Unsupervised Fusion Feature Matching for Data Bias in Uncertainty Active Learning

Wei Huang, Shuzhou Sun, Xiao Lin, Ping Li, *Member, IEEE*, Lei Zhu, Jihong Wang, C. L. Philip Chen, *Fellow, IEEE*, and Bin Sheng, *Member, IEEE*

***Abstract*— Active learning (AL) aims to sample the most valuable data for model improvement from the unlabeled pool. Traditional works, especially uncertainty-based methods, are prone to suffer from a data bias issue, which means that selected data cannot cover the entire unlabeled pool well. Although there have been lots of literature works focusing on this issue recently, they mainly benefit from the huge additional training costs and the artificially designed complex loss. The latter causes these methods to be redesigned when facing new models or tasks, which is very time-consuming and laborious. This article proposes a feature-matching-based uncertainty that resamples selected uncertainty data by feature matching, thus removing similar data to alleviate the data bias issue. To ensure that our proposed method does not introduce a lot of additional costs, we specially design a unsupervised fusion feature matching (UFFM), which does not require any training in our novel AL framework. Besides, we also redesign several classic uncertainty methods to be applied to more complex visual tasks. We conduct rigorous experiments on lots of standard benchmark datasets to validate our work. The experimental results show that our UFFM is better than the similar unsupervised feature matching technologies, and our proposed uncertainty calculation method outperforms random sampling, classic uncertainty approaches, and recent state-of-the-art (SOTA) uncertainty approaches.**

***Index Terms*— Active learning (AL), data bias, deep learning, feature fusion, feature matching, neural network, uncertainty.**

I. Introduction

In this era of data flooding, labeling all of them for supervised learning is very time-consuming and laborious and thus is not realistic [2], [3], [4], [5]. Albeit exhilarating prosperity in the line of semi-supervised [6] and unsupervised learning [7], supervised learning is still better than those two technologies in most scenarios [8]. Therefore, it is a key way to improve the model performance to take some of the most valuable data from the unlabeled pool for supervised learning, which is what active learning (AL) does [9], [10]. The existing AL methods are mainly divided into three groups.

1) Uncertainty-based approaches [3], [9], [11], [12], [13], [14]. This kind of algorithm calculated the uncertainty of unlabeled data based on the current learned model.

2) Diversity-based approaches [13], [15]. Diversity approaches preferentially selected the batch of data with the most dispersed feature distance.

3) Expected model change approaches [10], [16]. They took the processed unlabeled data (e.g., adding noise) as the inputs to observe the changes in the model outputs. In addition to these methods, some recent works also considered the relationship between these sampling strategies. Multicriteria active deep learning [17] selected informative samples by considering multiple criteria simultaneously (i.e., density, similarity, etc.), and it has achieved excellent performance on the classification tasks. In this article, our focus is to develop an uncertainty AL framework. Compared with the other two types of approaches, the uncertainty-based method has a lower cost when facing large-scale unlabeled data.

The uncertainty AL approach is to select the data with the worst confidence, which is similar to the leak filling in the human learning process. However, because the proportion and difficulty of categories in the training data are different, the learned model will inevitably tend to go overboard on partial categories. For example, if the current learned model has a poor learning effect on the bird category, it will give a high uncertainty to all the data containing birds in the unlabeled

Manuscript received 13 April 2021; revised 6 July 2021, 16 September 2021, 25 November 2021, and 25 March 2022; accepted 18 September 2022. This work is supported in part by the National Natural Science Foundation of China under Grant 62272298, Grant 61872241, and Grant 62077037; in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102; in part by the National Key Research and Development Program of China under Grant 2019YFB1703600; and in part by The Hong Kong Polytechnic University under Grant P0030419, Grant P0042740, and Grant P0035358. *(Wei Huang, Shuzhou Sun, and Xiao Lin contributed equally to this work.) (Corresponding author: Bin Sheng.)*

Wei Huang is with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: 191380039@st.usst.edu.cn).

Shuzhou Sun is with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China (e-mail: 1000479143@smail.shnu.edu.cn).

Xiao Lin is with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China, and also with the Shanghai Engineering Research Center of Intelligent Education and Bigdata, Shanghai 200240, China (e-mail: lin6008@shnu.edu.cn).

Ping Li is with the Department of Computing and the School of Design, The Hong Kong Polytechnic University, Hong Kong (e-mail: p.li@polyu.edu.hk).

Lei Zhu is with ROAS Thrust, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511400, China, and also with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: leizhu@ust.hk).

Jihong Wang is with the School of Physical Education, Shanghai University of Sport, Shanghai 200438, China (e-mail: cylwsy@163.com).

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Pazhou Laboratory, Guangzhou 510335, China (e-mail: philip.chen@ieee.org).

Bin Sheng is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: shengbin@sjtu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3209085.

Digital Object Identifier 10.1109/TNNLS.2022.3209085

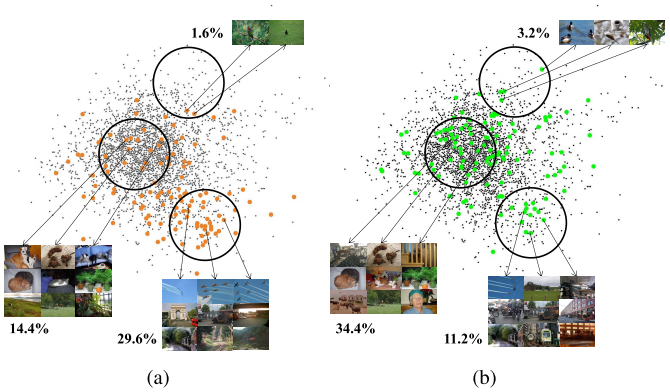


Fig. 1. t-SNE embedding of images on the PASCAL VOC 2007 training set under the task of object detection. The black points are the distribution of the original data, and the orange and cyan points are the sampled data by the uncertainty-based AL method. (a) From the baseline approach (ES [1]). (b) Sampling results of our method.

pool, which is the data bias problem [18]. Fig. 1 illustrates an example of the data bias problem. We use t-distributed stochastic neighbor embedding (t-SNE) [19] to show the distribution of data to be sampled (black dots), and we use orange and cyan dots to represent the sampled data. The baseline approach (i.e., entropy sampling (ES) [1]) sampled a large amount of data in the sparse area (the bottom black circle area) of the unlabeled pool but selected a few samples in the dense area (the middle black circle area). However, Our proposed AL framework resamples the uncertainty data obtained by original uncertainty approaches, and our sampling results are obviously more able to cover the entire unlabeled data pool.

Confidence estimation is a common manner to calculate the uncertainty, and the traditional uncertainty sampling methods, including least confidence (LC) [8], [9], margin sampling (MS) [20], ES [1], [2], etc., also belong to this. However, when the model to be learned is deep neural networks (DNNs), the probability distributions obtained by the traditional uncertainty approaches are too confident, which will lead to the serious data bias problem and thus be even worse than random sampling [18]. To alleviate the data bias problem, recent literature has sought improvement from multiple perspectives. Wasserstein adversarial active learning (WAAL) [21] adopted a Wasserstein distance to refactor the uncertainty calculation method to alleviate the data bias. Ensemble-based AL (ENS) [3] used an ensemble network to calculate data uncertainty. Loss prediction module (LPM) [8] took the unlabeled data as a part of model training to predict the target losses of unlabeled inputs. However, these methods will introduce additional calculation and training costs. Also, they are all highly task-related, which means that they need to be redesigned when facing other tasks.

Unsupervised feature matching does not require any training resources or unbearable computational costs. It can be used to calculate the similarity between unlabeled data [22], [23], [24]. This fact motivates us to use feature matching to compute the similarity in selected uncertainty data for alleviating the data bias problem in the uncertainty AL approaches. To achieve this goal, we first propose unsupervised fusion feature matching (UFFM), which can calculate the data similarity from the perspective of multilayer network features. Then, we design a novel uncertainty calculation method, which resamples uncertainty data obtained by other basic uncertainty AL methods and then removes the redundant ones, thereby significantly alleviating the data bias problem. Our method can be combined with any current uncertainty approaches to improve their performance.

Finally, considering that LC, MS, ES, etc. are only applicable to the classification task, we have also redesigned those methods to make them suitable for the object detection task. In summary, the contributions of our work are threefold.

1) We propose an efficient AL framework, which is to resample the selected uncertainty data based on feature matching to alleviate the problem of data bias. Compared with other methods that focus on this problem, our proposed method has a lower cost and can be combined with all the existing uncertainty methods to improve their performance.
2) We design UFFM, which fuses multiple layers of features to generate descriptors for feature matching. Our approach can perceive the details of the feature more comprehensively, while other similar methods can only perceive very limited information.
3) We improve several uncertainty methods originally designed for the classification tasks such that these uncertainty estimation manners can be adapted for handing complex images in the object detection task. The reason behind is that our uncertainty computation method considers all the objects in the image to calculate their uncertainty, thereby providing a more reliable uncertainty estimation than the original uncertainty methods.

II. Related Work

Here, we first review the most related works about deep-learning-based feature matching, which can be roughly classified as the supervised-based and unsupervised-based approaches. Then, we present the existing AL methods.

A. Deep-Learning-Based Feature Matching

Early descriptors are often hand-designed [25], [26]. Recently, many researchers focused on developing deep-learning-based methods for learning features due to their impressive performance in the diverse vision tasks. Here, we mainly review the deep-learning-based approaches, including the supervised approaches and unsupervised approaches.

1) Supervised Approaches: To solve the overfitting issue caused by the lack of training data, HashGAN [27] synthesized nearly real images to augment the training set and thus could obtain high-quality descriptors for image matching. Deep spherical quantization (DSQ) [28] used a convolutional neural network (CNN) to generate supervised and compact descriptors for image matching. Meanwhile, DSQ forced the network to leverage an L_2 normalization to alleviate the negative effect of norm variance. Deep product quantization (DPQ) [29] introduced a dictionary-based representation to ensure a more accurate image matching and classification

under maintaining an affordable computational complexity and memory. Shen *et al.* [30] added two additional fully connected layers onto the top of the backbone network to obtain binary descriptors, and it showed that a simple network can also extract effective semantic information through a proper design. Although the supervised-based approaches have achieved remarkable performance, collecting large-scale labeled datasets to train the feature matching network is a challenging task, especially for specific fields [25]. Meanwhile, the supervised-based approaches require prohibitive training costs but still have an overfitting risk.

2) Unsupervised Approaches: Unsupervised feature matching has drawn widespread attention in recent years due to its low costs on the training data. Similarity-adaptive deep hashing (SADH) [31] alternatively proceeded with three training modules (i.e., deep hash model training, similarity graph updating, and binary code optimization), which helped to update the similarity graph matrix more effectively than the traditional methods. DistillHash [32] obtained the descriptor through the relationship between the initial signals learned from local structures and the semantic similarity labels assigned by the optimal Bayesian classifier. Deep variational binaries (DVBs) [33] introduced the conditional auto-encoding variational Bayesian networks to exploit the feature space structure of the training data using the latent variables better to unveil the intrinsic structure of the whole sample space. However, these unsupervised methods often require a tedious redesign when facing different models, and it is difficult to guarantee that the intrinsic structure of the whole sample space can be obtained when facing different datasets. Instead, part-based weighting aggregation (PWA) [34] proposed a pure unsupervised feature matching method, which directly aggregated the features of the pretrained model. However, from their experiments, we find that this type of method cannot fully perceive the features when the difference between the pretraining dataset and the images to be matched was large. For addressing this problem, in this work, we propose to fuse the middle layer and the lower layer features to get a more complete descriptor for images.

B. AL Approaches

1) Uncertainty-Based AL Approaches: As one of the most commonly used methods in AL, the uncertainty-based methods are prone to data bias problems when facing large-scale data or DNNs. Apart from the above classical approaches, some recent advanced methods have achieved different performance improvements from multiple perspectives. Sparse modeling active learning (SMAL) [11] combined uncertainty, diversity, and density via sparse modeling to alleviate the data bias problem. However, sparse representation is challenging to guarantee stability when facing large-scale data. Batch mode active learning (BMAL) [12] started with a feature descriptor extraction coupled with a divergence matrix to alleviate the problem of redundancy between unlabeled points. However, the traditional feature extraction method used in BMAL can hardly contribute to DNNs. LPM [8] was jointly trained with the target model to predict the target loss of unlabeled inputs, but it increases the costs of network training. Localization-aware active learning (L-Aware) [4] proposed a localization tightness and localization stability to calculate the uncertainty. However, this work requires the network to provide intermediate prediction results (e.g., predictions by region proposal network (RPN) in Faster R-CNN [35]), which means that this method cannot be used in a model without intermediate prediction (e.g., one-stage object detector [36], [37], [38]). The ensemble-based method [3] used five committee networks to calculate uncertainty, which tends to be impractical in the existence of large-scale unlabeled data and DNNs. In this article, we use a feature matching algorithm to resample uncertain data obtained by the uncertainty AL methods. Our work is based on a pretrained model and thus does not introduce any training costs. Meanwhile, our method can be combined with any existing uncertainty AL methods and thus has an excellent generalization ability.

2) Diversity-Based AL Approaches: The diversity-based approaches preferred to select the batch of data with the most dispersed feature distance. Patra and Bruzzone [13] used a kernel k-means clustering algorithm to minimize the redundancy and keep the diversity among these samples after selecting a batch of uncertain samples. Yang *et al.* [39] regarded AL as a discrete optimization problem, and they imposed a diversity constraint on the objective function to make the selected data as diverse as possible. The Core-set approach [40] improved the competitiveness of the selected data by constructing a core subset. Although these methods have shown to be effective for simple and low-scale features, our empirical analysis suggests that they do not scale to learn more complex and large-scale features. We will compare this type of method and prove our point in the experiment.

3) Expected Model Change AL Approaches: Expected model change approaches take processed unlabeled data (e.g., adding noise) as inputs to observe the changes in the model outputs. Settles *et al.* [41] estimated the value of unlabeled data by measuring the changes in model parameters, but it ignored the underlying data distribution. To address this issue, Freytag *et al.* [42] directly calculated the expected change in model predictions and marginalized the unknown label. Furthermore, Käding *et al.* [43] proposed a new generalization of the expected model output change principle, and thus this expected model change AL approach can be used in DNNs. However, compared with the uncertainty-based methods, this kind of method has a higher cost, especially when faced with DNNs and large-scale unlabeled data.

III. Our Method

This section presents the implementation details of our proposed framework. First, we will introduce our UFFM, which can remove similar data in the sampling results obtained by the uncertainty AL approaches. Then, we will propose a novel uncertainty calculation technology coupled with UFFM to calculate the uncertainty of unlabeled data. The former can obtain the data with higher uncertainty, while the latter can further eliminate the data bias in the above-selected data. We finally introduce the implementation details of the proposed AL framework based on special tasks, including

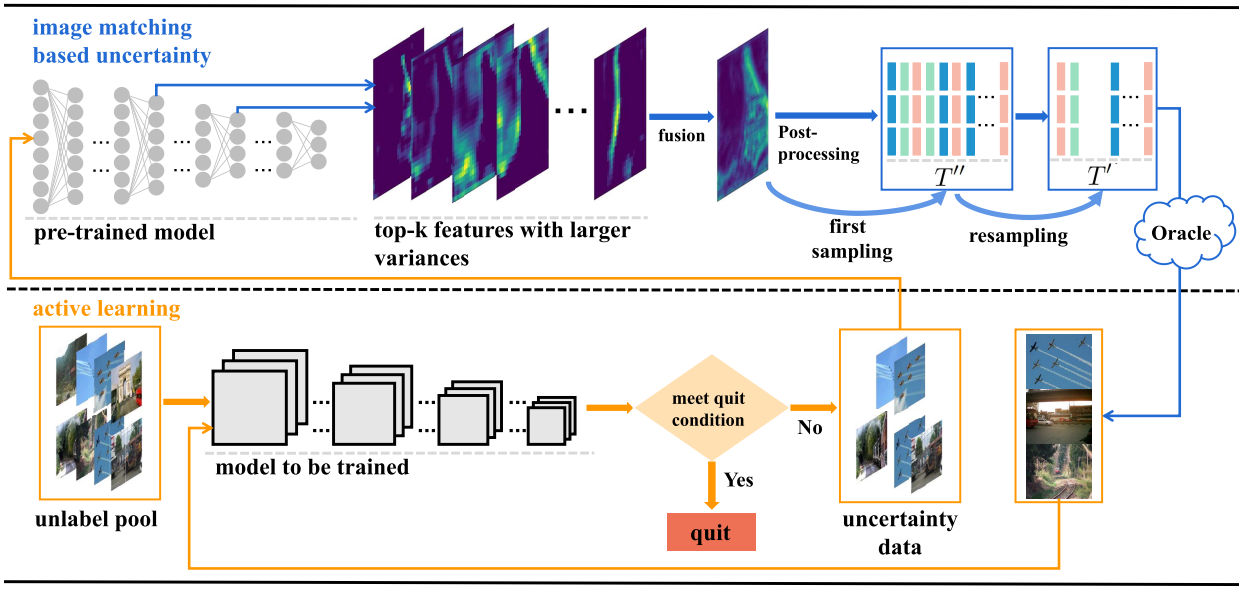


Fig. 2. AL framework consisting of UFFM and feature-matching-based uncertainty. Our framework first selects a certain amount of uncertainty data (more than the expected of AL) from the unlabeled pool through a basic uncertainty approach. Then, we use UFFM to resample the above-selected data to remove the similar data. The quit condition is the label budget is exhausted or the expected performance of the model is reached.

the classification task and the object detection task. Fig. 2 shows the schematic illustration of our proposed deep AL method.

A. Unsupervised Fusion Feature Matching (UFFM)

1) Selection of Layers: For the DNN models, the lower layers often detect the surface features of objects (e.g., edges, textures, shapes), while the higher layers reflect more abstract information (e.g., classification), and this has also been concluded in many works [4], [34]. Unlike the existing methods that only use high-layer features of the pretrained model for unsupervised feature matching, we choose features at the lower and middle layers in this article. Specifically, for the l-layers' model $\mathcal{L}_{i=1}^{l}$, we select the i'th layer $\mathcal{L}_{i'}$ and the i''th layer $\mathcal{L}_{i''}$ for unsupervised feature matching, where $1 \leq i' \leq i'' \leq l$, i', and i'' are determined according to the used model.

Such selection of layers is mainly based on two reasons. The first reason is that the high layer of the pretrained model tends to provide very limited useful features, because the pretraining dataset and the images to be matched may be very different, and the pretrained model cannot recognize the abstract features of the images to be matched. To prove our point, we show the inference results of images in Oxford5k through the model pretrained by ImageNet [44], where the similarity between Oxford5k and ImageNet is very limited. Fig. 3 shows that the high layer of the pretrained model can hardly detect the abstract features of the image to be matched. Another reason is that the neural network has information loss during downsampling, and thus the selection of both the lower and middle layers at the same time helps obtain complete features of the image to be matched. We will further discuss the benefits of fusing different layers in more detail in the ablation study.

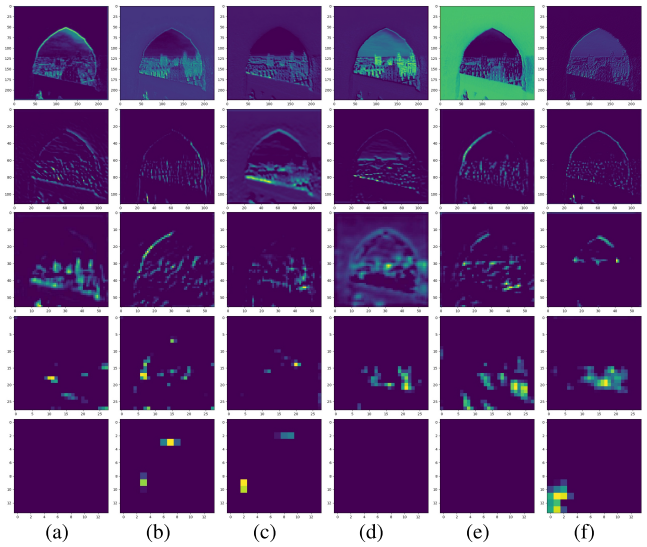


Fig. 3. Feature visualization of different channels in different layers. From top to bottom are block1_conv2, block2_conv2, block3_conv3, block4_conv3, and block5_conv3. The used model is Visual Geometry Group (VGG)16 pretrained on ImageNet, and the test image is selected from Oxford5k. (a) 8th. (b) 16th. (c) 24th. (d) 32th. (e) 40th. (f) 48th.

2) Selection of Channels: For the above-selected layers, we only choose partial channels for feature matching. It has three main considerations.

1) Feature maps of the DNN models usually have many channels, and the direct use of all these channels for feature matching undoubtedly requires a very large computational cost.

2) As high-dimensional features, more channels are more likely to contain noise, which is obviously not expected for the feature matching task.

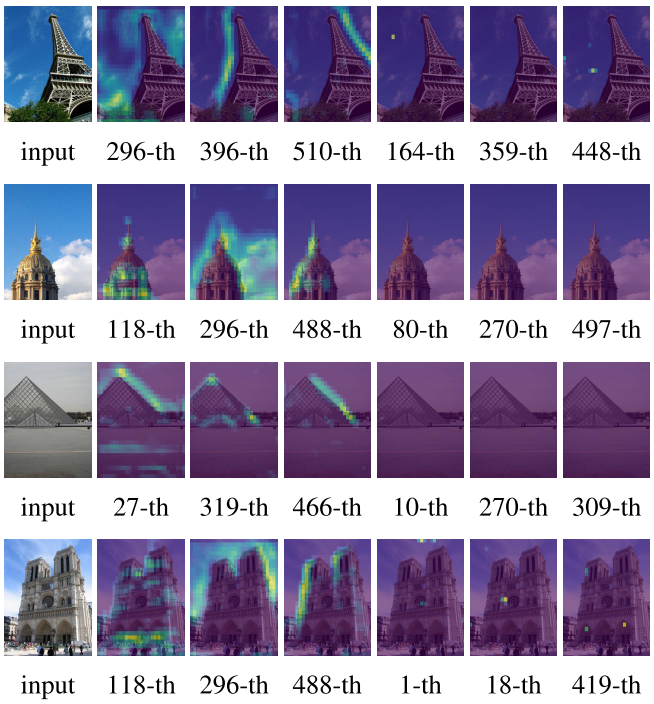


Fig. 4. Channels with different variances. Two to four columns are the three channels with the largest variance, and the last three columns are the ones with the smallest variance. The pretrained model used here is VGG16 trained by ImageNet, and the input images are selected from Oxford5k.

3) Much previous literature [34] argues that channels with larger variances are more discriminative. To verify this view, we visualize the channels with different variances, as shown in Fig. 4. From these visualization results, we can find that channels with larger variances can often detect fixed information (see from two to four columns), while channels with small variances can hardly reflect any features (see the last three columns). Therefore, this article only uses channels with larger variances for feature extraction. For the selected layer $\mathcal{L}_l$, we suppose that its shape is $h_l \times w_l \times c_l$, where c_l is the number of channels, and h_l and w_l are the height and width of the channels, respectively. We first calculate the variances of all the channels $\mathcal{V}_l = v_1, v_2, \ldots, v_c$ in $\mathcal{L}_l$, where the ith channel variance v_i can be calculated as

$$v_i = \frac{1}{m \times n} \sum_{m=1}^{h_l} \sum_{n=1}^{w_l} \left(x^i_{(m,n)} - \overline{x^i} \right)^2 \tag{1}$$

where $x^i_{(m,n)}$ is the value of position (m,n) at the ith channel. $\overline{x^i}$ is the average variance of the ith channel, and it can be calculated by

$$\overline{x^i} = \frac{1}{m \times n} \sum_{m=1}^{h_l} \sum_{n=1}^{w_l} \left(x^i_{(m,n)} \right). \tag{2}$$

Then we sort the variances of all the channels in descending order and select the top k channels for unsupervised feature matching. k is determined by the used model, and we will discuss it in more detail in the ablation study.

3) Feature Fusion: Since a single layer cannot extract features completely, our UFFM uses the lower and middle layer channels to describe the features. In recent years, many works have used the features of convolutional layers for feature matching. Compared with the fully connected layers, the convolutional layers are more interpretable. However, instead of using channels of a single convolutional layer, we also choose that of the lower layer to ensure the completion of image description. Pooling layers are pooled from the convolutional layer, so it can keep the features of the convolutional layer. Meanwhile, the relationship between the lower pooling layer and the higher convolutional layer of the current DNN is often multiple, so fusion of these two types of layers will be reasonable and simple.

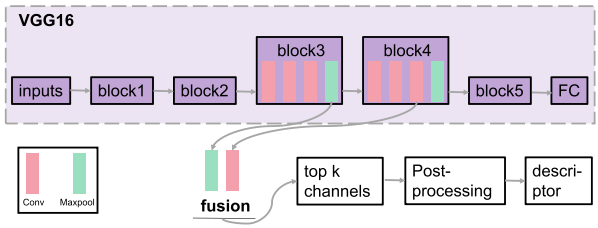


Fig. 5. Details of feature fusion. The network we used in this article is VGG16, and the fused features are block3_pool and block4_conv3.

For the selected lower pooling layer $\mathcal{L}_{l'}$ and the middle convolutional layer $\mathcal{L}_{l''}$, we suppose that their shapes are $h_{l'} \times w_{l'} \times c_{l'}$ and $h_{l''} \times w_{l''} \times c_{l''}$, respectively, where $h_{i'}/h_{i''} = w_{i'}/w_{i''} = N$, $N \in (1, 2, 3, \ldots)$. For these two selected layers, we first pool $\mathcal{L}_{l'}$ to the same size as $\mathcal{L}_{l''}$. Then we select the corresponding meaningful channels from the above layers and aggregate them. Sum-pooled convolutional features (SPoC) [23] argued that the aggregation of the convolution layer channels could be directly based on a sum pooling aggregation, without the need to use a fusion method such as Fisher vector and triangular embedding, since the convolutional layer features of DNNs were sufficiently discriminative. This article also does not use the traditional aggregation method. We adopt the approach of a bitwise addition to aggregate $\mathcal{L}_{l'}$ and $\mathcal{L}_{l''}$ into the shape of $h_{l''} \times w_{l''} \times (c_{l'} + c_{l''})$. We show the details of feature fusion in Fig. 5. We will further prove the effectiveness of this design in ablation study, especially the selection of fused features.

4) Postprocessing of Fused Features: For the image I, we can get its fused features $f_I \in \mathbb{R}^{h_{l''} \times w_{l''} \times (c_{l'} + c_{l''})}$ through the above steps. We first perform an l_2-normalization on f_I to obtain a new feature map, $f_{I'} = (f_I / \| f_I \|_2)$. Then, we use principal component analysis (PCA) to reduce the dimension of the normalized features and denote the result $f_{\mathcal{I}}$, $f_{\mathcal{I}} \in \mathbb{R}^{h'_{l''} \times w'_{l''} \times (c_{l'} + c_{l''})'}$, where $h_{l''}{}'$, $w_{l''}{}'$, and $(c_{l'} + c_{l''})'$ can be adjusted according to the expected complexity.

B. Statement for Data Bias in AL

Let $Q_{(x,y)}$ and $P_{(x,y)}$ denote the distribution of the unlabeled data pool and the selected data obtained by an AL method, respectively, and suppose their densities are $q(x, y) = q(y \mid x)q(x)$ and $p(x, y) = p(y \mid x)p(x)$, respectively. We use $\mathcal{H}(h \sim H)$ to represent the optimal sampling for the original distribution H under the condition of a given sampling rate (SR), where h obeys the distribution H. Based on this

definition, $\mathcal{H}((x, y) \sim P_{(x,y)})$ can be calculated as

$$\mathcal{H}((x, y) \sim Q_{(x,y)}) = -\iint q(y \mid x)q(x)\ln(q(y \mid x)q(x))d_x d_y. \quad (3)$$

$$\mathcal{H}((x, y) \sim P_{(x,y)}) = -\iint q(y \mid x)q(x)\ln(\mathrm{p}(y \mid x)p(x))d_x d_y. \quad (4)$$

We then use the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(Q_{(x,y)} \mid\mid P_{(x,y)})$ to describe the extent to which $P_{(x,y)}$ covers $Q_{(x,y)}$

$$D_{\mathrm{KL}}(Q_{(x,y)} \mid\mid P_{(x,y)}) = \mathcal{H}((x, y) \sim P_{(x,y)}) - \mathcal{H}((x, y) \sim Q_{(x,y)}) = \iint q(y \mid x)q(x)\ln\frac{q(y \mid x)q(x)}{p(y \mid x)p(x)}d_x d_y. \quad (5)$$

Therefore, we can obtain the optimal AL query function $\mathcal{Q}_{\mathrm{AL}}$ by minimizing $D_{\mathrm{KL}}(Q_{(x,y)} \mid\mid P_{(x,y)})$

$$\mathcal{Q}_{\mathrm{AL}} = \arg\min_{P_{(x,y)}} D_{\mathrm{KL}}(Q_{(x,y)} \mid\mid P_{(x,y)}). \quad (6)$$

However, from an example shown in Fig. 1, we can see that $P_{(x,y)}$ is biased toward partial categories in practice. Assuming that the optimal sampling of the training data under given conditions $Q_{\widetilde{\mathrm{AL}}}$. Obviously, in $P_{(x,y)}$, some high uncertainty data $Q_{\widetilde{\mathrm{AL}}} \backslash \mathcal{H}((x, y) \sim P_{(x,y)})$ are not queried by $\mathcal{Q}_{\mathrm{AL}}$, but some low uncertainty data $\mathcal{H}((x, y) \sim Q_{(x,y)}) \backslash Q_{\widetilde{\mathrm{AL}}}$ are selected instead.

C. Feature-Matching-Based Uncertainty

The existing uncertainty-based AL approaches are prone to suffer from the data bias problem since the learned models often have a preference for partial data. To alleviate this problem, we propose a novel uncertainty method, which improves the original uncertainty approach via feature matching to get the uncertainty of unlabeled data. Specifically, we first use the original uncertainty approach to get the uncertainty of the unlabeled data. Then, we use the UFFM proposed in this article to resample the selected data by the original approach to alleviate the data bias problem.

Let $D_{t=1}^{T}$ denote the unlabeled dataset, and T is the amount of data. Then, the original AL method selects fixed number of data to support model training

$$\mathcal{D}_{t=1}^{T'} = \mathcal{U}\left(U_{(D_{t=1}^{T})}, T'\right) \quad (7)$$

where U is the original uncertainty approach, which can calculate the uncertainty of unlabeled data based on the current learned network. $\mathcal{U}$ means to select T' data from $D_{t=1}^{T}$ according to the uncertainty. $\mathcal{D}_{t=1}^{T'} = \{d_1, d_2, \ldots, d_{T'}\}$ is the selected data, and T' is the expected number of data to be selected at the current stage. Unlike the original approaches, our proposed method first selects T'' data from the unlabeled pool

$$\mathcal{D}_{t=1}^{T''} = \mathcal{U}\left(U_{(D_{t=1}^{T})}, T''\right) \quad (8)$$

where $T'' > T'$. $D_{t=1}^{T''} = \{d_1, d_2, \ldots, d_{T''}\}$ denotes the selected T'' unlabeled data. All the data in $D_{t=1}^{T''}$ are sorted according to their uncertainty scores. Taking the jth data d_j in $D_{t=1}^{T''}$ as an example, we calculate the similarity through UFFM

$$M_{\mathrm{d}_j} = \mathrm{UFFM}\left(d_j, d_{(j+1)\sim T''}\right) \quad (9)$$

where $d_{(j+1)\sim T''} = [d_{j+1}, d_{j+2}, \ldots, d_{T''}]$. $\mathrm{UFFM}(d_j, d_{(j+1)\sim T''})$ can calculate the similarity between $d_j \in \mathbb{R}^{1\times\mathcal{D}_s}$ and $d_{(j+1)\sim T''} \in \mathbb{R}^{(T''-j)\times\mathcal{D}_s}$, where $\mathcal{D}_s$ is the dimension of the descriptor. Specifically, we obtain the distance through matrix multiplication $d_j \times [d_{(j+1)\sim T''}]^{\top}$, and the shape of the result is $\mathbb{R}^{1\times(T''-j)}$. We binarize the similarity by marking 10% data in $d_{(j+1)\sim T''}$ with the smallest distance as similar and the others as dissimilar. Finally, we add the data that are not similar to d_j to the AL results $\mathcal{S}_{t=1}^{T'}$. For all the data in $\mathcal{D}_{t=1}^{T''}$, we perform the above steps in turn until the amount of data reaches the expected number T'. We use the pseudocode to show our proposed sampling strategy; see Algorithm 1 for more details.

Algorithm 1 Feature-Matching-Based Uncertainty

Input: $D_{t=1}^{T}$, T', T'', ($T' < T''$)
Output: $\mathcal{S}_{t=1}^{T'}$

```
Compute the uncertainty of D_{t=1}^{T}, and select T'' data,
    D_{t=1}^{T''} = U(U_{(D_{t=1}^{T})}, T'');
q = 0
for m = 1 to T'' do
    Add d_m to the set of S_{t=1}^{T'};
    Compute the similarity,
  M_{d_m} = UFFM(d_m, d_{(m+1)~T''});
  Mark the data in d_{(m+1)~T''} as similar or dissimilar
according to M_{d_m};
    for n = m + 1 to T'' do
        if d_m and d_n are dissimilar then
            Add d_n to the set of S_{t=1}^{T'};
            q = q+1;
        if q ⩾ T' then
            break;
    end for
end for
```

In addition, for an actual application (e.g., image classification or object detection) of AL, T' is often a fixed value. However, T'' is a hyperparameter in our proposed framework. Here, we define SR, $\mathrm{SR} = (T''/T')$. Obviously, $\mathrm{SR} = 1$ means that our framework degenerates to the original uncertainty methods. However, if SR is large, the uncertainty of the data selected by our method may be lower. We set $\mathrm{SR} = 1.2$ in this article (i.e., $T'' = 1.2 \times T'$), and the influence of SR has been further discussed in the ablation study.

D. Further Design

We will verify our proposed method on image classification and object detection. Because these two tasks cover classification and regression, the generalization of our proposed method can be fully illustrated.

1) Further Design for Image Classification: LC, MS, and ES are the classic uncertainty methods for the image classification task. LC calculates the uncertainty of unlabeled data

through the maximum predicted probability, while MS and ES consider the first two and all probabilities, respectively. For the image I in the dataset with c classes, LC uncertainty I_{LC}, MS uncertainty I_{MS}, and ES uncertainty I_{ES} can be calculated as follows:

$$\mathrm{I_{LC}} = (1 - \widehat{C}), \quad \text{s.t. } \widehat{C} = \underset{i\in[1,\ldots,c]}{\arg\max}(p_i) \tag{10}$$

$$\mathrm{I_{MS}} = (\widehat{C} - \widehat{C}'), \quad \text{s.t. } \widehat{C} = \underset{i\in[1,\ldots,c]}{\arg\max}(p_i),\ \widehat{C}' = \underset{i\in[1,\ldots,c]\setminus\widehat{C}}{\arg\max}(p_i) \tag{11}$$

$$I_{\mathrm{ES}} = \sum_{i=1}^{c} p_i \log(p_i) \tag{12}$$

where p_i represents the confidence of the ith class. Furthermore, we can use Algorithm 1 to obtain feature-matching-based uncertainty.

2) Further Design for Object Detection: The above three classic uncertainty methods do not consider the situation that an image may contain multiple objects to be recognized in object detection. Hence, we redesign these three methods to make them more suitable for object detection. Our redesigned methods include redesigned LC (RLC), Redesigned MS (rms), and redesigned ES (RES), and their uncertainties I_{RLC}, I_{rms}, and I_{RES} can be calculated as follows:

$$\mathrm{I_{RLC}} = \sum_{j=1}^{n_I}(1 - \widehat{C}_j), \quad \text{s.t. } \widehat{C}_j = \underset{i\in[1,\ldots,c]}{\arg\max}\left(p_i^j\right) \tag{13}$$

$$\mathrm{I_{rms}} = \sum_{j=1}^{n_I}\left(\widehat{C_j} - \widehat{C_j}'\right), \quad \text{s.t. } \widehat{C_j} = \underset{i\in[1,\ldots,c]}{\arg\max}\left(p_i^j\right)$$
$$\widehat{C_j}' = \underset{i\in[1,\ldots,c]\setminus\widehat{C_j}}{\arg\max}\left(p_i^j\right) \tag{14}$$

$$I_{\mathrm{RES}} = \sum_{j=1}^{n_I}\sum_{i=1}^{c} p_i^j \log\left(p_i^j\right) \tag{15}$$

where n_I is the number of objects in image I. p_i^j represents the confidence that the jth prediction box in the image is the ith class. Similarly, we use these redesigned methods coupled with Algorithm 1 to obtain feature-matching-based uncertainty. Apart from the above three classic uncertainty methods, we also compare the state-of-the-art (SOTA) uncertainty approaches; see experiments for more details.

IV. Experimental Results

In this article, we propose a method that can alleviate the data bias problem in AL. To verify the effectiveness of our work, we have conducted many experiments. First, we introduce the datasets for feature matching, classification, and object detection. Next, we verify our proposed UFFM on the task of feature matching. Then, we prove that our proposed AL framework can achieve competitive performance on the task of classification and detection. Finally, we discuss the design details of the proposed framework and its advantages in mitigating data bias through the ablation study.

TABLE I

Performance Comparison Between Pure Unsupervised Feature Matching Approaches (P) and Fine-Tuning-Based Image Matching Approaches (F) Under Different Feature Descriptors (d). The Pretrained Model Here Is VGG16, and the Selected Fusion Features Are Block3_Pool and Block4_Conv3

method	d	Oxford5k		Paris6k	
		P	F	P	F
NetVLAD [24]	128	55.5	63.5	64.3	73.5
MAC [45]	128	55.7	76.8	70.6	78.8
ours	128	56.2	-	70.9	-
SPoC [23]	256	53.1	-	-	-
R-MAC [45]	256	56.1	78.2	72.9	83.5
RVD-W [46]	256	60	-	-	-
Razavian et al. [22]	256	-	67	-	53.3
ours	256	62.5	-	73.3	-
CroW [47]	512	70.8	-	79.7	-
InterActive [48]	512	65.6	-	79.2	-
PWA [34]	512	72	87.8	82.3	94.9
CNNBoW [45]	512	-	79.7	-	83.8
ours	512	73.2	-	84.5	-

A. Datasets

1) Datasets for Feature Matching: **Oxford5k [52].** The Oxford buildings dataset consists of 5026 images collected from Flickr by searching particular Oxford landmarks. This dataset includes 11 different landmarks, and each landmark contains five query images with ground truth. For landmarks in each image, it contains one of the following possible labels.

1) Good. It means that the landmark is very clear.

2) OK. It represents that at least 25% of the landmark is clear.

3) Bad. This means the landmark is not present.

4) Junk. It represents that less than 25% of the landmark is visible.

Paris6k [53]. The Paris dataset consists of 6412 images collected from Flick by searching Paris landmarks. The label format of this dataset is the same as Oxford5k. Both Oxford5k and Paris6k are standard datasets for evaluating feature matching methods.

2) Dataset for Classification: **CIFAR-10 [54].** The CIFAR-10 dataset consists of 60k color images with ten categories, (6k images per category). The dataset has 50k training images and 10k test images. **Fashion-MNIST [55].** Fashion-MNIST is a fashion product dataset, including 60k training images and 10k test images. The dataset has ten categories, which are more difficult than the original MNIST dataset. Both CIFAR-10 and Fashion-MNIST are classic classification datasets.

3) Dataset for Object Detection: **PASCAL VOC 2007 [56].** VOC 2007 contains 20 object categories, and it includes 2.5k training images, 2.5k validation images, and 5k test images. **PASCAL VOC 2012.** VOC 2012 is an augmented version of VOC 2007, which contains about 5k training images and 5k validation images. VOC 2007 and VOC 2012 are standard datasets commonly used for vision tasks, including classification, detection, segmentation, etc.

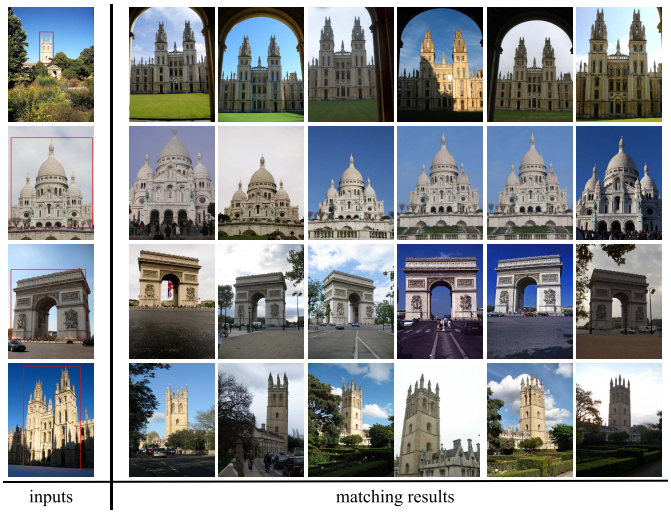


Fig. 6. Matching results by UFFM. The query images and results in the first two lines are from Oxford, and the other lines are from Paris. The pretrained model here is VGG16, and the selected fusion features are block3_pool and block4_conv3.

B. Evaluation of Our Proposed UFFM

Our UFFM is a feature matching technology, which can cooperate with the original uncertainty approaches to calculate the uncertainty of the unlabeled data to alleviate the data bias problem. We follow the evaluation protocol of Oxford and Paris to crop the image with the provided bounding box. The matching results by our UFFM are shown in Fig. 6.

Meanwhile, we report the quantitative matching results under the metric of mean Average Precision (mAP) in Table I. Here, we mainly compare two lines of methods using pure unsupervised feature matching [23], [24], [34], [45], [47] and starting with unsupervised coupled with fine-tuning [22], [34], [45]. From the quantitative results in Table I, we have the following observations.

1) Our UFFM outperforms the existing pure unsupervised feature matching in all the descriptor dimensions. We argue that this mainly benefits from reasonable channel selection and fusion, and we will prove this point in the ablation study.

2) Although our method is slightly inferior to the fine-tuning-based approaches, our training and matching costs are far less. Meanwhile, considering that our method is mainly designed for uncertainty calculation of AL, we believe that this small gap does not significantly affect AL.

3) Generally, the fine-tuning-based methods outperform pure unsupervised methods. It shows that the pretrained model cannot fully sense the features of the image to be matched, especially the abstract features. This is consistent with the conclusion we observed in Fig. 3. Therefore, this further illustrates the rationality of our abandonment of the high layer that can only provide limited features.

4) We argue that our method is better than other methods in helping to alleviate the data bias problem, because our method performs more complete feature matching through feature fusion, and these are exactly the features that the model wants to learn.

TABLE II

RESULTS UNDER METRIC 2 (i.e., LABELING BUDGET UNDER EXPECTED PERFORMANCE) ON FASHION-MNIST. SIMILARLY, WE REPEAT EACH EXPERIMENT FIVE TIMES AND REPORT THE AVERAGE BUDGET

method	InceptionV3 [49]			ResNet-50 [50]			MobileNetV3 [51]		
	0.85	0.9	0.95	0.85	0.9	0.95	0.85	0.9	0.95
random	3.4	4.4	5.2	4.2	5.2	5.8	3.6	4.6	5.2
LC [9]	3.0	4.0	4.8	3.8	5.0	5.6	3.2	4.0	4.8
MS [20]	2.8	3.8	4.8	3.6	4.8	5.4	3.0	3.8	4.8
ES [1]	2.8	3.6	4.6	3.6	4.6	5.4	3.0	3.6	4.8
Käding et al. [43]	2.8	3.6	4.4	3.6	4.4	5.4	2.8	3.6	4.6
SMAL [11]	2.8	3.4	4.4	3.6	4.4	5.4	2.8	3.6	4.4
Core-set [40]	2.6	3.4	4.4	3.4	4.4	5.4	2.8	3.6	4.4
ENS [3]	2.6	3.4	4.2	3.4	4.4	5.2	2.8	3.6	4.2
FKSS [18]	2.6	3.4	4.2	3.2	4.2	5.2	2.6	3.4	4.2
ours	2.0	3.0	4.0	3.0	4.0	5.0	2.0	3.0	4.0

That is, our method will examine the feature similarity among unlabeled data more comprehensively when calculating the uncertainty. Instead, other approaches only match the very limited features provided by the high layer.

C. Evaluation of Our AL Method on the Image Classification Task

1) Experimental Setting: We explore three classic classification networks including InceptionV3 [49], ResNet-50 [50], and MobileNetV3 [51] as evaluation models of AL. For all THE three classification networks, their hyperparameters are: optimizer is stochastic gradient descent (SGD), weight_decay = 0.00004, decay factor of learning rata = 0.94, learning_rate = 0.01, momentum = 0.9, and batch size = 32. The basic settings of AL are: the pretrained model is VGG16, and the descriptor length is 256. We evaluate different AL methods using the following two metrics.

Metric 1: performance under fixed labeling budget. A larger score under Metric 1 indicates a better AL method. If the selected data require a massive amount of labeling costs, this AL algorithm loses its meaning. Therefore, we set the labeling budget for all THE AL frameworks to be less than 50% of the original unlabeled data for fair comparisons.

Metric 2: labeling budget under expected model performance. The less the labeling costs, the better the sampling approaches. Hence, a smaller score under Metric 2 indicates a better AL framework. The expected performance should be adjusted according to the model used; refer to Table II for details.

2) Compared Methods: We compare our proposed method against the following baseline methods.

1) Random sampling: sampling the data uniformly at random from the unlabeled set.

2) Classical uncertainty approaches: LC [9], MS [20], and ES [1].

3) Recent SOTA methods: Patra and Bruzzone [13] used a kernel k-means clustering algorithm to minimize the redundancy and kept the diversity among these samples after selecting a batch of uncertain samples. To highlight the advantages of our method in alleviating data bias, the baseline method [13] used the same uncertainty calculation and feature extraction

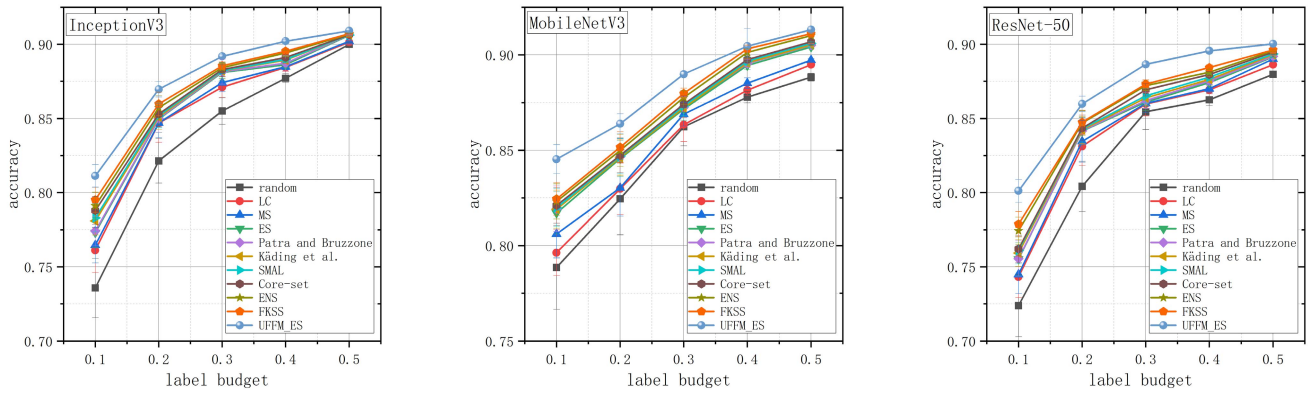


Fig. 7. Comparisons under Metric 1 (i.e., performance under fixed labeling budget) on CIFAR 10. The pretrained model we used here is VGG16 trained by ImageNet. We repeat each experiment five times and report standard deviation by error bar. The evaluation networks used here are InceptionV3 [49], ResNet-50 [50], and MobileNetV3 [51]. The compared baselines included classic uncertainty methods (LC [9], MS [20], ES [1]), recent uncertainty-based methods (ENS [3], Patra and Bruzzone [13], SMAL [11], and FKSS [18]), recent diversity-based methods (Core-set [40]), recent expected model change methods (Käding *et al.* [43]), and random sampling. The basic uncertainty approach combined with UFFM is ES.

mechanism as our method. The only difference is that Patra and Bruzzone [13] obtained the final sampling results through the clustering algorithm, while our proposed method uses a feature matching algorithm. Note that unlike the original article [13], we use k-means++ as the clustering algorithm and run it multiple times to get better performance. Ensemble-based AL (ENS) [3] used an ensemble network to calculate data uncertainty. Note that the training data in the first stage of this original article are generated on the basis of labels, and the purpose is that the initial sets are balanced over all the classes. But the real raw unlabeled data will not have any label information, so our framework is purely random sampling in the first epoch. For the sake of fairness, the data of the first epoch of ENS in this article are also a purely random sampling. Fisher kernel self-supervision (FKSS) [18] proposed a low-complexity feature density matching method and used it to calculate the uncertainty of unlabeled data. Similarly, for fairness, we use the complete standard dataset when comparing this method, rather than using only partial data of the dataset in the original article to create artificial data. The Core-set approach [40] is a diversity-based AL technology, and we follow the training tricks and hyperparameters in the original article. Käding *et al.* [43] is an expected model change AL technology. Following this article [43], we also use a stochastic gradient approximation with just a single sample to estimate model parameter updates, and the models we use are all DNNs to ensure that the baseline can play its advantages, thereby ensuring a fair comparison. SMAL [11] combined uncertainty, diversity, and density via sparse modeling to alleviate the data bias problem. SMAL [11] divided the dataset into a seed set (labeled set), an unlabeled set, a validation set, and a testing set. For a fair comparison, we use the randomly selected data in the first epoch as the seed set and then follow the original article to set other details of this method.

3) Results and Analysis: We conducted experiments on CIFAR 10 and Fashion-MNIST with the above two metrics, including Metric 1 and Metric 2. Our proposed framework and other compared baseline methods follow the same training process. We first randomly sample 5% of the dataset (about 3k images) as the training data for the first epoch. Then for each subsequent epoch, we use the AL framework to sample 5% of the dataset and use it to continue training the model. Finally, for Metric 1, we repeat AL sampling and training until the sampled data reach the fixed labeling budget. For Metric 2, we repeat until the trained model reaches the expected performance. The basic uncertainty approach combined with our proposed method UFFM here is ES. We report the result on CIFAR 10 and Fashion-MNIST in Fig. 7 and Table II, respectively. From these results, we have several observations.

1) Our method outperforms all the baselines by a clear margin. Specifically, our framework has a higher performance under a fixed labeling budget (see Fig. 7), and our framework requires less labeled data under the expected performance (see Table II).
2) Considering that the training data at the first stage are randomly sampled, we repeat each experiment five times and report the standard deviation (error bar in Fig. 7) for comparisons. From the results, we can clearly find that our method is more stable than the compared methods. Note that the stability of AL is related to the degree of data bias in each epoch. Hence, it indicates that our framework is superior to other methods in alleviating the data bias problem. We will prove this in detail in the ablation study.
3) As the fixed labeling budget or expected model performance grows, the improvement of our method over compared methods tends to decrease, as shown in Fig. 7. However, the application scenarios of AL often have a small labeling cost. Hence, we believe that our method should have a superior performance in real cases.
4) Under part expected performance, there will be the epoch gap between our proposed framework and comparing baseline, and our approach uses fewer epochs to achieve sample performance (e.g., our proposed AL framework with MobileNetV3 uses 4.0 epochs while the baseline of random sampling needs 5.2 epochs to meet the expected performance of 0.95). The epoch gap shows

that our method can use less labeling budget, which means that more training resources can be saved.

D. Evaluation of Our AL Method on the Object Detection Task

1) Experimental Setting: Here, we consider multiple detectors, including EfficientDet [36], Faster R-CNN [35], and single shot multibox detector (SSD) [37]. The settings of EfficientDet are: the backbone is EfficientDet-D0, the optimizer is SGD, initial learning rate = 0.08, warmup learning rate = 0.001, warmup steps = 2500, and batch size = 128. The hyperparameters of Faster R-CNN are: the backbone is ResNet-101 [50], optimizer is SGD, weight_decay = 0.00005, learning_rate = 0.0001, and batch_size = 32. The settings of SSD include: the backbone is MobileNetV2 [57], momentum is 0.94, and batch size = 24. Moreover, the metrics used in the image classification task are also adopted in this section.

2) Compared Methods: The following methods are used for comparisons: 1) random sampling; 2) redesigned classical uncertainty approaches: RLC, rms, and RES; and 3) recent SOTA methods: Patra and Bruzzone [13] used a diversity-based AL approach, and the comparison details are described in Section IV-C. LPM [8] trained with the target active model, and it could be used to predict the target loss of unlabeled inputs. We use the same module and model connected to three layers of the target model for all the detectors. The internal structure and module training follow the settings in the original article. Localization-aware (L-Aware) [4] AL method used localization tightness and localization stability to calculate uncertainty. For Faster R-CNN, we use the region proposals provided by its RPN to calculate localization tightness. While for EfficientDet and SSD, we directly calculate localization stability since they do not have an intermediate proposal. BMAL [12] was a batch mode AL technique, which started with a feature descriptor extraction coupled with a divergence matrix to alleviate the problem of redundancy among unlabeled points. We follow most of the experimental details of BMAL [12]. For example, the Gabor filter is applied to the images for feature extraction, and PCA is used to reduce dimensionality. To reduce the computational costs, we follow the original article to use a subsampling strategy. Also, we use BMAL [12] combined with the method we proposed in Section III to make it more suitable for object detection.

3) Results and Analysis: We conduct solid evaluations on our AL framework in terms of the object detection task and use two metrics (i.e., Metric 1 and Metric 2) proposed in Section IV-C for comparisons. For Metric 1, we regard the training set of PASCAL VOC 2007 (about 2.5k images) as the original unlabeled data $\text{VOC}_I^{2.5\text{k}}$. We let 5% of $\text{VOC}_I^{2.5\text{k}}$ as the training data for the first epoch, and in each subsequent epoch, we select 5% of the original unlabeled data as the new training data. The above step is repeated until the labeling budget is exhausted. While for Metric 2, we unite the train set of PASCAL VOC 2007 (about 2.5k images) and PASCAL VOC 2012 (about 5k images) as the original unlabeled data $\text{VOC}_I^{7.5\text{k}}$. We let 5% of $\text{VOC}_I^{7.5\text{k}}$ as the training set for the first epoch, and in each subsequent epoch, we select 5% of the original unlabeled data as the added training data. The above step is repeated until the trained model reaches the expected performance. The results of Metric 1 and Metric 2 are reported in Fig. 8 and Table III, respectively. The basic uncertainty approach combined with our proposed method UFFM here is RES. From the above results, we can conclude that.

1) Our method outperforms all the baselines on the object detection task. Meanwhile, object detectors we used in experiments cover two-stage (Faster R-CNN) and one-stage (EfficientDet and SSD), which shows that our framework is model-agnostic and thus can assist networks with any structure perform AL.
2) From the standard deviation reported in Fig. 8, we find that our method is more stable than all the baselines. We will demonstrate in the ablation study that this advantage is mainly due to the fact that our method can alleviate the data bias problem.
3) As the fixed labeling budget or the expected model performance increases, some uncertainty methods are even worse than random sampling. The main reason is that the problem of data bias has reached a serious degree. When the sampling epochs increase, the gap between our method and other methods is also decreased. However, we think this is normal. As AL progresses, there is

TABLE III

LABELING BUDGET UNDER EXPECTED PERFORMANCE. THE DATASETS WE USED HERE ARE PASCAL VOC 2007 AND VOC 2012. SIMILARLY, WE REPEAT EACH EXPERIMENT FIVE TIMES AND REPORT THE AVERAGE BUDGET

method	EfficientDet [36]			Faster R-CNN [35]			SSD [37]		
	0.7	0.75	0.8	0.7	0.75	0.8	0.55	0.60	0.65
random	2.8	4.0	5.2	3.0	4.0	5.2	3.2	4.2	5.2
RLC [9]	2.4	3.6	4.6	2.8	3.6	4.6	2.8	3.6	4.8
RMS [20]	2.4	3.4	4.4	2.6	3.6	4.6	2.6	3.6	4.8
RES [1]	2.4	3.4	4.4	2.6	3.6	4.4	2.4	3.6	4.6
BMAL [12]	2.4	3.4	4.4	2.4	3.6	4.4	2.4	3.4	4.6
LPM [8]	2.4	3.2	4.2	2.4	3.4	4.4	2.4	3.4	4.4
L-Aware [4]	2.2	3.2	4.2	2.4	3.2	4.2	2.2	3.2	4.4
ours	2.0	3.0	4.0	2.0	3.0	4.0	2.0	3.0	4.0

TABLE IV

PERFORMANCE WITH 256-D DESCRIPTORS UNDER DIFFERENT LAYER SELECTIONS. THE LABELING BUDGET USED HERE IS 0.5, AND THE EXPERIMENT FOLLOWS THE SETTINGS IN FIGS. 7 AND 8. THE PRETRAINED MODEL USED HERE IS VGG16 TRAINED ON IMAGENET. LOWER LAYER, MIDDLE LAYER, AND HIGH LAYER ARE BLOCK3_POOL, BLOCK4_CONV3, AND BLOCK5_POOL, RESPECTIVELY, AND OUR METHOD IS THE FUSION OF LOWER AND MIDDLE LAYERS

method	feature matching		classification	detection			
	Oxford 5k	Paris 6k	ResNet-50	EfficientDet (D0)			
	mAP (%)	mAP (%)	accuracy (%)	mAP (%)	AP_S	AP_M	AP_L
lower layer only	60.9	71.5	88.6	77.8	40.9	82.1	89.1
middle layer only	61.6	72.1	89.1	78.1	43.3	82.7	89.5
high layer only	60.2	70.1	87.4	76.9	25.1	81.6	88.7
ours	62.5	73.3	90.0	78.9	44.9	83.6	90.3

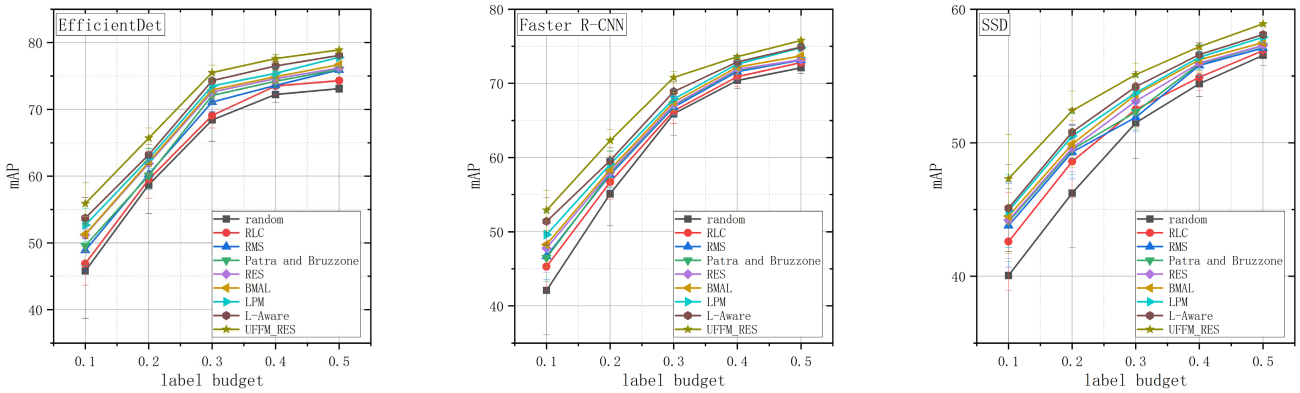


Fig. 8. Performance on PASCAL VOC 2007 under fixed labeling budget. The pretrained model we used here is VGG16 trained by ImageNet. We repeat each experiment five times and report standard deviation by error bar. The evaluation detectors used here are EfficientDet [36], Faster R-CNN [35], and SSD [37]. The compared baselines included classic uncertainty methods (RLC [9], rms [20], and RES [1]), recent SOTA methods (Patra and Bruzzone [13], L-Aware [4], BMAL [12], and LPM [8]), and random sampling. The basic uncertainty approach combined with UFFM is RES.

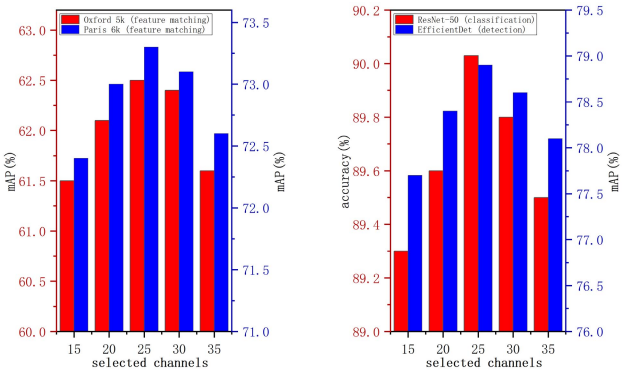


Fig. 9. Matching results with 256-D descriptor under different selected top-k channels with maximum variances. The labeling budget used here is 0.5, and the experiment follows the settings in Section IV.

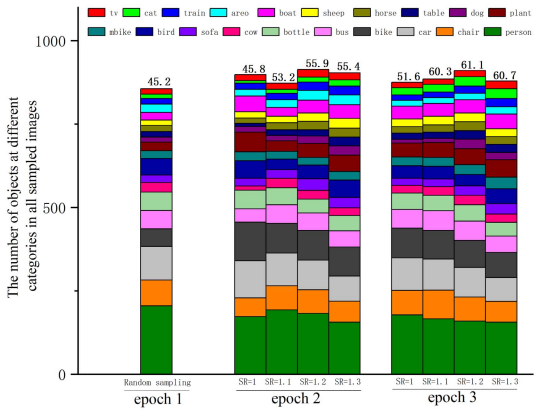


Fig. 10. y-axis denotes the number of objects at each category in all the sampled images under a specific epoch number. Here, in each bar, we also show the corresponding classification accuracy (e.g., 55.9% at the epoch 2 with SR = 1.2) of our method under a specific epoch number and a specific SR. The labeling budget used here is 0.5, and the experiment follows the settings in Figs. 7 and 8. The dataset and learned model are PASCAL VOC 2007 and EfficientDet, respectively. The pretrained model for feature matching is VGG16, and block3_pool and block4_conv3 are the selected features.

less and less valuable data, so the scope for model improvement will be limited.

4) The epoch gap of training in the image classification task can also be observed here.

E. Ablation Study

1) Better Layer Combination: UFFM in this article aims to fuse the features of different layers to perceive richer information. To further demonstrate the effectiveness of this fusion mechanism, we report the performance under different layer combinations in Table IV, and we find that our proposed method outperforms other approaches, especially for small objects. It demonstrates the advantages of our UFFM method in detailed feature perception. Meanwhile, we can also find that our method is the best, in which the middle layer is better than the lower layer, and the high layer is the worst. The main reason is that although lower layers can perceive more features, it also introduces lots of noise. Since the pretraining dataset (ImageNet) is different from the target dataset (Oxford5k, Paris6k, CIFAR 10, PASCAL VOC, etc.), the perceptible features of the high layer are also limited. Note that the results in Table IV are not to deny the advantages of high-layer features in the supervised task. It only shows that in the scene where the intersection of the images to be processed and the pretraining data is small, the high-layer features are weaker than the middle and lower layer features.

2) Suitable Channel Selection: Many previous works have found that the perception ability of a specific channel is relatively fixed, and channels with larger variances perceive more information. In Fig. 9, we report the results under the top-k channels with the largest variances. We find that too small channel numbers cannot perceive complete information, while too larger channel numbers introduce additional noise. Based on the experimental results in Fig. 9, we empirically set the number of channels as 25.

3) More Optimized Sampling Distribution: In Fig. 10, we report the sampling distribution of different object categories and the model performance at different epochs under multiple SRs. Sampling distribution can help us easily compare the changes in category quantity in different epochs,

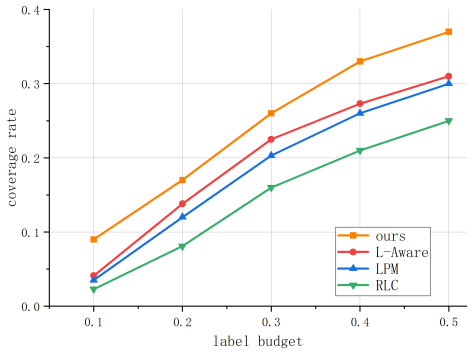


Fig. 11. Coverage rate under fixed labeled budget. The data and detector used here are EfficientDet and PASCAL VOC 2007, respectively, and the settings of the experiment are the same as that in Section IV-D. Feature matching uses UFFM proposed in this article. The compared baselines include classic uncertainty methods RLC [9], recent SOTA methods L-Aware [4], and LPM [8].

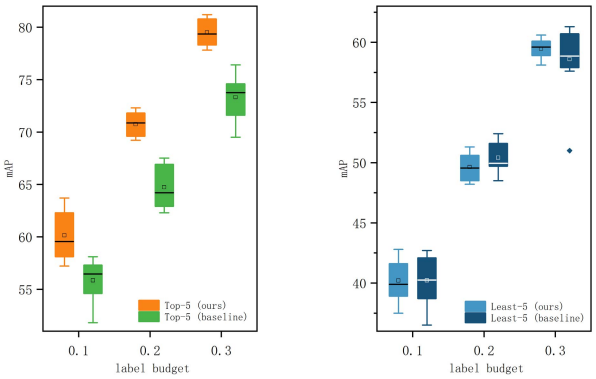


Fig. 12. Performance on PASCAL VOC 2007 under fixed labeling budget. The test set has 20 classes, top-5 represents five classes with the highest mAP, and Least-5 represents five classes with the worst performance. Each experiment is repeated ten times, and the baseline method is ES.

and the model performance can help us choose a better SR. From Fig. 10, we observe that the original uncertainty method (SR $=$ 1) has an obvious data bias problem. For example, compared with the first epoch, the second epoch has a significant increase in the number of samples for bike, plant, boat, etc. According to the principle of the uncertainty AL method, we think that it is because the current model has a poor learning effect on these categories. However, such a direct sampling method ignores the similarity of the internal data of the same classification, which causes the data bias problem. For the same uncertainty unlabeled data, the sampling results of our method obviously do not have the above problems, which means that the increase in the number of samples for bike, plant, boat, etc. does not fluctuate as much as the original method. This is because our proposed method can remove similar images in these image classifications. According to the results reported in Fig. 10, we set up SR $= 1.2$ in this article.

To further quantify the advantages of our work in mitigating the data bias problem, we define the coverage rate here, which can be used to measure the coverage degree of the sampled data to the original data. The coverage rate is $\mathcal{S}/\mathcal{O}$, where $\mathcal{O}$ is the original data and $\mathcal{S}$ is the residual data after removing the similar parts from the sampled data. We report the coverage rate in Fig. 11, which shows that our proposed AL framework yields better sampling results than other methods.

TABLE V

PERFORMANCE UNDER REDESIGNED UNCERTAINTY APPROACHES AND ORIGINAL UNCERTAINTY APPROACHES. THE DATASET USED HERE IS PASCAL VOC 2007. OUA AND RUA ARE THE ORIGINAL UNCERTAINTY APPROACHES AND REDESIGNED UNCERTAINTY APPROACHES, RESPECTIVELY

	method	backbone	labeling budget			
			0.2	0.3	0.4	0.5
OUA	EfficientDet [36]	D0	62.5	73.8	76.2	77.1
	Faster R-CNN [35]	ResNet-101	59.8	68.9	72.7	74.6
	SSD [37]	MobileNetV2	51.1	54.2	56.9	58.2
RUA	EfficientDet [36]	D0	65.7	75.5	77.6	78.9
	Faster R-CNN [35]	ResNet-101	62.3	70.8	73.6	75.8
	SSD [37]	MobileNetV2	52.4	55.1	57.2	58.9

4) More Stable Training Process: Data bias causes an instability problem in the training process, which can also be drawn from Figs. 7 and 8. Here, we will further study the underlying causes of this phenomenon. We think that this is mainly because the data bias incurs that partial categories do not progress steadily. To verify this, we report the learning process in Fig. 12, and we observe that for the baseline method, the progress of the top-5 categories in the next epoch is very limited. This is mainly because the current learned model will focus on the Least-5 categories, which means that the next epoch will be mainly sampled Least-5 categories. Instead, our method will resample the sampling results of the Least-5 categories that are currently focused on. Because what we remove are the data that have a limited help for model learning, we can guarantee the learning of the top-5 categories without reducing the learning effect of the Least-5 categories, thereby ensuring a stable learning process.

5) More Effective Redesigned Uncertainty Approaches: We redesign several uncertainty-based approaches to make them fit the object detection task. Those redesigned approaches consider the uncertainty of all the objects in the image rather than just one object. This improvement is mainly because we have observed that the complexity of the images used for object detection is much greater than image classification, that is, the images used for detection often contain many objects, and the test confidence of these objects is different greatly because of the impact of category, size, etc. We report the performance of our proposed and original uncertainty approaches in Table V, and we find that our redesigned uncertainty approaches are more suitable for the task of object detection.

F. Feature Redundancy

1) Feature Redundancy in Existing Unsupervised Feature Matching: Unsupervised feature matching has attracted lots of attention in recent years. Using pretrained models for feature extraction of images to be matched has almost become a de facto approach. ImageNet is a very large dataset, and its commonly used subset ILSVRC2012 (ImageNet2012) still

TABLE VI

PERFORMANCE AND TRAINING COSTS OF DIFFERENT TRAINING TECHNIQUES. NONPRETRAIN AND PRETRAIN STAND FOR TRAINING FROM SCRATCH AND PRETRAINING ON IMAGENET, RESPECTIVELY. THE LABELING BUDGET USED HERE IS 0.5, AND THE EXPERIMENT FOLLOWS THE SETTINGS IN FIGS. 7 AND 8. NOTE THAT THE PRETRAINED MODELS HERE ARE FROM THE TENSORFLOW GITHUB REPOSITORY, SO THE PRETRAINING COSTS ARE NOT INCLUDED IN THE TRAINING COSTS

		classification (ResNet-50)		detection (EfficientDet (D0))	
		accuracy (%)	train cost (h)	mAP (%)	train cost (h)
Unsupervised (UFFM)		90.0	0.8	78.9	7.5
Supervised	Non-pretrained	90.5	1.3	79.3	14.0
	Pre-trained	91.3	1.1	79.8	12.5

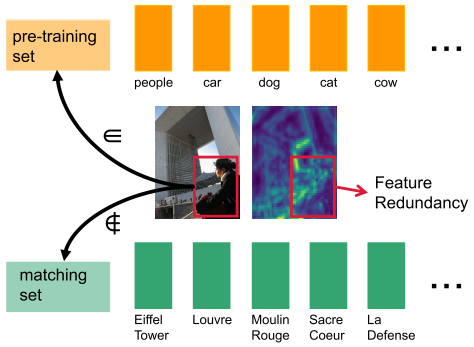


Fig. 13. Instance of feature redundancy. The test image is selected from Paris6k; the feature shown here is the fusion of the top-25 channels with the largest variance in block3_conv3 of VGG16.

has more than one million training images. Currently, many vision models are pretrained on ImageNet, and these pretrained models used in unsupervised feature matching are also usually based on it. Oxford5k and Paris6k are standard datasets used for feature matching. Currently, there are many unsupervised feature matching works that use pretrained models on ImageNet to evaluate the proposed methods on these two datasets.

However, we find in the experiment that the above unsupervised feature matching paradigm has the feature redundancy problem. Feature redundancy is defined here that nonmatching targets in the images to be matched are perceived by the pretrained model, which will have a negative impact on feature matching because the features of nonmatching targets will interfere with feature matching. We argue that feature redundancy exists because of the category difference between the pretraining dataset and that used for matching. For example, ILSVRC2012 has 1000 categories, while Oxford5k and Paris6k have only a very limited number of categories. We illustrate an example of this problem in Fig. 13. People are the nonmatching target in Paris6k, but it is a category of the pretraining dataset ILSVRC2012. Therefore, the pretrained model can perceive the features of the human category, and this type of feature is the so-called "redundant features." Redundant features negatively affect image matching because the dataset to be matched does not consider itself to include such features. That is, two similar images may be classified into different categories due to redundant features. Obviously, the more the category difference between the pretraining dataset and that used for matching, the higher the possibility of feature redundancy.

2) Influence of Feature Redundancy: To further quantify the influence of feature redundancy, we compare our unsupervised method against supervised methods. In addition, to show the advantages of our proposed unsupervised method in reducing the training costs, we compare two supervised methods, which are pretrained on ImageNet and trained from scratch, respectively. We report the results in Table VI, and we have several observations.

1) Feature redundancy has an impact on the model performance. The reason is that the supervised methods match the features of the target dataset more accurately, so it can better alleviate the data bias.

2) Our proposed unsupervised method saves considerable training costs at the expense of acceptable model performance (saving up to 38.5% and 46.4% of the training costs on classification and detection, respectively). It is conceivable that in the face of large-scale data, our method will have greater advantages in cost saving.

3) Unlike supervised methods, our primary goal is to pursue a trade-off between model performance and training costs. Therefore, our method has a greater practical value, especially in the face of large-scale unlabeled data.

V. CONCLUSION

This article presents a novel uncertainty calculation method to alleviate the problem of data bias in uncertainty-based AL. Using UFFM to resample the selected uncertainty data, the feature matching method we design does not introduce too many additional costs. Our AL framework based on feature matching outperforms random sampling, classic uncertainty approaches, and recent SOTA uncertainty approaches in the task of image classification and object detection. Meanwhile, unlike those AL methods that can only be used based on specific tasks or models, our framework is task-agnostic and model-agnostic and thus can be combined with almost any current uncertainty method to improve their performance. We have proved the effectiveness of our framework on image classification and object detection through experiments. In fact, our method can be applied to more complex vision tasks such as pedestrian reidentification and segmentation, and we take them as future work.

REFERENCES

[1] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2372–2379.

[2] H. H. Aghdam, A. Gonzalez-Garcia, A. López, and J. Weijer, "Active learning for deep detection neural networks," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 3671–3679.

[3] W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, "The power of ensembles for active learning in image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9368–9377.

[4] C. C. Kao, T. Y. Lee, P. Sen, and M. Y. Liu, "Localization-aware active learning for object detection," in *Proc. Asian Conf. Comput. Vis.*, 2019, pp. 506–522.

[5] L. Nie, M. Liu, and X. Song, *Multimodal Learning Toward Micro-Video Understanding* (Synthesis Lectures on Image, Video, and Multimedia Processing). Morgan & Claypool Publishers, 2019.

[6] M. Smieja, M. Wolczyk, J. Tabor, and B. C. Geiger, "SeGMA: Semi-supervised Gaussian mixture autoencoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 9, pp. 1–12, Aug. 2020.

[7] T. Ergen and S. S. Kozat, "Unsupervised anomaly detection with LSTM neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 8, pp. 3127–3141, Aug. 2010.

[8] D. Yoo and I. S. Kweon, "Learning loss for active learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 93–102.

[9] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. Conf. Res. Develop. Inf. Retr.*, 1994, pp. 3–12.

[10] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proc. Int. Conf. Mach. Learn.*, 2001, pp. 441–448.

[11] G. Wang, J.-N. Hwang, C. Rose, and F. Wallace, "Uncertainty-based active learning via sparse modeling for image classification," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 316–329, Jan. 2019.

[12] S. Chakraborty, V. Balasubramanian, Q. Sun, S. Panchanathan, and J. Ye, "Active batch selection via convex relaxations with guaranteed solution bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 1945–1958, Oct. 2015.

[13] S. Patra and L. Bruzzone, "A cluster-assumption based batch mode active learning technique," *Pattern Recognit. Lett.*, vol. 33, no. 9, pp. 1042–1048, Jul. 2012.

[14] Y. Lin, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive annotation: A deep active learning framework for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput.-Assist. Intervent.*, 2017, pp. 399–407.

[15] Z. Wang and J. Ye, "Querying discriminative and representative samples for batch mode active learning," in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 158–166.

[16] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *Proc. Int. Conf. Mach. Learn.*, 2006, pp. 1081–1088.

[17] J. Yuan, X. Hou, Y. Xiao, D. Cao, W. Guan, and L. Nie, "Multi-criteria active deep learning for image classification," *Knowl.-Based Syst.*, vol. 172, pp. 86–94, May 2019.

[18] D. Gudovskiy, A. Hodgkinson, T. Yamaguchi, and S. Tsukizawa, "Deep active learning for biased datasets via Fisher kernel self-supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9038–9046.

[19] G. Hinton and S. Roweis, "Stochastic neighbor embedding," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2002, pp. 857–864.

[20] E. Elhamifar, G. Sapiro, A. Yang, and S. S. Sasrty, "A convex optimization framework for active learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 209–216.

[21] C. Shui, F. Zhou, C. Gagné, and B. Wang, "Deep active learning: Unified and principled method for query and training," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2019, pp. 1–10.

[22] A. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "A baseline for visual instance retrieval with deep convolutional networks," *ITE Trans. Media Technol. Appl.*, vol. 4, pp. 1–8, Mar. 2014.

[23] A. Babenko and V. Lempitsky, "Aggregating deep convolutional features for image retrieval," *Comput. Sci.*, pp. 1–9, Jun. 2015.

[24] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5297–5307.

[25] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1224–1244, May 2018.

[26] X. Lu, Y. Chen, and X. Li, "Discrete deep hashing with ranking optimization for image retrieval," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2052–2063, Jun. 2020.

[27] Y. Cao, B. Liu, M. Long, and J. Wang, "HashGAN: Deep learning to hash with pair conditional Wasserstein GAN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1287–1296.

[28] S. Eghbali and L. Tahvildari, "Deep spherical quantization for image search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11682–11691.

[29] B. Klein and L. Wolf, "End-to-end supervised product quantization for image search and retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5036–5045.

[30] Y. Shen, J. Qin, J. Chen, L. Liu, F. Zhu, and Z. Shen, "Embarrassingly simple binary representation learning," in *Proc. Int. Conf. Comput. Vis. Workshop*, 2019, pp. 2883–2892.

[31] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3034–3044, Dec. 2018.

[32] E. Yang, T. Liu, C. Deng, W. Liu, and D. Tao, "DistillHash: Unsupervised deep hashing by distilling data pairs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2941–2950.

[33] Y. Shen, L. Liu, and L. Shao, "Unsupervised binary representation learning with deep variational networks," *Int. J. Comput. Vis.*, vol. 127, nos. 11–12, pp. 1614–1628, Dec. 2019.

[34] J. Xu, C. Shi, C. Qi, C. Wang, and B. Xiao, "Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, pp. 3671–3679.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[36] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[37] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9905, 2016, pp. 3671–3679.

[38] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 3212–3232, Nov. 2019.

[39] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *Int. J. Comput. Vis.*, vol. 113, no. 2, pp. 113–127, 2015.

[40] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[41] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," in *Proc. Neural Inf. Process. Syst.*, 2008, pp. 1289–1296.

[42] A. Freytag, E. Rodner, and J. Denzler, "Selecting influential examples: Active learning with expected model output changes," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 562–577.

[43] C. Käding, E. Rodner, A. Freytag, and J. Denzler, "Active and continuous exploration with deep neural networks and expected model output changes," 2016, *arXiv:1612.06129*.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[45] F. Radenovic, G. Tolias, and O. Chum, "CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 3–20.

[46] S. S. Husain and M. Bober, "Improving large-scale image retrieval through robust aggregation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1783–1796, Sep. 2017.

[47] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2016, pp. 685–701.

[48] L. Xie, L. Zheng, J. Wang, A. Yuille, and Q. Tian, "InterActive: Inter-layer activeness propagation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2016, pp. 270–279.

[49] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2016, pp. 2818–2826.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[51] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[52] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[53] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[54] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Comput. Sci. Dept., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 4, 2009, pp. 3671–3679, vol. 1.

[55] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[56] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2010.
[57] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

Wei Huang received the B.Eng. degree in computer application technology from the Henan University of Science and Technology, Luoyang, China, in 2007. He is currently pursuing the Ph.D. degree in optical engineering with the School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, China.

His current research interests include image processing, deep learning, and computer vision.

Shuzhou Sun received the B.Eng. degree in computer science and technology from Henan Agricultural University, Zhengzhou, China, in 2018. He is currently pursuing the M.Eng. degree in computer science with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China.

His current research interests include computer vision, deep learning, and image processing.

Xiao Lin received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2015.

She is currently a Full Professor with the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University (SHNU), Shanghai. She is also a Visiting Scholar with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Her current research interests include image processing, computer vision, and machine learning.

Ping Li (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2013.

He is currently an Assistant Professor with the Department of Computing and an Assistant Professor with the School of Design, The Hong Kong Polytechnic University, Hong Kong. He has published many top-tier scholarly research articles, pioneered several new research directions, and made a series of landmark contributions in his areas. He has an excellent research project reported by the *Association for Computing Machinery (ACM) TechNews*, which only reports the top breakthrough news in computer science worldwide. More importantly, however, many of his research outcomes have strong impacts on research fields, addressing societal needs and contributed tremendously to the people concerned. His current research interests include image/video stylization, colorization, artistic rendering and synthesis, realism in nonphotorealistic rendering, computational art, and creative media.

Lei Zhu received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2017.

He is currently working as an Assistant Professor with the ROAS Thrust, The Hong Kong University of Science and Technology (Guangzhou) [HKUST(GZ)], Guangzhou, China, where he is also an affiliated Assistant Professor in electrical and computer engineering (ECE). Before that, he was a Post-Doctoral Researcher with the Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge, Cambridge, U.K. His current research interests include computer graphics, computer vision, and deep learning.

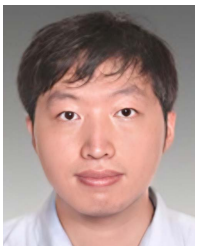
Jihong Wang received the Ph.D. degree in sport from the Shanghai University of Sport, Shanghai, China, in 2017.

He is currently an Associate Researcher with the Shanghai University of Sport. His current research interests include sports engineering, sports health promotion, and sports education and training.

C. L. Philip Chen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently a Chair Professor and the Dean of the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. Being a Program Evaluator of the Accreditation Board of Engineering and Technology Education (ABET) in the U.S., for computer engineering, electrical engineering, and software engineering programs, he successfully architects the University of Macau's Engineering and Computer Science programs receiving accreditations from Washington/Seoul Accord through Hong Kong Institute of Engineers (HKIE), which is considered as his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty of Science and Technology. His current research interests include systems, cybernetics, and computational intelligence.

Dr. Chen is a Fellow of the American Association for the Advancement of Science (AAAS), the International Association for Pattern Recognition (IAPR), CAA, and HKIE; a member of Academia Europaea (AE), European Academy of Sciences and Arts (EASA), and the International Academy of Systems and Cybernetics Science (IASCYS). He received the IEEE Norbert Wiener Award in 2018 for his contribution in systems and cybernetics, and machine learning. He is also a highly cited Researcher by Clarivate Analytics in 2018 and 2019. He was a recipient of the 2016 Outstanding Electrical and Computer Engineers Award from his alma mater, Purdue University (in 1988), after he graduated from the University of Michigan at Ann Arbor, Ann Arbor, MI, USA, in 1985. He was the IEEE Systems, Man, and Cybernetics Society President from 2012 to 2013, the Editor-in-Chief of the IEEE TRANSACTIONS ON CYBERNETICS from 2020 to 2021 and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS from 2014 to 2019, and currently, an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS. He was the Chair of Technical Committee (TC) 9.1 Economic and Business Systems of International Federation of Automatic Control from 2015 to 2017, and currently is a Vice President of Chinese Association of Automation (CAA).



Bin Sheng (Member, IEEE) received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, Hong Kong, in 2011.

He is currently a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. He is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. His current research interests include virtual reality and computer graphics.