated, rotated, and scaled to align each output kernel space of neural networks, as shown in Fig. 4 (b). The above mode of operation can be widely used in different fields, such as the feature alignment of large objects and small objects in feature pyramid networks (FPN) [8], the use of knowledge distillation to integrate large models and small models, and the handling of zero-shot domain transfer and other issues.



(a) Offset refinement

(b) Anchor refinement

(c) Feature selection

Fig. 5. More functions.

3.3 More Functions

In addition to the functions that can be applied to different tasks, implicit knowledge can also be extended into many more functions. As illustrated in Fig. 5, through introducing addition, one can make neural networks predict the offset of center coordinate. It is also possible to introduce multiplication to automatically search the hyper-parameter set of an anchor, which is very often needed by an anchor-based object detector. Besides, dot multiplication and concatenation can be used, respectively, to perform multi-task feature selection and to set pre-conditions for subsequent calculations.

4. IMPLICIT KNOWLEDGE IN OUR UNIFIED NETWORKS

In this section, we shall compare the objective function of conventional networks and the proposed unified networks, and to explain why introducing implicit knowledge is important for training a multi-purpose network. At the same time, we will also elaborate the details of the method proposed in this work.

4.1 Formulation of Implicit Knowledge

Conventional Networks:

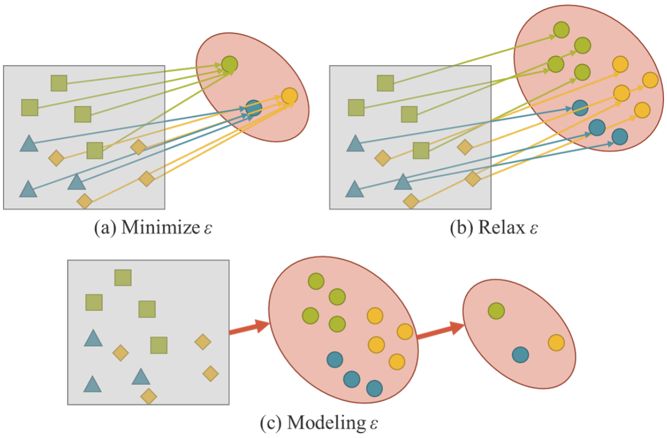
For the objective function of conventional network training, we can use the equation shown as follows:

$$\begin{aligned} &y = f_\theta(\mathbf{x}) + \varepsilon \\ &\text{minimize } \varepsilon \end{aligned} \tag{1}$$

where $\mathbf{x}$ is observation, θ is the set of parameters of a neural network, f_θ represents operation of the neural network, ε is error term, and y is the target of a given task.

In the training process of a conventional neural network, usually one will minimize ε to make $f_\theta(\mathbf{x})$ as close to the target as possible. This means that we expect different observations with the same target to be a single point in the sub-space obtained by f_θ, as illustrated in Fig. 6 (a). In other words, the solution space we expect to obtain is discriminative only for the current task t_i and invariant to tasks other than t_i in various potential tasks, $T \setminus t_i$, where $T = \{t_1, t_2, ..., t_n\}$.

For general-purpose neural network, we hope that the obtained representation can serve all tasks belonging to T. Therefore, we need to relax ε to make it possible to find solution of each task at the same time on manifold space, as shown in Fig. 6 (b). However, the above requirements make it impossible for us to use a trivial mathematical method, such as maximum value of one-hot vector or threshold of Euclidean distance, to get the solution of t_i. In order to solve the problem, we must model the error term ε to find solutions for different tasks, as shown in Fig. 6 (c).


(a) Minimize ε

(b) Relax ε

(c) Modeling ε

Fig. 6. Modeling error term.

Unified Networks:

To train the proposed unified networks, we use explicit and implicit knowledge together to model the error term, and then use it to guide the multi-purpose network training process. The corresponding equation for training is as follows:

$$\begin{aligned} &y = f_\theta(\mathbf{x}) + \varepsilon + g_\phi(\varepsilon_{ex}(\mathbf{x}), \varepsilon_{im}(\mathbf{z})) \\ &\text{minimize } \varepsilon + g_\phi(\varepsilon_{ex}(\mathbf{x}), \varepsilon_{im}(\mathbf{z})) \end{aligned} \tag{2}$$

where ε_{ex} and ε_{im} are operations which modeling, respectively, the explicit error and implicit error from observation $\mathbf{x}$ and latent code $\mathbf{z}$. g_ϕ here is a task specific operation that serves to combine or select information from explicit knowledge and implicit knowledge.

There are some existing methods to integrate explicit knowledge into f_θ, such as attention mechanism and dynamic kernel, so we can rewrite Eq. (2) into Eq. (3).

$$y = f_\theta(\mathbf{x}) \star g_\phi(\mathbf{z}) \tag{3}$$

where $\star$ represents some possible operators that can combine f_θ and g_ϕ. In this work, the operators introduced in Section 3 will be used, which are addition, multiplication, and concatenation.

If we extend derivation process of error term to handling multiple tasks, we can get the following equation:

$$F(\mathbf{x}, \theta, \mathbf{Z}, \Phi, Y, \Psi) = 0 \tag{4}$$

where $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_T\}$ is a set of implicit latent codes of T different tasks. Φ are the parameters that can be used to generate implicit representation from $\mathbf{Z}$. Ψ is used to calculate the final output parameters from different combinations of explicit representation and implicit representation.

For different tasks, we can use the following formula to obtain prediction for all $\mathbf{z} \in \mathbf{Z}$.

$$d_\Psi(f_\theta(\mathbf{x}), g_\Phi(\mathbf{z}), y) = 0 \tag{5}$$

For all tasks we start with a common unified representation $f_\theta(\mathbf{x})$, go through task-specific implicit representation $g_\Phi(\mathbf{z})$, and finally complete different tasks with task-specific discriminator d_Ψ.

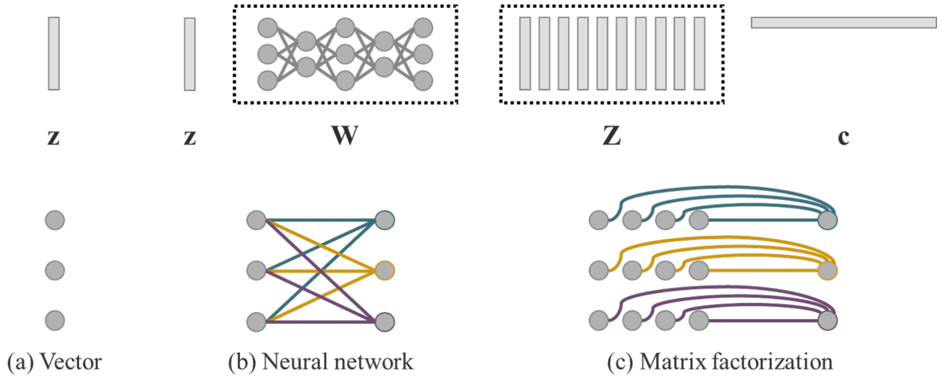
4.2 Modeling Implicit Knowledge

The implicit knowledge we proposed can be modeled in the following ways:

Vector / Matrix / Tensor:

$$\mathbf{z} \tag{6}$$

Use vector $\mathbf{z}$ directly as the prior of implicit knowledge and the implicit representation. At this time, it must be assumed that each dimension is independent of each other.



(a) Vector (b) Neural network (c) Matrix factorization

Fig. 7. We proposed to use three different ways for modeling implicit knowledge. The top row shows the formation of these three different modeling approaches, and the bottom row shows their corresponding mathematical attributes; (a) Vector: single base, and each dimension is independent with another dimensions; (b) Neural Network: single or multiple basis, and each dimension is dependent to another dimensions; and (c) Matrix factorization: multiple basis, and each dimension is independent with another dimensions.

Neural Network:

$$\mathbf{W}\mathbf{z} \tag{7}$$

Use vector $\mathbf{z}$ as the prior of implicit knowledge, then use the weight matrix $\mathbf{W}$ to perform linear combination or nonlinearization and then become an implicit representation. At this time, it must be assumed that each dimension is dependent on each other. We can use more complex neural network to generate implicit representation, or use Markov chain to simulate the correlation of implicit representation between different tasks.

Matrix Factorization:

$$\mathbf{Z}^{\mathbf{T}}\mathbf{c} \tag{8}$$

Use multiple vectors as prior of implicit knowledge, and these implicit prior basis $\mathbf{Z}$ and coefficient $\mathbf{c}$ will form implicit representation. We can further do sparse constraint to $\mathbf{c}$ and convert it into sparse representation form. In addition, we can impose non-negative constraint on $\mathbf{Z}$ and $\mathbf{c}$ to convert them into non-negative matrix factorization (NMF) form.

4.3 Training

Assuming that our model does not have any prior implicit knowledge at the beginning, that is to say, it will not have any impact on explicit representation $f_\theta(\mathbf{x})$. When the combining operator $\star \in \{addition, concatenation\}$, the initial implicit prior $\mathbf{z} \sim N(0,\sigma)$, and when the combining operator $\star$ is $multiplication$, $\mathbf{z} \sim N(1,\sigma)$. Here, σ is a very small value which is close to zero. As for $\mathbf{z}$ and ϕ, they both are trained with backpropagation algorithm during the training process.

4.4 Inference

Since implicit knowledge is irrelevant to observation **x**, no matter how complex the implicit model g_ϕ is, it can be reduced to a set of constant tensors before the inference phase is executed. In other words, the formation of implicit information has almost no effect on the computational complexity of our algorithm. In addition, when the above operator is multiplication, if the subsequent layer is a convolutional layer, then we use Eq. (9) below to integrate. When one encounters an addition operator, and if the previous layer is a convolutional layer and it has no activation function, then we use Eq. (10) shown below to integrate.

$$\begin{aligned} \mathbf{x}_{(l+1)} &= \sigma(W_l(g_\phi(\mathbf{z})\mathbf{x}_l) + b_l) \\ &= \sigma(W_l^{'}(\mathbf{x}_l) + b_l), \text{where } W_l^{'} = W_l g_\phi(\mathbf{z}) \end{aligned} \tag{9}$$

$$\begin{aligned} \mathbf{x}_{(l+1)} &= W_l(\mathbf{x}_l) + b_l + g_\phi(\mathbf{z}) \\ &= W_l(\mathbf{x}_l) + b_l^{'}, \text{where } b_l^{'} = b_l + g_\phi(\mathbf{z}) \end{aligned} \tag{10}$$

5. EXPERIMENTS

Our experiments adopted the MSCOCO dataset [9], because it provides ground truth for many different tasks, including **object detection**, **instance segmentation**, **panoptic segmentation**, **keypoint detection**, **stuff segmentation**, **image caption**, **multi-label image classification**, and **long tail object recognition**. These data with rich annotation content can help train a unified network that can support computer vision-related tasks as well as natural language processing tasks.

5.1 Experimental Setup

In the experimental design, we chose to apply implicit knowledge to three aspects, including **feature alignment for FPN**, **prediction refinement**, and **multi-task learning in a single model**. The tasks covered by multi-task learning include object detection, multi-label image classification, and feature embedding. We chose YOLOv4-CSP [15] as the baseline model in the experiments, and introduce implicit knowledge into the model at the position pointed by the arrow in Fig. 8. All the training hyper-parameters are compared to default setting of Scaled-YOLOv4 [15].

In Sections 5.2, 5.3, and 5.4, we used the simplest vector implicit representation and addition operator to verify the positive impact on various tasks when implicit knowledge is introduced. In Section 5.5, we will use different operators on different combinations of explicit knowledge and implicit knowledge, and discuss the effectiveness of these combinations. In Section 5.6, we shall model implicit knowledge by using different approaches. In Section 5.7, we analyze the model with and without the introduction of implicit knowledge. Finally in Section 5.8, we shall train object detectors with implicit knowledge and then compare the performance with state-of-the-art methods.

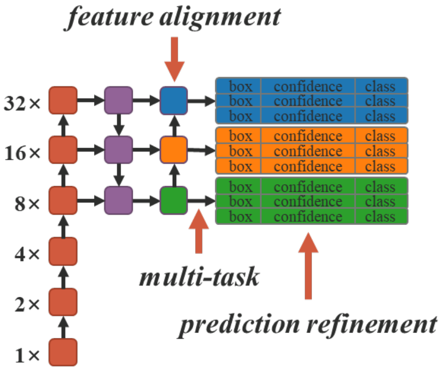


Fig. 8. Base architecture. We introduce implicit knowledge modeling for doing feature alignment, prediction refinement, and multi-task learning.

5.2 Feature Alignment for FPN

We add implicit representation into the feature map of each FPN for feature alignment, and the corresponding experiment results are illustrated in Table 1. From the results shown in Table 1 we can say: After using implicit representation for feature space alignment, all performances, including AP_S, AP_M, and AP_L, have been improved by about 0.5%, which is a very significant improvement.

Table 1. Ablation study of feature alignment.

Model	$\mathbf{AP}^{val}$	$\mathbf{AP}^{val}_{50}$	$\mathbf{AP}^{val}_{75}$	$\mathbf{AP}^{val}_{S}$	$\mathbf{AP}^{val}_{M}$	$\mathbf{AP}^{val}_{L}$
baseline	47.8%	66.3%	52.1%	30.1%	52.5%	62.0%
+ *i*FA	**47.9%**	**66.6%**	**52.3%**	**30.6%**	**53.1%**	**62.6%**

* baseline is YOLOv4-CSP-fast, tested on 640×640 input resolution.
* FA: feature alignment.

5.3 Prediction Refinement for Object Detection

Implicit representations are added to YOLO output layers for prediction refinement. As illustrated in Table 2, we see that almost all indicator scores have been improved. Fig. 9 shows how the introduction of implicit representation affects the detection outcome. In the object detection case, even we do not provide any prior knowledge for implicit representation, the proposed learning mechanism can still automatically learn (x,y), (w,h), (obj), and $(classes)$ patterns of each anchor.

Table 2. Ablation study of prediction refinement.

Model	$\mathbf{AP}^{val}$	$\mathbf{AP}^{val}_{50}$	$\mathbf{AP}^{val}_{75}$	$\mathbf{AP}^{val}_{S}$	$\mathbf{AP}^{val}_{M}$	$\mathbf{AP}^{val}_{L}$
baseline	47.8%	66.3%	52.1%	30.1%	52.5%	62.0%
+ *i*PR	**47.8%**	**66.5%**	**52.1%**	**30.3%**	**53.3%**	61.5%

* baseline is YOLOv4-CSP-fast, tested on 640×640 input resolution.
* PR: prediction refinement.

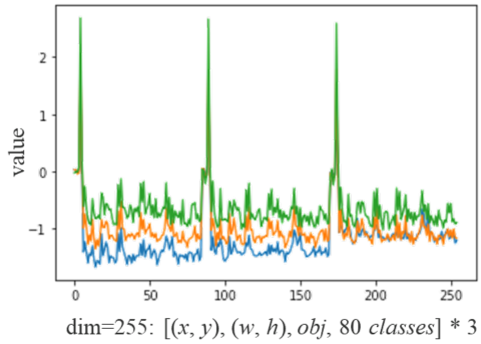


Fig. 9. Value of learned implicit representation for prediction refinement. The learned implicit knowledge could automatically mapping $[(x,y),(w,h),obj,cls]$ information of different anchors.

5.4 Canonical Representation for Multi-Task

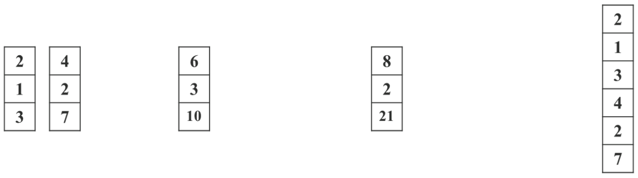
When one wants to train a model that can be shared by many tasks at the same time, since the joint optimization process on loss function must be executed, multiple parties often pull each other during the execution process. The above situation will cause the final overall performance to be worse than training multiple models individually and then integrating them. In order to solve the above problem, we propose to train a canonical representation for multi-tasks. Our idea is to augment the representation power by introducing implicit representation to each task branch, and the effect it causes is listed in Table 3. As the data illustrated in Table 3, without the introduction of implicit representation, some index scores improved after multi-task training, and some dropped. After introducing implicit representation to joint detection and classification (JDC), in the model category corresponding to + *i*JDC, we can clearly see that the overall index score has increased significantly, and it has surpassed the performance of single-task training model. Compared to when implicit representation was not introduced, the performance of our model on medium-sized objects and large-sized objects has also been improved by 0.3% and 0.7%, respectively. In the experiment of joint detection and embedding (JDE), because of the characteristic of implicit representation implied by feature alignment, the effect of improving the index score is more significant. Among the index scores corresponding to JDE and + *i*JDE listed in Table 3, all index scores of + *i*JDE surpass the index that does not introduce implicit representation. Among them, the AP for large objects even increased by 1.1%.

Table 3. Ablation study of multi-task joint learning.

Model	**AP^{val}**	**AP$^{val}_{50}$**	**AP$^{val}_{75}$**	**AP$^{val}_{S}$**	**AP$^{val}_{M}$**	**AP$^{val}_{L}$**
baseline	48.0%	66.8%	52.3%	30.0%	53.0%	62.7%
JDC	47.7%	**66.8%**	51.9%	**30.8%**	52.4%	61.6%
+ *i*JDC	**48.1%**	**67.1%**	52.2%	**31.1%**	52.7%	62.3%
JDE	**48.1%**	66.7%	**52.4%**	**30.7%**	**53.2%**	61.9%
+ *i*JDE	**48.3%**	**66.8%**	**52.6%**	**30.7%**	**53.4%**	**63.0%**

[*] baseline is YOLOv4-CSP [15], tested on 640×640 input resolution.

[*] JD{C, E}: joint detection & {clssification, embedding}.



(a) Input pair (b) '+' addition (c) '×' multiplication (d) '⊕' concatenation

Fig. 10. Implicit modeling with (b) addition, (c) multiplication, and (d) concatenation operators.

5.5 Implicit Modeling with Different Operators

Table 4 shows the experimental results of using different operators shown in Fig. 10 to combine explicit representation and implicit representation. In the implicit knowledge for feature alignment experiment, we see that addition and concatenation both improve performance, while multiplication actually degrades performance. The experimental results of feature alignment are in full compliance with its physical characteristics, because it must deal with the scaling of global shift and all individual clusters. In the implicit knowledge for prediction refinement experiment, since the operator of concatenation will change the dimension of output, we only compare the effects of using addition and multiplication operators in the experiment. In this set of experiments, the performance of applying multiplication is better than that of applying addition. Analyzing the reason, we found that center shift uses addition decoding when executing prediction, while anchor scale uses multiplication decoding. Because center coordinate is bounded by grid, the impact is minor, and the artificially set anchor owns a larger optimization space, so the improvement is more significant.

Table 4. Ablation study of different operators.

Model	**AP**^{val}	**AP**$_{50}^{val}$	**AP**$_{75}^{val}$	**AP**$_{S}^{val}$	**AP**$_{M}^{val}$	**AP**$_{L}^{val}$
baseline	47.8%	66.3%	52.1%	30.1%	52.5%	62.0%
+ *i*FA	**47.9%**	**66.6%**	**52.3%**	**30.6%**	**53.1%**	**62.6%**
× ***i*FA**	47.4%	65.8%	51.6%	29.6%	52.2%	**62.1%**
⊕ ***i*FA**	**47.8%**	**66.5%**	**52.2%**	**30.3%**	**52.9%**	**62.3%**
+ *i*PR	**47.8%**	**66.5%**	**52.1%**	**30.3%**	**53.3%**	61.5%
× ***i*PR**	**48.0%**	**66.7%**	**52.3%**	29.8%	**53.4%**	61.8%

* baseline is YOLOv4-CSP-fast, tested on 640×640 input resolution.
* {+, ×, ⊕}: {addition, multiplication, concatenation}.

Based on the above analysis, we designed two other sets of experiments – {× *i*FA*, × *i*PR*}. In the first set of experiments – × *i*FA*, we split feature space into anchor cluster level for combination with multiplication, while in the second set of experiments – × *i*PR*, we only performed multiplication refinement on width and height in prediction. The results of the above experiments are illustrated in Table 5. From the figures shown in Table 5, we find that after corresponding modifications, the scores of various indices have been comprehensively improved. The experiment shows that when designing how to combine explicit and implicit knowledge, we must first consider the physical meaning of the combined layers to achieve a multiplier effect.

Table 5. Ablation study of different operators.

Model	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
baseline	47.8%	66.3%	52.1%	30.1%	52.5%	62.0%
× ***i*FA***	**47.9%**	**66.6%**	52.0%	**30.5%**	**52.6%**	**62.3%**
× ***i*PR***	**48.1%**	**66.5%**	**52.1%**	**30.1%**	**53.3%**	61.9%

* baseline is YOLOv4-CSP-fast, tested on 640×640 input resolution.

5.6 Modeling Implicit Knowledge in Different Ways

We tried to model implicit knowledge in different ways, including vector, neural networks, and matrix factorization. When modeling with neural networks and matrix factorization, the default value of implicit prior dimension is twice that of explicit representation dimension. The results of this set of experiments are shown in Table 6. We can see that whether it is to use neural networks or matrix factorization to model implicit knowledge, it will improve the overall effect. Among them, the best results have been achieved by using matrix factorization model, and it upgrades the performance of AP, AP_{50}, and AP_{75} by 0.2%, 0.4%, and 0.5%, respectively. In this experiment, we demonstrated the effect of using different modeling ways. Meanwhile, we confirmed the potential of implicit representation in the future.

Table 6. Ablation study of different modeling approaches.

Model	AP^{val}	AP^{val}_{50}	AP^{val}_{75}	AP^{val}_{S}	AP^{val}_{M}	AP^{val}_{L}
baseline	47.8%	66.3%	52.1%	30.1%	52.5%	62.0%
+ *i*FA	**47.9%**	**66.6%**	**52.3%**	**30.6%**	**53.1%**	**62.6%**
+ *wi*FA	**47.8%**	**66.4%**	52.0%	**30.8%**	**52.8%**	61.9%
+ *ic*FA	**48.0%**	**66.7%**	**52.6%**	**30.3%**	**53.2%**	**62.5%**

* baseline is YOLOv4-CSP-fast, tested on 640×640 input resolution.

* {*i*, *wi*, *ic*}: {vector, neural network, matrix factorization}, see 4.2.

5.7 Analysis of Implicit Models

We analyze the number of parameters, FLOPs, and learning process of model with/without implicit knowledge, and show the results in Table 7 and Fig. 11, respectively. From the experimental data, we found that in the model with implicit knowledge set of experiments, we only increase the amount of parameters and calculations by less than one ten thousandth, but significantly improve the performance of the model, and the training process can converge quickly and correctly.

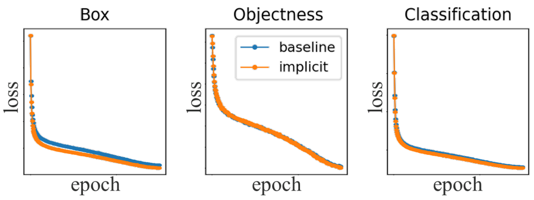


Fig. 11. Learning curve of model with and without implicit knowledge.

Table 7. Information of model with/without implicit knowledge.

Model	**AP^{val}**	**# parameters**	**MFLOPs**
baseline 1	47.8%	52908989	117517.2952
implicit 1	**48.0%**	**52911546** (+0.005%)	**117519.4372** (+0.002%)
baseline 2	51.4%	37262204	326256.1624
implicit 2	**51.9%**	**37265016** (+0.008%)	**326264.7304** (+0.003%)

* baseline 1 is YOLOv4-CSP-fast, tested on 640×640 input resolution.

* baseline 2 is YOLOv4-P6-light, tested on 1280×1280 input resolution.

* implicit $\{1, 2\}$ are baseline $\{1, 2\}$ with + *i*FA, × *i*PR.

5.8 Implicit Knowledge for Object Detection

We compare the effectiveness of the proposed method with object detection's state-of-the-art methods. The benefits of introducing implicit knowledge are shown in Table 8. For the entire training process, we follow the scaled-YOLOv4 [15] training process, that is, train from scratch 300 epochs first, and then fine-tune 150 epochs. Table 9 illustrates the comparisons with the state-of-the-art methods. One thing worth noting is that our proposed method does not have additional training data and annotations, and it achieves comparable accuracy as Scaled-YOLOv4 and improves 88% inference speed in the same condition. By introducing the unified network of implicit knowledge, we still achieve results that are sufficient to match the state-of-the-art methods.

Table 8. Benefit from implicit knowledge.

Model	**AP^{val}**	**AP$^{val}_{50}$**	**AP$^{val}_{75}$**	**AP$^{val}_{S}$**	**AP$^{val}_{M}$**	**AP$^{val}_{L}$**
baseline	51.4%	69.5%	56.4%	35.2%	55.8%	64.6%
implicit	**51.9%**	**69.8%**	**56.8%**	**36.0%**	**56.3%**	**65.0%**
fine-tuned implicit	**52.5%**	**70.5%**	**57.6%**	**37.1%**	**57.2%**	**65.4%**

* baseline is YOLOv4-P6-light, tested on 1280×1280 input resolution.

* implicit is baseline with + *i*FA, × *i*PR.

Table 9. Comparion of state-of-the-art.

Method	**pre.**	**seg.**	**add.**	**AP^{test}**	**AP$^{test}_{50}$**	**AP$^{test}_{75}$**	**FPS^{V100}**
YOLOR (ours)				55.4%	73.3%	60.6%	30
ScaledYOLOv4 [15]				55.5%	73.4%	60.8%	16
YOLOR (ours)			✓	58.2%	75.8%	63.8%	30
EfficientDet [13]	✓			55.1%	74.3%	59.9%	6.5
SwinTransformer [10]	✓	✓		57.7%	–	–	–
CenterNet2 [26]	✓		✓	56.4%	74.0%	61.6%	–
CopyPaste [6]	✓	✓	✓	57.3%	–	–	–

* pre. : large dataset image classification pre-training.

* seg. : training with segmentation ground truth.

* add. : training with additional images.

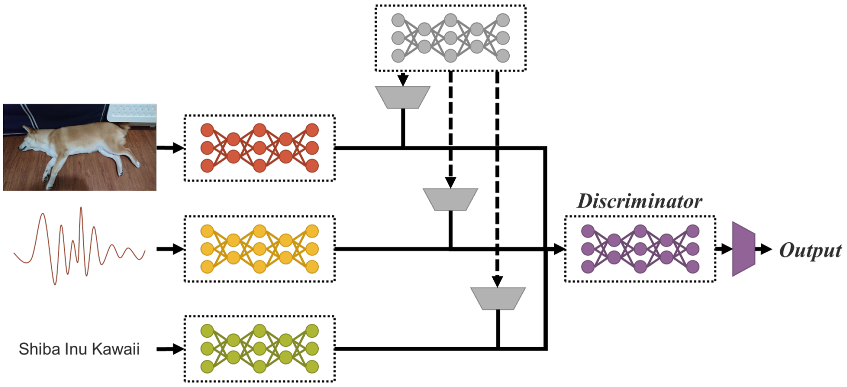


Fig. 12. Multimodal unified netwrok.

5.9 Implicit Knowledge for Different Methods

Finally, we apply proposed implict knowledge model on different methods. The performance is shown in Table 10. We can find that the proposed unified network can benefits the performance of all methods, especially for high quality prediction.

Table 10. Implicit knowledge for different methods.

Model	AP^{box}	AP_{50}^{box}	AP_{75}^{box}	AP^{mask}	AP_{50}^{mask}	AP_{75}^{mask}
Faster R-CNN [27]	37.4%	58.1%	40.4%	–	–	–
+ implicit	**37.6%**	**58.5%**	**40.8%**	–	–	–
Mask R-CNN [28]	38.2%	58.8%	41.4%	34.7%	55.7%	37.2%
+ implicit	**38.3%**	**59.1%**	**41.9%**	**34.8%**	**55.8%**	**37.3%**
Sparse R-CNN [29]	37.9%	56.0%	40.5%	–	–	–
+ implicit	**38.0%**	**56.3%**	**40.5%**	–	–	–
FCOS [30]	36.6%	56.0%	38.8%	–	–	–
+ implicit	**36.6%**	**56.1%**	**39.1%**	–	–	–
ATSS [31]	39.4%	57.6%	42.8%	–	–	–
+ implicit	**39.6%**	**57.8%**	**42.8%**	–	–	–

6. CONCLUSIONS

In this paper, we show how to construct a unified network that integrates implicit knowledge and explicit knowledge, and prove that it is still very effective for multi-task learning under the single model architecture. In the future, we shall extend the training to multi-modal and multi-task, as shown in Fig. 12.

ACKNOWLEDGEMENTS

The authors wish to thank National Center for High-performance Computing (NCHC) for providing computational and storage resources.

REFERENCES

1. M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, Vol. 54, 2006, pp. 4311-4322.
2. S. Bai, J. Z. Kolter, and V. Koltun, "Deep equilibrium models," *Advances in Neural Information Processing Systems*, 2019, pp. 688-699.
3. S. Bai, V. Koltun, and J. Z. Kolter, "Multiscale deep equilibrium models," *Advances in Neural Information Processing Systems*, 2020, pp. 5238-5250.
4. Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proceedings of IEEE International Conference on Computer Vision Workshop*, 2019, pp. 1971-1980.
5. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of European Conference on Computer Vision*, 2020, pp. 213-229.
6. G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2918-2928.
7. X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 510-519.
8. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117-2125.
9. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of European Conference on Computer Vision*, 2014, pp. 740-755.
10. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of IEEE International Conference on Computer Vision*, pp. 10012-10021.
11. V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *Advances in Neural Information Processing Systems*, 2020, pp. 7462-7473.
12. S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," *Advances in Neural Information Processing Systems*, 2015, pp. 2440-2448.
13. M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10781-10790.
14. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017, pp. 5998-6008.
15. C.-Y. Wang, A. Bochkovskiy, and H.-Y. Mark Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13029-13038.
16. C.-Y. Wang, S. Mathulaprangsan, B.-W. Chen, Y.-H. Chin, J.-J. Shiu, Y.-S. Lin, and J.-C. Wang, "Robust face verification via bayesian sparse representation," in *Proceed-*

ings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, 2016, pp. 1-4.

17. C.-Y. Wang, A. Santoso, S. Mathulaprangsan, C.-C. Chiang, C.-H. Wu, and J.-C. Wang, "Recognition and retrieval of sound events using sparse coding convolutional neural network," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2017, pp. 589-594.
18. C.-Y. Wang, T.-C. Tai, J.-C. Wang, A. Santoso, S. Mathulaprangsan, C.-C. Chiang, and C.-H. Wu, "Sound events recognition and retrieval using multi-convolutional-channel sparse coding convolutional neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, 2020, pp. 1875-1887.
19. T. Wang, X. Zhang, and J. Sun, "Implicit feature pyramid network for object detection," *arXiv Preprint*, 2020, arXiv:2012.13563.
20. W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," *Proceedings of IEEE International Conference on Computer Vision*, 2021, pp. 568-578.
21. X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794-7803.
22. J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proceedings of International Conference on Learning Representations*, 2015, arXiv:1410.3916.
23. J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, 2008, pp. 210-227.
24. M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, and Han Hu, "Disentangled non-local neural networks," in *Proceedings of European Conference on Computer Vision*, 2020, pp. 191-207.
25. H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, *et al*., "ResNeSt: Split-attention networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2736-2746.
26. X. Zhou, V. Koltun, and P. Krähenbühl, "Probabilistic two-stage detection," *arXiv Preprint*, 2021, arXiv:2103.07461.
27. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, 2015, pp. 91-99.
28. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *Proceedings of IEEE International Conference on Computer Vision*, 2017, pp. 2961-2969.
29. P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, et al., "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14454-14463.
30. Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," *Proceedings of IEEE International Conference on Computer Vision*, 2019, pp. 9627-9636.

31. S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9759-9768.

Chien-Yao Wang received the Ph.D. degree in Computer Science and Information Engineering from National Central University, Zhongli, Taiwan, in 2017. He is currently an Assistant Research Fellow with the Institute of Information Science, Academia Sinica, Taiwan. His research interests include signal processing, deep learning, and machine learning. Currently, his research focuses on multi-task representation learning for multimodal signal.

I-Hau Yeh received his Master degree of Electronic Engineering from National Chiao-Tung University, Taiwan, in 1986. He co-founded Elan Microelectronics as the President in May 1994, and currently is the Chairman and President. He has built Elan as a world leader in human-machine interfaces, notably in touchscreen controller, touchpad module, pointing stick and fingerprint sensors. In addition, he is also the Chairman of Avisonic Technology, Pixord Corporation, Chimei Motor and Metanoia Communication of the Elan Group, which is dedicated to developing technologies for human device interface, image processing, and networking. Moreover, Mr. Yeh was honored with Outstanding Alumni by National Taipei University of Technology in 1997, and Outstanding Alumni by National Chiao-Tung University, Taiwan, in 2006. He was awarded with "The Best CEO" by Taiwan's Professional Management Association in 2013, and was granted the Honorary Doctorate Degree of Engineering by National Taipei University of Technology in 2020. He was also named on Harvard Business Review's "Taiwan Top 100 best performing CEOs" in 2020 and 2022. Currently, he has been granted with 24 patents internationally.

Hong-Yuan Mark Liao received his Ph.D. degree in Electrical Engineering from Northwestern University in 1990. In July 1991, he joined the Institute of Information Science, Academia Sinica, Taiwan and currently, is a Distinguished Research Fellow and Director. He has worked in the fields of multimedia information processing, computer vision, pattern recognition, multimedia protection, and artificial intelligence for more than 30 years. He was appointed as an Honorary Chair Professor of National Chiao-Tung University from 2016 to 2019. He received the Young Investigators' Award from Academia Sinica in 1998; the Distinguished Research Award from the National Science Council in 2003, 2010 and 2013; the Academia Sinica Investigator Award in 2010; the TECO Award from the TECO Foundation in 2016, and the 64th Academic Award from the Ministry of Education in 2020. His professional activities include President, Image Processing and Pattern Recognition Society of Taiwan (2006-2008); Editorial Board Member, ACM Computing Surveys (associate editor, 2018-2021, senior editor, 2021-present), IEEE Signal Processing Magazine (2010-2013); Associate Editor, IEEE Transactions on Image Processing (2009-13), IEEE Transactions on Information Forensics and Security (2009-2012) and IEEE Transactions on Multimedia (1998-2001). He has been a Fellow of the IEEE since 2013.