Z-Domain Entropy Adaptable Flex for Semi-supervised Action Recognition in the Dark

Zhi Chen*  Zijun Fan*  Yongjie Li  Huaien Gao
1142154569@qq.com  1395869798@qq.com  1183474278@qq.com  1496609@qq.com

Shan Lin†
alice333.happy@163.com

Guangzhou Xi Ma Information Technology Company
101 Waihuan Xi Road, Da Xue Cheng, Guangzhou, Guangdong, China 510006

Abstract

The subtask of Human Action Recognition (AR) in the dark is gaining a lot of traction nowadays, which takes a significant place in the field of computer vision. The implementation of its application includes self-driving at night, human-pose estimation, night surveillance, etc. Currently, solutions such as DLN for AR have emerged. However, due to the poor accuracy even when leveraging on large amounts of datasets and complex architectures, the development of AR in the dark has been slow to progress. In this paper, we propose a novel and straightforward method: Z-Domain Entropy Adaptable Flex. This constructs a neural network architecture R(2+1)D, including (i) a self-attention mechanism, which combines and extracts corresponding and complementary features from the dual pathways; (ii) Zero-DCE low light image enhancement, which improves enhanced quality; and (iii) FlexMatch method, which can generate the pseudo-labels flexibly. With the help of pseudo-labels from FlexMatch, our proposed Z-DEAF method facilitates the process of gaining desired classification boundaries. This works by repeating Expanding Entropy and Shrinking Entropy. It aims to solve the problem of unclear classification boundaries between the categories. Our model obtains superior performance in experiments, and achieves state-of-the-art results on ARID.

1. Introduction

Action Recognition under dark conditions has gained widespread attention and more practical applications in real life such as self-driving in dim environment [4]. Nevertheless, there is still a lack of relevant and effective methods for Semi-Supervised Action Recognition in the dark (SS-ARID), because it requires efficiency and high accuracy. There are two main reasons for this: (i) inadequately labeled datasets in the dark which can be costly if it needs manual annotation; (ii) improper and unreasonable enhancement methods which could likely cause corruption of the datasets. Due to the reasons above, Semi-Supervised AR in the dark has gradually taken a place in solving the model degradation caused by the adverse visual condition [16, 27, 34].

As mentioned, the main question is how do we use a small amount of labeled data, plus a large amount of unlabeled data for our training. The strategy for combining two types of data becomes the key to solving the problem. Since the two types of data perform disharmony of features in domain, we can take the task as unsupervised. In unsupervised learning, the lack of category information leads to poor features, that is, poor extraction performance. Thanks to the R(2+1)D method [32] in the 3D-CNN model, we succeeded in extracting and getting abundant features to ensure the final classification.

Recently, the Semi-supervised Domain Adaptation via Minimax Entropy [8, 9, 23] (MME). It has been proposed to adversely optimize an adaptive few-shot model [2, 3]. The adaptation is achieved by alternately maximizing the conditional entropy of unlabeled target data with respect to the classifier and minimizing it with respect to the feature extractor.

Based on the MME [23] method, we proposed a novel and straightforward method, named Z-Domain Entropy Adaptable Flex for SS-ARID (Fig.1). It combines the concept of the K-means clustering architecture [21]. This model utilizes normalization to screen out the significant features from the feature extractor so that we can focus on
enhancing the suitable feature information. Meanwhile, this can further reduce divergence of the data so that the subsequent entropy process can distinguish the boundaries between different classes.

To sum up, our main work here is to (i) extend the MME [23] method from general datasets to dark datasets; and (ii) solve the problem that MME [23] does not perform well under semi-supervised conditions. On one hand, we combine the FlexMatch [39] method with the traditional MME [23] method to generate pseudo-labels in order to improve the domain adaptable performance. On the other hand, we also use the Zero-DCE enhancement [11] method to replace traditional enhancement methods [12, 30, 38, 40] to better protect dark data from too much damage. From the results of our ablation experiments, we found that Z-DEAF has proved to improve the performance which has reflected in the accuracy of the classification.

2. Related Works

R(2+1)D Based 3D ConvNet Architecture 3D ConvNet [13] is developed from 2D ConvNet by rising dimension. 2D convolution applied on an image will output an image, 2D convolution applied on multiple images also results in an image. Hence, 2D ConvNets lose temporal information of the input signal right after every convolution operation. Only 3D convolution preserves the temporal information of the input signals resulting in an output volume.

Since the task is based on video action recognition which touch on temporal dimension, we believe that 3D ConvNet is well-suited for spatiotemporal feature extraction. In 3D convolution, filters are designed in a 3D fashion, where channels and temporal information are represented in different dimensions. Compared to 2D ConvNet, 3D ConvNet has the ability to model temporal information better owing to 3D convolution and 3D pooling operations. They are performed spatio-temporally while in 2D ConvNets they are done only spatially [31].

To cut down complexity of the network and gain a better accuracy on feature extraction, a ResNet version of 3D convolution, the R(2+1)D convolutional neural network is introduced in [32]. It is a network for action recognition that employs R(2+1)D convolutions in a ResNet inspired architecture. The use of these convolutions over regular 3D Convolutions reduces computational complexity, prevents overfitting, and introduces more non-linearities that allow for a better functional relationship to be modeled [31].

Domain Adaptation Concept Deep convolutional neural networks have significantly improved image classification accuracy with the help of large quantities of labeled training data, but often generalize poorly to new domains. Recent transfer learning method [1, 17, 29], in which domain adaptation (DA) methods [10, 18, 19, 24, 33, 35, 36] improve generalization on unlabeled target data by aligning distributions. And it has been applied to various applications such as image classification [22], semantic segmentation [26], and object detection [7, 25]. However, it fails to learn discriminative class boundaries on target domains. We show that in the Semi-Supervised Domain Adaptation (SSDA) setting where a few target labels are available, such
methods often do not improve performance relative to just training on labeled source and target examples and can even make it worse [10, 19, 24].

We propose a novel approach for SSDA that overcomes the limitations of previous methods and significantly improves the accuracy of deep classifiers on novel domains with only a few labels per class.

3. Method

Our method is based on K-means clustering algorithm [21] and inspired by the essence of concept of entropy. By optimizing standard cross-entropy loss for training feature extractor and classifier for classification, reduces the distribution gap while learning classification boundaries for the task. The classifier (top layer) predicts a K-way class probability vector by computing cosine similarity between K class-specific weight vectors and the output of a feature extractor (lower layers), followed by a softmax. Each class weight vector is defined as the “base point”, that can be regarded as a representative point of that class. Through multiple trainings and iterations that repeating the process of Expanding Entropy and Shrinking Entropy yields desired classification boundaries for the task.

3.1. K-means clustering architecture

K-means algorithm is an unsupervised clustering algorithm. It is widely used due to good clustering effect and easy to implement. If the category of the data is not known before classification, like the unlabeled data in this task, we can use K-means to classify the data. The idea of k-means algorithm is intuitive. For a given sample set, it is divided into K clusters according to the distance between samples. Make the base points in the cluster as close together as possible. Meanwhile make the gap between clusters as large as possible. Through numerous iterations and training, it can effectively classify different types of data. If the gap between clusters is larger, the classification boundary between the categories will be clearer and the classification will be more accurate.

If expressed by mathematical expression, suppose the cluster \( C \) is divided into \( C = \{ C_1, C_2, \ldots, C_k \} \), the input is the sample set: \( D = \{ x_1, x_2, \ldots, x_m \} \), our goal is to minimize the squared error \( E \), which can be expressed as:

\[
E = \sum_{i=1}^{k} \sum_{x \in C_i} \| x - \mu_i \|^2 \]  

(1)

Where \( \mu_i \) is the mean vector of cluster \( C_i \), also called the mass center, can be expressed as:

\[
\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \]  

(2)

Since it is not easy to directly calculate the minimum value of the above formula, we can consider using heuristic iterative method. The heuristic method adopted by K-means is very simple, which can be vividly described by the follow group of figures.

Fig. 3 shows the initial data set, assuming \( k=2 \). In Fig. 4, we randomly chose 2 mass centers correspond to the category, the red cross and blue cross in this picture. And then respectively measure the distance between each of the points in the sample and the mass centers. Then the category of the mass center with the smallest distance to the sample is used as the category of this sample, as shown in Fig. 5. After calculating the distance between the samples and red cross and blue cross, we obtain the categories of all sample points after the first iteration. At this point, we re-
calculate 2 new mass centers of the points currently marked red and blue respectively. The positions of the new red cross and blue cross have changed. Then repeat the process that we did in Fig.5, that is, marking the category of all points as the nearest cross and finding the new mass center. Iterate until the result of this step is the same as that of the previous step. The two categories we end up with are shown in Fig.6.

### 3.2. Expanding Entropy

In semi-supervised domain adaptation, we are given source images and the corresponding labels in the source domain $D_l = \{(x_l^i, y_l^i)\}_{i=1}^{m_l}$. We also given a number of unlabeled target images in the target domain $D_u = \{(x_u^i)\}_{i=1}^{m_u}$. By training the model on $D_l$, we can evaluate the model on $D_u$.

We apply a R(2+1)D method of 3D convolutional neural network and perform $\ell_2$ normalization on the output of the network, which is the extracted feature. And then the normalized feature vector is used as an input to the classifier, which consists of weight vectors $W = [w_1, w_2, \ldots, w_K]$, where $K$ represents the number of classes.

The “base points” are parameterized by the weight vector of the last linear layer. As Fig.8 shows, the first “base point” will be near source distributions because source labels are dominant. Then, we propose to estimate the position of next “base point” by moving each $w_i$ toward target features using unlabeled data in the target domain. By operating entropy expanding with respect to the “base point” established by the previous iteration, and implementing multiple iterations until we generate a relatively stable and invariant “base point” for each class. This step we call it Expanding Entropy, and it can be graphically described as “base point” moves from near source domain to target domain (Fig.9), that is, from order to disorder, which is the essence of entropy. To achieve this, we increase the entropy measured by the similarity between $W$ and unlabeled target features. Entropy is calculated as follows,

$$H = -\mathbb{E}_{(x,y)\in D_u} \sum_{i=1}^{K} p(y = i \mid x) \log_2 p(y = i \mid x)$$  

Where $K$ is the number of classes and $p(y = i \mid x)$ represents the probability of prediction to class $i$, namely $i$th dimension of $P(x) = S \left( \frac{1}{T} W^T F(x) \right)$, where $S$ indicates a softmax function. To have higher entropy, that is, to have uniform output probability, each $w_i$ should be similar to all target features. Hence, expanding the entropy advance the model to generate the next “base point” for each class.

### 3.3. Shrinking Entropy

In order to obtain discernible and clear classification boundaries, we need to cluster unlabeled target features around the “base points”. We propose to decrease the entropy on unlabeled target examples by the feature extractor.
The proposed FlexMatch introduced a concept named Curriculum Pseudo Labeling (CPL) [6] and it has been improved that it can be easily adapted to some of the SSL algorithms and remarkably improve their performances [39]. To be more specific, the method includes two main ideas, (i) Lower the threshold value of the classes with low classification accuracy and give these classes more opportunities to be learnt for improving their value of Highest-Confident. (ii) Maintain threshold for classes that already rank high accurate to ensure high accuracy.

To this end, calculating evaluation accuracies for each class and use them to scale the threshold, as:

$$ T_t(c) = a_t(c) \cdot \tau $$

Where $T_t(c)$ is the flexible threshold for class $c$ at time step $t$ and $a_t(c)$ is the corresponding evaluation accuracy. When the threshold is high, the number of samples whose predictions fall into this class and above the threshold can reflect the learning effect of a class. Namely, the class with fewer samples having their prediction confidence reach the threshold is considered to have a greater learning difficulty or a worse learning status, formulated as:

$$ L_{u,t} = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{I}(\max(q_b) > \tau) $$

Where $\sigma_t(c)$ reflects the learning effect of class $c$ at time step $t. \ P_{m,t}(y|u_n)$ is the model’s prediction for unlabeled data $u_n$ at time step $t$, and $N$ is the total number of unlabeled data. When the unlabeled dataset is balanced, larger $\sigma_t(c)$ indicates a better estimated learning effect. By applying the following normalization to $\sigma_t(c)$ to make its range between 0 to 1, it can then be used to scale the fixed threshold $\tau$:

$$ \beta_t(c) = \frac{\sigma_t(c)}{\max_c \sigma_t} $$

$$ T_c = \beta_t(c) \cdot \tau $$

The best-learned class has its $\beta_t(c)$ equal to 1, causing its flexible threshold equal to $\tau$. As learning proceeds, the threshold of a well-learned class is raised higher to selectively pick up higher-quality samples. Eventually, when all classes have reached reliable accuracies, the thresholds will all approach $\tau$. This new threshold is used for calculating the unsupervised loss in FlexMatch, which can be formulated as:
\[ L_{u,t} = \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}(\max(q_b > 0)) \]  
\[ T_t(\text{argmax}(q_b, P_m(y | \Omega(u_b)))) \]  

Where \( q_b = P_m(y | \omega(u_b)) \). The flexible thresholds are updated at each iteration. Finally, we can formulate the loss in FlexMatch as the weighted combination of supervised and unsupervised loss:

\[ L_t = L_s + \lambda L_{u,t} \]  

Where \( L_s \) is the supervised loss on labeled data:

\[ L_s = \frac{1}{B} \sum_{b=1}^{B} H(y_b, P_m(y | \omega(u_b))) \]  

Practically, every time the prediction confidence of an unlabeled data \( u_b \) is above the fixed threshold \( \tau \), the data, and its predicted class are marked and will be used for calculating \( \beta_t(c) \) at the next time step. So far, we have successfully converted our unlabeled data into trained pseudo-labeled data and consider pseudo-labeled data as labeled data.

### 4. Experiments

#### 4.1. Experimental Details

In this section, we conduct experiments on ARID datasets which ranks the first benchmark datasets for Action Recognition in the dark [37]. To be more specific, we take the ARID datasets as unlabeled target domain, while we takes the HMDB51 [15], UCF101 [28], Kinetics-600 [5,14] and Moments in Time datasets [20] as labeled source domain. As a comparison, the former contains 3088 dark videos and the latter contains 2625 clear videos. All the videos are divided into 11 categories: drink, jump, pick, pour, push, run, sit, stand, turn, walk and wave. (The goal of dataset is to achieve satisfactory accuracy on a set of 330 dark videos.) The goal of our experiment is to improve the accuracy on a set of 330 dark videos. Meanwhile, our strategy is using the FixMatch method to generate pseudo-labels by a certain value of step, in order to change the distribution of data over a domain. As an improvement strategy, in changing the way of generating the pseudo-labels, we take FlexMatch method for better adaptable domain, and create the pseudo-labels with the same way and the same value of step. We make comparison for top-1 results of various situations through ablation experiments, and take 0.6 for threshold as the optimal scheme under FlexMarch strategy. It is well noted that the unlabeled samples from our target source do not contain our test dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Convert Ratio</th>
<th>Top-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>FixMatch</td>
<td>30%</td>
<td>41.21%</td>
</tr>
<tr>
<td>FlexMatch</td>
<td>30%</td>
<td>45.75%</td>
</tr>
<tr>
<td>FlexMatch</td>
<td>60%</td>
<td>47.27%</td>
</tr>
<tr>
<td>FixMatch</td>
<td>90%</td>
<td>46.67%</td>
</tr>
<tr>
<td>FlexMatch</td>
<td>90%</td>
<td>49.39%</td>
</tr>
</tbody>
</table>

Table 1. The Convert Ratio and Top-1 accuracy results of a few competitive models and ours

#### 4.2 Results and Comparisons

As a result of different scales between videos, we first zoom out the video in the format of (-1,256). Extract 32 frames respectively from each video. Then we clip and normalize these frames uniformly. After preprocessing, we process the video into a series of frames of size 3×32×112×112. As for the feature extractor, we adopt R(2+1)D-34 which has pretrained on IG65M to accelerate our training. In the stage of feature extraction, we would obtain the output feature of size 512×4×7×7. Noted that we classify the feature into unlabeled strong and unlabeled weak, and we enhance the unlabeled strong by the method of Zero-DCE while we do nothing to unlabeled weak.

The backbone of classification network is based on our proposed Z-DEAF method for its training, which takes used of linear network as our classifier. From Table 1, it is worth noting that our ablation experiment is to adjust the number of the pseudo-labels by the setting of the threshold value in the method of FlexMatch, referring to the accuracy of about 50% when we set the threshold to 0.9 with FixMatch.

Our model is optimized by AdamW optimizer, letting learning rate be \( 1 \times 10^{-3} \). The number of training epochs are 300 to make sure well-train as possible. To improve the model generality, a parameter \( \alpha = 1 \times 10^{-3} \) is set in weight-decay. For efficiency, the batch size of labeled and unlabeled is 64 and 75 for each considering the actual situation.

### 5. Conclude

In conclusion, we propose a new method for Semi-Supervised Action Recognition in the Dark, named Z-DEAF. It aims to solve the problem of lacking labeled datasets in dimmed environments, and improve adaptable problem of domain adaptation. In order to achieve these goals, it well takes advantage of semi-supervised learning by integrating methods such as Zero-DCE enhancement, MME method and FlexMatch method. The method will result in avoiding any unnecessary extra work, as well as gain better accuracy on the ARID dataset. Especially, we also add self-attention mechanism into the feature extractor
in order to percept the required information of features. We have conducted ablation experiments on the ARID dataset, and the experiments indicate our proposed method is feasible and powerful.

References


