

IKNN-SVM: A Hybrid Incremental Algorithm for Image Classification

Huimin Che^{1,*}, Bo Ding¹, Huaimin Wang¹, Ben Hu¹ and Huifang Che²

¹ PDL Laboratory, College of Computer, National University of Defense Technology

² College of Computing and Information System, The University of Melbourne

*Corresponding author

Abstract—Incremental learning is a machine learning paradigm that the training dataset grows dynamically at runtime and the prediction accuracy can be improved incrementally. Since real-life datasets are usually open-ended and dynamic, incremental learning is important to many image classification applications. In existing incremental algorithms, KNN (K Nearest Neighbors) has the advantage of dealing with the hugely and incrementally multiclass nature of visual object recognition. However, it provides good prediction only when the query image is similar to one of the images in the dataset, which results in low accuracy in many cases. In this paper, we propose a hybrid incremental learning algorithm named IKNN-SVM which can improve the recognition accuracy while reserving the advantages of KNN, i.e., quickly learning from the increasing number of data. To achieve this goal, our incremental algorithm combines SVM (Support Vector Machine) with high recognition accuracy and KNN together. And to promote the incremental learning efficiency, IKNN-SVM introduce kd-tree to implement fast searching. The experimental result based on the open dataset Caltech256 shows that the recognition accuracy of our method improves by 12%-26% compared with incremental KNN but with short time consumption.

Keywords—component; support vector machine; kd-tree; k nearest neighbors; image classification; incremental learning

I. INTRODUCTION

Incremental learning focuses on adjusting what has been learned by newly-arrived training examples at run-time [1]. Since the training data are usually open-ended and dynamic in real life, Incremental learning becomes increasingly important, especially in the practice of machine learning with regard to image classification [2]. For example, an image recognition cloud service should be able to utilize the ever-increasing data from Internet to improve its accuracy during its whole lifetime, and a robot should be able to learn from the human guidance as well as its own experience and incrementally extend its knowledge on object's images when it is performing a task. To cope with those practical problems, a set of incremental algorithms such as [1][3][4] for image classification have been proposed.

In an incremental learning algorithm for image classification, the existing classifier has to be frequently adjusted according to the newly-arrived training examples. Thus, an important performance index for such kind of algorithms is the adjusting time. In this dimension, the KNN(K Nearest Neighbors)-based incremental learning algorithms [5]

have a natural advantage over other algorithms, since its adjustment only needs simply adding the new examples into its knowledge base. Besides, KNN-based incremental learning algorithms has the following advantages as well: (1) Requiring little training time. (2) Effortless to deal with large and incremental data. (3) Avoiding overfitting of parameters, which is a central issue in learning based approaches. (4) The error rate of KNN classifier tends to the Bayes optimal as the samples size tends to infinity[6].

However, the KNN-based incremental algorithm have its own weak points as well. Firstly, it takes a lot of prediction time because the algorithm simply stores training and increasing data at training time and delays its learning until classification[7]. Secondly, it often has a low accuracy in image classification when the similarity of the query images and the images in the dataset is low [8]. To address those challenges, we propose a KNN-SVM hybrid incremental algorithm for image classification named IKNN-SVM in this paper. To promote the prediction accuracy, the core idea of [6] is borrowed and an improved method to adapt to our incremental learning algorithm is applied. To save prediction time, we select a kd-tree structure in KNN to store training and increasing images which helps searching close neighbors with a shorter time. A set of experiments based on the open dataset Caltech256 [9] shows that (1) IKNN-SVM performs a higher recognition accuracy than traditional KNN and close to SVM with linear kernel. (2) In the scenarios of high real-time requirements, IKNN-SVM reacts more quickly compared with some other incremental methods.

The rest of the paper is organized as follows: the related work of incremental learning for image classification is introduced in Section 2. In Section 3, the essential thought and procedures of IKNN-SVM is explained. The experiment and evaluation of IKNN-SVM based on Caltech256 is shown in Section 4 and we make a conclusion based on these results in Section 5.

II. BACKGROUND AND RELATED WORK

Image classification is used to classify a detected object into different categories. A suitable classification system and a sufficient number of training samples are important factors for a successful classification[10]. Image classification methods can be roughly divided into two categories[8]: learning-based classifiers and nonparametric classifiers. Nearest-Neighbor-Image classifier, as the most common non-parametric methods

rely on Nearest-Neighbor distance estimation classifies an image by the class of its nearest (most similar) image in the database. Learning-based classifiers are the leading image classifiers until now and SVM classifier, as one of it, is popular and widely used in practical. There are a lot of works on images datasets using SVM classifier like [11][12][13], and these works has achieved high classification accuracy. However, the most of the researches choose to combine KNN with another algorithm to neutralize KNN's disadvantages in image classification. A new method, NBNN, has been proposed in a representative work [8], combining Bayes and KNN, to employ NN distances in the space of the local image descriptors, and a multi-stage classification approach has been proposed in [13] based on sigmoid and K-nearest neighbor classifier.

The training process of large scale datasets requires big consumption of computation and time. Therefore, incremental learning methods has been widely used for image classification. Support vector based incremental learning method such as [15], which uses Karush-Kuhn-Tucker (KKT) conditions, adiabatic increments and bookkeeping method. In [16], the authors have implemented an incremental SVM based on linear kernel. In practical application, researchers are accustomed to use cluster based incremental learning. For example, [17] presents an incremental learning algorithm with local clusters for appearance-based gaze estimation. Nearest neighbors based incremental learning such as kNNJoin+ in [18] supports efficient incremental computation of KNN join results with updates on high-dimensional data. In contrast with these existing works, our algorithm implements incremental learning on kd-tree based KNN. The algorithm can bring a lower incremental time consumption in incremental phase than some other algorithms like [15] and [17]. In addition, after borrowing idea from [6], the adaptation of a hybrid incremental learning makes our algorithm in a high level of all the incremental learning algorithms.

III. IKNN-SVM CLASSIFIER

IKNN-SVM is a hybrid incremental learning method which combines incremental kd-tree based KNN and SVM. In this section, the architecture of IKNN-SVM is introduced at first, and then the procedure is explained from two phases: training phase and prediction phase.

A. Overall Architecture

The Figure 1 describes the architecture of this method, which including three parts: training initialized images, training increasing images and testing. The process of 1, 5 and 7 are parts of the feature extraction processes. Process 2 and 6 are operations on kd-tree: a tree is created in process 2 and new nodes is inserted in process 6. Process 8 is to search close neighbors of the test images, take them as an input to train SVM models in process 4 and use these models to make prediction in process 9. In the initialized data training phase, the procedure is 1-2, including extracting features and creating a kd-tree. When new images added, executing 5-6 to insert new image features into the tree. In the prediction phase, close neighbors are picked in the kd-tree of a test image to train SVM models and prediction is made on this models. Before introducing the processes of these two phases, we briefly describe the concept of kd-tree.

kd-tree is a space-partitioning data structure for organizing points in a k-dimensional space, in a kd-tree, every node is a k-dimensional point. In this paper, a balanced kd-tree is established, which means that the heights of the two child subtrees of any nodes differ by at most one because the more balance the tree is, the distribution of nodes would be more uniform and the search time would be shorter[19]. The operation on creation takes $O(M\log N)$ time where n is the number of nodes in the tree prior to the operation, and the basic operations (e.g., add, delete) for one node all cost $O(\log N)$ time.

B. Training Phase

The training input for IKNN-SVM is the images from dataset, so the processing procedures of training phase is divided into two parts: initialized training phase and increment phase.

1) *Initialized training phase.* Initializing training phase needs two steps: feature extraction and creating a kd-tree. These two steps mapping to the process 1 and 2 in Figure 1. After the feature extraction, all the extracted features are regarded as nodes to create a kd-tree. This tree is built as an initialized knowledge base and would be used in the later phases.

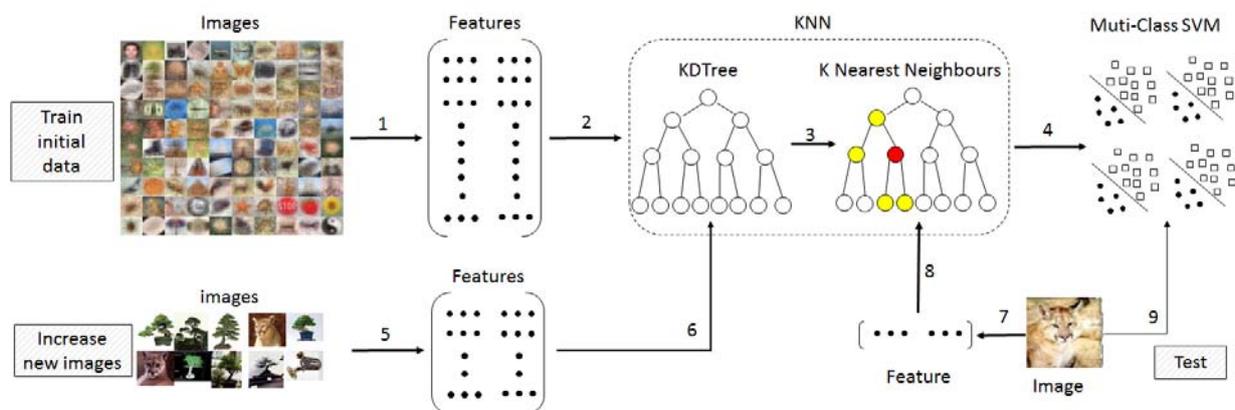


FIGURE 1. ARCHITECTURE OF IKNN-SVM

2) *Increment phase.* When new images appears and need to be added into the existing knowledge base, process 5 and 6 shown in Figure 2 would be executed. After feature extraction, new samples would be inserted into the existing kd-tree as nodes in a balanced way. The basic operations shows below: Firstly, traversing the kd-tree to find a suitable location to save the new node. Once the location is confirmed, add the new node to this location. Inserting points in this manner may cause unbalance of the tree and a re-balanced approach for close neighbors searching is needed. After this phase, the kd-tree is growing to be denser and the knowledge base is extended.

C. Prediction Phase

Prediction phase refers to the close neighbors searching and SVM training and prediction. When an image need to be recognized, the feature of it is inserted into kd-tree first and then be set as a central point to search close neighbors. As shown in Figure 2 (a), the method searches close neighbors around it using Euclidean distance in formula (1) in which x, y are two points and n is the dimension of each point.

$$d(x, y) := \sqrt{\sum_{i=0}^n (x_i - y_i)^2} \quad (1)$$

If the close neighbors of a chosen point have the same label as the point, like C, the point would be labeled and exit; else, take the close neighbors as inputs to train multi-class SVM models. This process is shown in Figure 2 (b). At last, make prediction and access the label for the test image exploiting SVM models. There are two important processes for prediction: searching close neighbors in kd-tree and training and prediction on SVM.

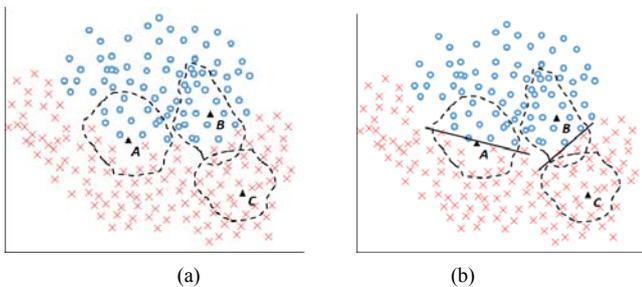


FIGURE II. MAIN IDEA OF COMBINING KNN AND SVM.

1) *Searching close neighbors in kd-tree.* Firstly, the approach would search the nearest neighbor and take parent node of the test sample as the nearest neighbor called "current best"; then find the parent or another child as "current node"; if the "current node" is closer than the current best, then it becomes the "current best"; repeating the above steps constantly until the "current best" is the best or it has traced back to the root node. We can access K nearest neighbors in the same way.

2) *Training and prediction on SVM.* In the prediction phases, we choose to combine KNN and SVM to promote the recognition accuracy for image classification. After searching the close neighbors, these points are collected and then trained by SVM. In this work, more than two classes are trained in object categories, and if using one-versus-one SVM method and the test sample appears around the point B in Figure 2, the unbalance of the number of different samples may bring deviation in separating hyperplane. Therefore, we choose one-versus-all multi-class SVM method to train SVM models in the process 4 of Figure 1 and make a prediction using these models in the step 9 in Figure 1.

IV. EXPERIMENT AND EVALUATION

In this section we do experiment on IKNN-SVM, and compare its performance with incremental linear SVM[16] and traditional incremental KNN. The experiment dataset is Caltech-256, including 30608 pictures of objects belonging to 256 categories with about 80 to 827 images per category and the size of each image is roughly 300 x 200 pixels.

In the experiment, the amount of all images in the dataset is 30608 and 15304 images, 50% from each classes of Caltech 256 dataset, are randomly chosen as initialized training data. The training data is increased by 10% from each classes (about 3509 images) every time. In each step, the rest dataset including 3060 images will be predicted wherever in the process of initialized training or increment phases.

A. Training Phase Evaluation

In training phase, the time saving purpose can be achieved from two aspects: feature extraction and kd-tree operations.

1) *Feature extraction.* Feature extraction is an integral part of training models and the inputted original images are pre-handled by the deep learning architecture caffe[20] with AlexNet using convolutional neural network[21]. In training phase, the input is a batch of images, but it is a waste of time and resources for large-scale data to extract all the features of training images in every phases. Therefore, we choose to preserve features into physical storage as long as they are extracted. In our experiment, we extract full connected layers in Alex net using caffe. It costs about 59 minutes for feature extraction to train the initial 15304 images and which means in average 0.23s for each image. In incremental learning phase, feature extraction only costs 13 minutes for 3509 images because we can access the features of initialized images from last step directly in physical storage.

TABLE I. COMPARISON OF INSERTING NODE INTO KD-TREE AND RETRAINING

Stages	Incremental kd-tree	Reconstructing kd-tree
Initial Data	2.83 ± 0.04s	2.83 ± 0.04s
Increment 1	3.20 ± 0.05s	3.32 ± 0.04s
Increment 2	3.28 ± 0.03s	3.71 ± 0.03s
Increment 3	3.35 ± 0.03s	5.5 ± 0.05s
Increment 4	4.89 ± 0.04s	6.12 ± 0.04s

2) *kd-tree creating and inserting*. Creating and inserting operation for kd-tree is the process of preserving features into physical storage. Feature extraction phase produces a 4096 dimensional vectors for each images. As shown in Table 1, creating balanced kd-tree for basic 15304 image features costs about 2.83s. In each increment phase, 3509 test images need to be incremental learned. IKNN-SVM only needs to insert new images into the previous kd-tree, while the way of reconstruction need to rebuild the whole tree. The comparison result shown in Table 1 illustrates that using the method of incremental kd-tree can save time comparing with reconstructing the whole tree.

B. Prediction Phase Evaluation

As shown in Table 2, 3060 images are tested in every stage using IKNN-SVM. The classification performance of our experiment is separated into five groups. The first group shows the recognition accuracy and time consumption which includes feature extraction, close neighbors searching, SVM models training and predicting for each image in the classification group. Later groups demonstrate the recognition accuracy and time consumption on prediction phase after new images are learned incrementally.

TABLE II. RECOGNIZE ACCURACY AND TIME CONSUMPTION IN PREDICTION PHASE

K	Initialized Data(50%)		Increment1 (10%)		Increment2 (20%)		Increment3 (30%)		Increment4 (40%)	
	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
50	58.78%	0.35s	59.76%	0.39s	60.35%	0.40s	61.32%	0.41s	61.83%	0.42s
100	60.58%	0.40s	62.00%	0.44s	62.69%	0.45s	63.34%	0.46s	64.00%	0.47s
150	62.20%	0.42s	63.31%	0.50s	63.77%	0.52s	64.60%	0.53s	65.83%	0.54s
200	62.97%	0.55s	64.54%	0.58s	65.08%	0.60s	65.59%	0.60s	66.22%	0.62s
250	63.48%	0.64s	65.22%	0.67s	65.85%	0.68s	66.76%	0.69s	67.45%	0.72s
300	63.88%	0.74s	65.51%	0.76s	66.39%	0.78s	67.16%	0.78s	67.87%	0.79s
350	64.45%	0.85s	66.28%	0.87s	66.93%	0.89s	67.47%	0.88s	67.99%	0.88s
400	65.08%	0.97s	66.65%	0.98s	67.10%	1.00s	67.62%	0.99s	68.24%	0.99s
450	65.42%	1.08s	66.76%	1.09s	67.08%	1.12s	68.13%	1.10s	68.27%	1.10s
500	65.62%	1.21s	66.96%	1.20s	67.38%	1.24s	68.04%	1.21s	68.47%	1.21s

C. Compared with Other Incremental Method

The differences between incremental linear SVM and traditional incremental KNN methods should be considered when evaluating the performance of IKNN-SVM. The comparison of the recognition accuracy is shown in Table 3. The value of K is set as 10 in the experiment to achieve the optimum solution in naive KNN. The comparison of traditional incremental KNN and IKNN-SVM are shown in Table 3, and we can see that even K is fixed to 10, the recognition accuracy of IKNN-SVM is still higher than naive KNN. The recognition accuracy of incremental SVM is higher than IKNN-SVM, but they are highly similar when K=500.

Figure 4 demonstrates the comparison of incremental waiting time for IKNN-SVM with K values 500 and incremental linear SVM. In our experiment, we define the incremental waiting time as a time quantum from the beginning of incremental learning to the time when a test image is predicted. The blue line in the figure represents the tendency of incremental linear SVM and the orange line

Figure 3 highlights the tendency of recognition accuracy and predicts time consumption after periodical incremental learning in different K values. From this tendency together with the Table 2, we can get the conclusions: (1) As the value of K enlarging, the recognition accuracy is promoting and more prediction time is needed. (2) With the kd-tree becomes denser after incremental learning, the recognition accuracy is rising steadily. (3) The magnitude of change in time consumption is very small as the kd-tree grows.

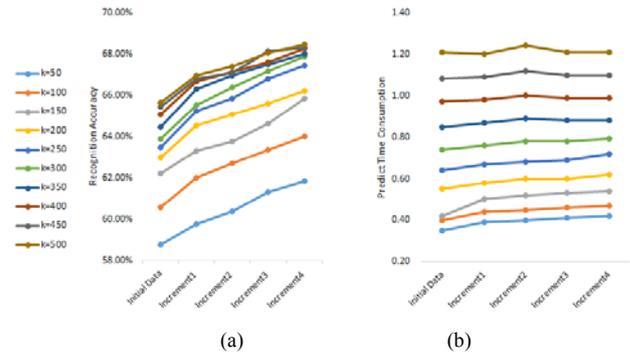


FIGURE III. TENDENCY OF RECOGNITION ACCURACY AND TIME CONSUMPTION IN PREDICTION PHASE AFTER PERIODICAL INCREMENTAL LEARNING IN DIFFERENT K VALUES.

represents the change of incremental waiting time for IKNN-SVM with K values of 500. The figure shows that with the increasing number of images, the incremental waiting time of incremental SVM becomes larger, while the time consumption of IKNN-SVM is rising slowly and stuck at 3.3s.

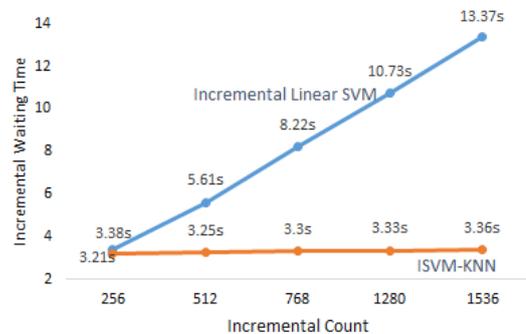


FIGURE IV. COMPARISON OF INCREMENTAL WAITING TIME FOR IKNN-SVM WITH K VALUES 500 AND INCREMENTAL LINEAR SVM.

TABLE III. COMPARISON OF IKNN-SVM AND NAIVE INCREMENTAL KNN IN RECOGNITION ACCURACY

Methods	Initialized Data	Increment 1	Increment 2	Increment 3	Increment 4
KNN(K=10)	40.62%	41.25%	42.84%	42.53%	43.12%
IKNN-SVM (K=10)	53.02%	53.53%	54.70%	55.47%	56.87%
IKNN-SVM (K=500)	65.62%	66.96%	67.38%	68.04%	68.47%
Incremental SVM	68.96%	70.15%	69.56%	69.84%	70.61%

V. CONCLUSION

In this paper, we give a brief introduction on kd-tree and introduce the procedures of the hybrid incremental learning algorithm IKNN-SVM. In this algorithm, KNN is used to search the most similar neighbors and the SVM is used to classify samples and make predictions. The use of kd-tree promotes the efficiency in incremental phase by inserting new images into the previous kd-tree quickly and in recognition phase by searching quickly. The experimental results demonstrate that the recognition accuracy of IKNN-SVM is higher than KNN but lower than linear SVM. However, we gain significant benefits in incremental phase by less time consumption. Therefore, IKNN-SVM algorithm is most suitable for real-life scenarios to deal with fast incremental learning problems. Our future work includes optimizing the algorithm to improve its performance for practical use.

ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China (nos. 9118008 and 61202117), the special program for the applied basic research of the National University of Defense Technology (no. ZDYJCYJ20140601), and the Jiangsu Future Networks Innovation Institute Prospective Research Project on Future Networks (no. BY2013095-2-08).

REFERENCES

- [1] D. A. Ross, J. Lim and R. S. Lin, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, pp.125-141, 2008.
- [2] M. Ristin, M. Guillaumin and J. Gall, "Incremental learning of NCM forests for large-scale image classification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3654-3661, 2014.
- [3] R. Polikar, L. Upda and S. S. Upda, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. C31, pp. 497-508, 2001,.
- [4] L. Fei-Fei, R. Fergus, P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, pp. 59-70, 2007.
- [5] G. Zhao, K. Xuan, D. Taniar, et al. "Incremental k-nearest-neighbor search on road networks," *Journal of Interconnection Networks*, vol. 9: pp. 455-470, 2008.
- [6] H. Zhang, A. C. Berg and M. Maire, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, vol. 2, pp. 2126-2136, 2006.
- [7] D. W. Aha. "Lazy learning," Kluwer academic publishers, 1997.
- [8] O. Boiman, E. Shechtman, M. Irani. "In defense of nearest-neighbor based image classification" *Computer Vision and Pattern Recognition*, 2008. *CVPR 2008*. IEEE Conference on. IEEE, pp. 1-8, 2008
- [9] G. Griffin, A. Holub, P. Perona. "Caltech-256 object category dataset," 2007.
- [10] D. Lu, Q. Weng. "A survey of image classification methods and techniques for improving classification performance,". *International journal of Remote sensing*, vol. 28, pp. 823-870, 2007.
- [11] E. Pasolli, F. Melgani and D. Tuia, "SVM active learning approach for image classification using spatial information," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 2217-2233, 2014.
- [12] M. Marszałek, C. Schmid, H. Harzallah and J. van de Weijer. Learning object representations for visual object class recognition. In *Visual Recognition Challenge*, 2007.
- [13] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.
- [14] Y. Sugano, Y. Matsushita, Y. Sato, et al. "An incremental learning method for unconstrained gaze estimation," *European Conference on Computer Vision*. Springer Berlin Heidelberg, pp. 656-667, 2008.
- [15] C P. Diehl, G. Cauwenberghs. "SVM incremental learning, adaptation and optimization," *Neural Networks*, 2003. *Proceedings of the International Joint Conference on. IEEE*, pp. 2685-2690, 2003.
- [16] C. H. Tsai, C. Y. Lin, C. J. Lin. "Incremental and decremental training for linear classification," *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 343-352, 2014.
- [17] Y. Sugano, Y. Matsushita, Y. Sato, et al. "An incremental learning method for unconstrained gaze estimation," *European Conference on Computer Vision*. Springer Berlin Heidelberg, pp. 656-667, 2008.
- [18] C. Yu, R. Zhang, Y. Huang, et al. "High-dimensional knn joins with incremental updates,". *Geoinformatica*, Vol. 14, pp. 55-82, 2010.
- [19] Bentley J L. "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, Vol.18, pp. 509-517, 1975:.
- [20] J. Redmon, S. Divvala, R. Girshick, et al. "You only look once: Unified, real-time object detection,". *arXiv preprint arXiv:1506.02640*, 2015.
- [21] A. Krizhevsky, I. Sutskever, G. E. Hinton. "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*. pp. 1097-1105, 2012.