

IMPROVE NANO-CUBE DETECTION PERFORMANCE USING A
METHOD OF SEPARATE TRAINING OF SAMPLE SUBSETS

by

Sai Krishnanand Nagavelli

Submitted in Partial Fulfillment of the Requirements

for the Degree of Master

of

Computing and Information Systems

YOUNGSTOWN STATE UNIVERSITY

December, 2016

IMPROVE NANO-CUBE DETECTION PERFORMANCE USING A
METHOD OF SEPARATE TRAINING OF SAMPLE SUBSETS

Sai Krishnanand Nagavelli

I hereby release this thesis to the public. I understand that this thesis will be made available from the OhioLINK ETD Center and the Maag Library Circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

Sai Krishnanand Nagavelli, Student

Date

Approvals:

Dr. Yong Zhang, Co-Advisor

Date

Dr. Feng George Yu, Co-Advisor

Date

Dr. John Sullins, Committee Member

Date

Dr. Salvatore Sanders, Dean of Graduate Studies

Date

©

Sai Krishnanand Nagavelli

2016

DEDICATION

I dedicate this thesis to my parents for all their love, support, sacrifices and putting me through the best education possible. I appreciate my friends for their continuous encouragement which made me keep going all the way.

ABSTRACT

The transmission electron microscopy (TEM) is an imaging technique whereby beams of electrons are driven through a thin specimen so that its structure on the nanoscale can be captured. Due to the unique capabilities of TEM, it has been applied to many fields such as the studies of biological tissues, virology, analyzing reactive chemical compounds, monitoring crystal growth and examining 3D printing quality, etc. As a result, a large quantity of TEM data has been produced that are far beyond human processing capabilities. This thesis applies an ensemble learning method based on the AdaBoost to automatically detect cube-shaped nanoparticles in a single TEM image. The specific aim is to improve the detection performance by training classifiers with different subsets of the original positive samples. The subsets are organized according to the degree of particle overlapping so that the classifier can pick up the Haar-like features that are more sensitive to overlapping. Promising results have been observed in the preliminary tests with a 7.89% increase of the overall detection rate.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Yong Zhang for the continuous support of my thesis, for his patience, motivation, and immense knowledge. His guidance helped me in all the experiment tasks as well as writing of this thesis. I could not have imagined having a better advisor and mentor for my thesis study.

Besides my advisor, I would like to thank the rest of my thesis committee Dr. John Sullins, and Dr. Feng George Yu, for their encouragement throughout my academic course work at YSU and sparing their valuable time to serve in the committee.

TABLE OF CONTENTS

ABSTRACT	IV
ACKNOWLEDGEMENTS	V
LIST OF FIGURES	VII
LIST OF TABLES	VIII
1. INTRODUCTION	1
1.1 ENSEMBLE LEARNING METHOD	2
1.2 MAJOR ISSUES AND CONTRIBUTIONS	2
2. METHODS	4
2.1 CASCADE ADABOOST	4
2.2 RAPID HAAR FEATURE COMPUTATION	6
3. SAMPLE PREPARATION	9
3.1 ORIGINAL POSITIVE SAMPLES	9
3.2 GROUPING OF POSITIVE SAMPLES	10
3.3 NEGATIVE SAMPLES.....	12
4. EXPERIMENTS	13
4.1 TRAINING WITH SUBSETS FROM CUBE_A.....	16
4.2 TRAINING WITH COMBINED SUBSETS OF CUBE_A & CUBE_E.....	16
5. ANALYSIS.....	17
5.1 DETECTION USING THE SUBSETS OF CUBE_A	17
5.2 DETECTION USING THE SUBSETS OF CUBE_A & CUBE_E	19
5.3 COMPARATIVE ANALYSIS.....	21
6. CONCLUSIONS	27
7. REFERENCES	28

LIST OF FIGURES

FIGURE 1. ILLUSTRATION OF CASCADE ADABOOST METHOD	5
FIGURE 2. ILLUSTRATION OF INTEGRAL IMAGE CALCULATION.....	7
FIGURE 3. ILLUSTRATION OF COMPUTING AN INTEGRAL IMAGE	8
FIGURE 4. ILLUSTRATION OF COMPUTING FEATURE VALUE USING AN INTEGRAL IMAGE.....	8
FIGURE 5. THE IMAGES FROM WHICH THE POSITIVE SAMPLES WERE SELECTED	10
FIGURE 6. FIVE SUBSETS OF POSITIVE SAMPLES ACCORDING TO THEIR OVERLAPPED RATIOS.....	11
FIGURE 7. NEGATIVE SAMPLES USED IN THE TRAINING	12
FIGURE 8. DETECTION RESULTS USING THE 5 SUBSETS OF CUBE_A FOR TRAINING	18
FIGURE 9. RESULTS USING THE COMBINED 5 SUBSETS OF CUBE_A AND CUBE_E FOR TRAINING	20
FIGURE 10. LABELLED PARTICLES IN THE IMAGE OF CUBE_B	22

LIST OF TABLES

TABLE 1. IMAGE SETS AND OBJECTS	9
TABLE 2. NUMBER OF POSITIVE SAMPLES IN EACH SUBSET.....	11
TABLE 3. RESULTS USING SUBSETS OF CUBE_A FOR TRAINING.....	17
TABLE 4. RESULTS OF USING SUBSETS OF CUBE_A AND CUBE_E.....	19
TABLE 5. SUPER TRAINING SUBSETS (CUBE_A + CUBE_E).....	21
TABLE 6. RESULTS ON SUBSETS OF 0% OVERLAP RATIO IN CUBE_B.....	23
TABLE 7. RESULTS ON SUBSETS OF 0-25% OVERLAP RATIO IN CUBE_B.....	23
TABLE 8. RESULTS ON SUBSETS OF 25-50% OVERLAP RATIO IN CUBE_B.....	24
TABLE 9. RESULTS ON SUBSETS OF 50-75% OVERLAP RATIO IN CUBE_B.....	25
TABLE 10. RESULTS ON SUBSETS OF 100% OVERLAP RATIO IN CUBE_B.....	25
TABLE 11. IMPROVEMENT USING SEPARATE TRAINING OF SUBSETS.....	26
TABLE 12. OVERALL RESULTS.....	26

1. INTRODUCTION

Unlike other types of microscopes, the transmission electron microscope uses beams of electrons instead of normal light sources to form an image of the object being examined. The advantage of using electrons is that they have very small wavelengths and thus can capture images with a resolution far exceeding optical microscopes to reach the nanoscales[1]. The setup of a transmission electron microscope is composed of three basic components: (a) An electron gun that shoots out beams of electrons passing through the specimen; (b) An image production system that receives and manages electrons; (c) An image recording component that maneuvers and coordinates the object and camera. In the transmission electron microscopy, electromagnetic lens are used and the output is displayed on a monitor screen. In order to make an image of a specimen, a TEM blasts the specimen with electrons in a vacuumed environment to avoid the potential side effects of resistance or interference. The recording component has a screen that can be used to adjust the object and a camera to take a digital image. TEMs are capable of capturing material structures in the atom level with a very high resolution. Additionally, TEMs that are optimized for a specific environment are capable of even characterizing the morphology, composition, crystallography and chemical properties of a given sample.

TEMs have been successfully used in a wide range of applications, including the study of biological tissues, drug analysis and inspection of semiconductor circuitries. This thesis focuses on the detection of nanoparticles in transmission electron microscopy images that are useful for the study of chemical absorption and in situ reactions dynamics.

1.1 ENSEMBLE LEARNING METHOD

Since the size of a raw TEM image is very large (in the range of 1-20 MB), the traditional method of manually analyzing a single image is a time consuming process. When dealing with a large number of TEM images, it is simply impractical to rely on a human specialist. Furthermore, there is a real challenge that a person cannot provide a consistent analysis of fine image details of many samples such as their shapes, sizes and surface areas. This is a task that requires more sophisticated object detection methods, especially the accurate detecting and counting of individual particles. This task also poses a challenge to the regular image processing methods because of the presence of image noises and the overlapping effect of semi-transparent nanoparticles[2]. To tackle this challenging problem, a variant of boosting method known as the Cascade AdaBoost algorithm is used. This method has several advantages: (1) It can transform a large number of relatively weak classifiers into a very strong classifier via an adaptive boosting strategy; (2) By sequentially evaluating and adding new classifiers, it acts like a feature selection filter that effectively pick a small set of rectangle features; (3) As a linear combination of selected tree-stumps, the final detector can process a large number images efficiently.

1.2 MAJOR ISSUES AND CONTRIBUTIONS

It was a difficult task to detect severely overlapped cubes, especially those with more than 50% overlapped areas and those on the image boundary with missing parts. In addition, it was not clear whether the types of positive and negative sample would have any effect on the training and detection results.

This thesis attempts to tackle the first problem using a method that trains a strong classifiers using the subsets of original positive samples. Five subsets of positive samples were selected based the degrees of overlapping and were then used in the training and testing phases. The main contributions of the thesis are:

- It extends the detection range from simple and clean cubes to more severely overlapped ones by training a classifier with different subsets of samples according to their respective overlapping percentages.
- Comparative studies were conducted using a super training subset in each case to evaluate the improvement of detection accuracy. It was found that the performance of a classifier depends not only on samples and feature sizes but also on the overlapping percentages.
- It was observed that the method of separate training with sample subsets not only increased the overall detection rate but also minimized the false alarm rate.

2. METHODS

2.1 CASCADE ADABOOST

AdaBoost is a popular ensemble learning method that is designed to create a strong classifier by combining a number of weak classifiers with just above average performance or better than random guessing. AdaBoost is considered a meta-algorithm and it can be used to work with various weak classifiers, including the decision tree stumps[1], [2]. Each single classifier carries out a basic task based on the single dimension of a given input vector. It has an advantage that a high detection rate can be achieved by adding a large number of weak classifiers. Another advantage is that it requires much less parameter tweaking as compared to other algorithms. However, the method might take a much longer training time and detection time, especially when the number of image features is larger.

To handle the cases of real-time object detection in images and videos, the Cascade AdaBoost technique was developed [3] (see Figure 1). In the Cascade AdaBoost method, the F entities stand for the number of negative samples that are rejected which are collected in a set of sub-windows. The T entities stand for the number of positive samples that are accepted. The input training set represented by all of sub-windows consists of both positive and negative samples. The goal is to determine which samples in the training set are positive and which are negative. In order to achieve this goal, the Cascade AdaBoost method removes the majority of negative samples in early stages[4]. This means that the samples provided by the training set will continuously decrease in size as it passes through the subsequent stages of the algorithm.

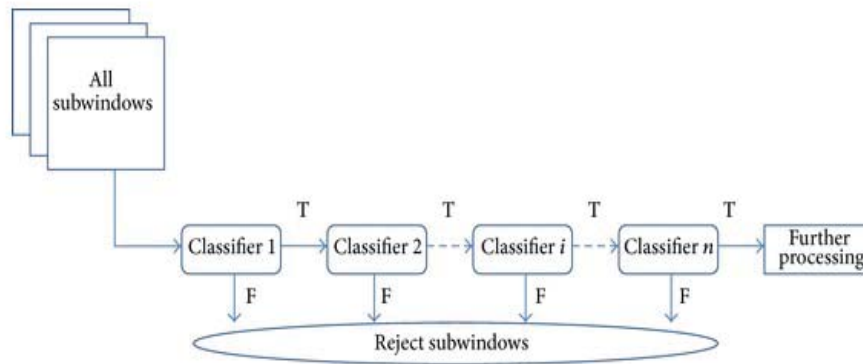


Figure 1: Illustration of the Cascade AdaBoost method

A few key parameters must be setup before commencing of the training process[5], [6]. The *Maxi* and *Mini* represent the maximum allowable false positive and minimum acceptable detection rates. Fal_{target} is set as the target for the false alarm rate. The samples in the training set can be categorized into two types: *Posi* for positive samples and *Nega* for negative samples[7], [8]. The training is carried out in two loops. During the inner loop, a check is performed on the target value once a new classifier is selected. The training process is carried out until the false alarm target is obtained. If the maximum false positive rate drops below the false alarm target, the training process terminates[9].

Alternatively, the negative samples are reset and the number of detections of the false positives are set to null. The outer loop is to redirect back to the training until the maximum false positive rate drops below the false positive alarm target[10]. The procedure of the Cascade AdaBoost process can be described as adaptively selecting the values of various parameters, including the initial value assignments and the specific steps followed. Several key parameters are described below.

- $Maxi$: maximum allowable false alarm rate
- $Mini$: minimum acceptable detection rate
- Fal_{target} : false positive alarm target
- $Posi$: collection of positive samples
- $Nega$: collection of negative samples

The initial values of the parameters are:

- $Maxi_0=1$
- $Mini_0=1$
- $s=0$

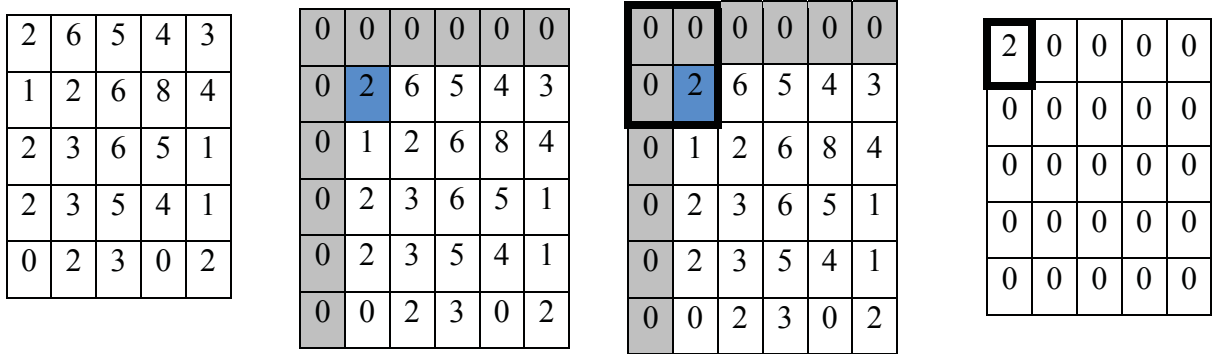
The steps for the procedure are given below:

- While $Maxi_s > Fal_{target}$
 - $s = s + 1$; $Nega_s = 0$; $Maxi_s = Maxi_{s-1}$
 - while $Maxi_s > f \times Maxi_{s-1}$
 - ❖ $Nega_s = Nega_{s+1}$.
 - ❖ utilize $Posi$ and $Nega$ to train a classifier with $Nega_s$ traits
 - ❖ check the current classifier on validation set so as to determine $Maxi$ and $Mini_s$.
 - ❖ determine threshold for the s th classifier so that the current cascade classifier has a detection rate $> (Mini \times Mini_{s-1})$
 - $Nega$ is null.
- If $Maxi_s > Fal_{target}$, then analyze the current classifier on the collection of negative samples and place false detection in the collection $Nega$

2.2 RAPID HAAR FEATURE COMPUTATION

An integral image is a transitional form of the initial image. The fast calculation of Haar features of a rectangle can be done with an integral image[11]. On a given axis, an integral image at a place (c, d) describes the aggregate number of pixels with respect to the left and above the origin vertical axis and horizontal axis respectively. A mathematical representation of an integral image is illustrated in Figure 2.

$$ii(c, d) = \sum_{e^1 \leq e, d^1 \leq d} i(c^1, d^1) \quad (1)$$

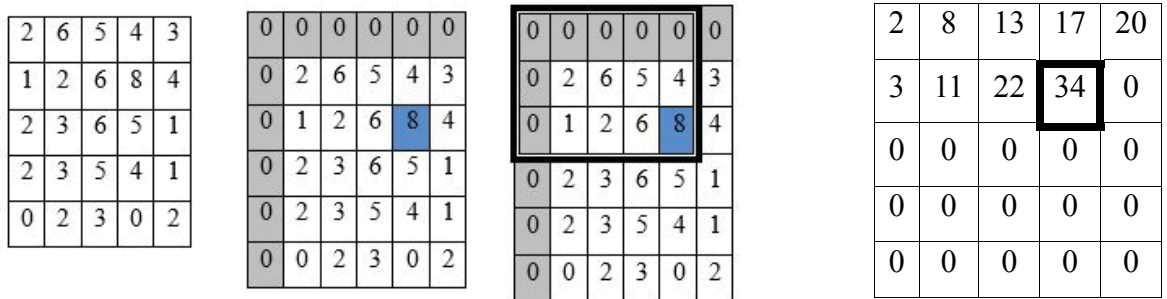


Original image

a

b

Integral image



Original image

a

b

Integral image

Figure 2: Illustration of integral image calculation

Feature values of a rectangle can be calculated in real time provided an integral image is given. Figure 2 represents the transformation of an original image which consists of a 5x5 rectangle to its integral image. The original image undergoes a transformation where a row/column of 0s are added to the top/left of the image turning it into a 6x6 one. The process takes many rectangles of varying sizes and calculates their sum aggregates and assigning them to the bottom rightmost cells. The result is an integral image.

Original image					Integral image				
2	6	5	4	3	2	8	13	17	20
1	2	6	8	4	3	11	22	34	41
2	3	6	5	1	5	16	33	50	58
2	3	5	4	1	7	21	43	64	73
0	2	3	0	2	7	23	48	69	80

Figure 3: Illustration of computing an integral image.

Position 1		Position 2	
Position 3		Position 4	

Figure 4: Illustration of computing feature value using an integral image.

In figure 3, the integral sum within the highlighted rectangle area is 30, which can be calculated on a condition that the integral image values at the four corners are known.

The formula to calculate the feature value of the highlighted rectangle in Figure 4 is:

$$a (\text{position 4}) + b (\text{position 1}) - c (\text{position 2}) - d (\text{position 3}) \quad (4)$$

For the example in Figure 3, it is $50 + 2 - 17 - 5 = 30$, the same as the aggregate sum in the original rectangle.

3. SAMPLE PREPARATION

3.1 ORIGINAL POSITIVE SAMPLES

Positive training samples were cropped manually from TEM images of cube-shaped particles and the number of objects (cubes) used in this study are listed in Table 1. Multiple labelers were involved in the cropping process of each image and hence the effect of labeling errors may exist. But the effect of mislabeled data set is beyond the scope of this thesis. The focus of this study is on the analysis of separate training of sample subsets. The original images have different resolutions, but the cropped positive samples were all transformed into a standard size (24 x 24) before they were used in the training phase of Cascade AdaBoost and therefore the impact of image resolution was reduced to the minimal level. The original image set with some partially labeled cubes are shown in Figure 5.

TABLE 1: IMAGE SET AND OBJECTS

Images	Usage	Total number of objects Cropped by multiple labelers.
cube_a	Training	139
cube_e	Training	112
cube_b	Testing	76

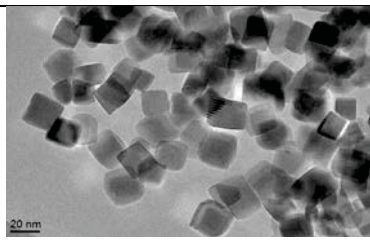
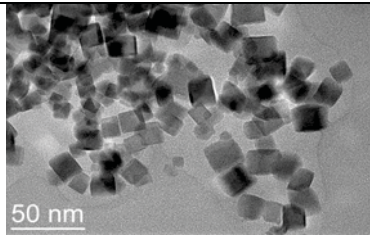
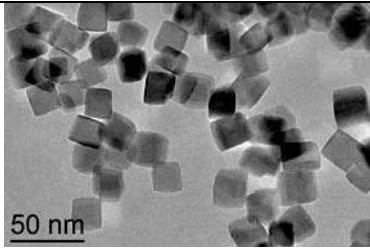
		<p>Cube a (training)</p>
		<p>Cube e (training)</p>
		<p>Cube b (testing)</p>

Figure 5: The images from which the positive samples were selected

3.2 GROUPING OF POSITIVE SAMPLES

One challenge of detecting cubes in TEM images is the occlusion and cubes are semi-transparent (a cube is not completely blocked, instead the electrons passed through the overlapped cubes and created areas of varying intensities). As a result, the traditional methods of handling object occlusion may not be effective. The semi-transparent nature of occluded cubes offers a unique opportunity because it provides more information about the degree of overlapping, albeit the degree may vary depending upon the number of cubes, rotation angles and particle thickness. To utilize this property, a separate training method is proposed. The first step is to put all positive samples into five different subsets ranging from 0% to 100% overlapping ratios (see Figure 6 and Table 2).

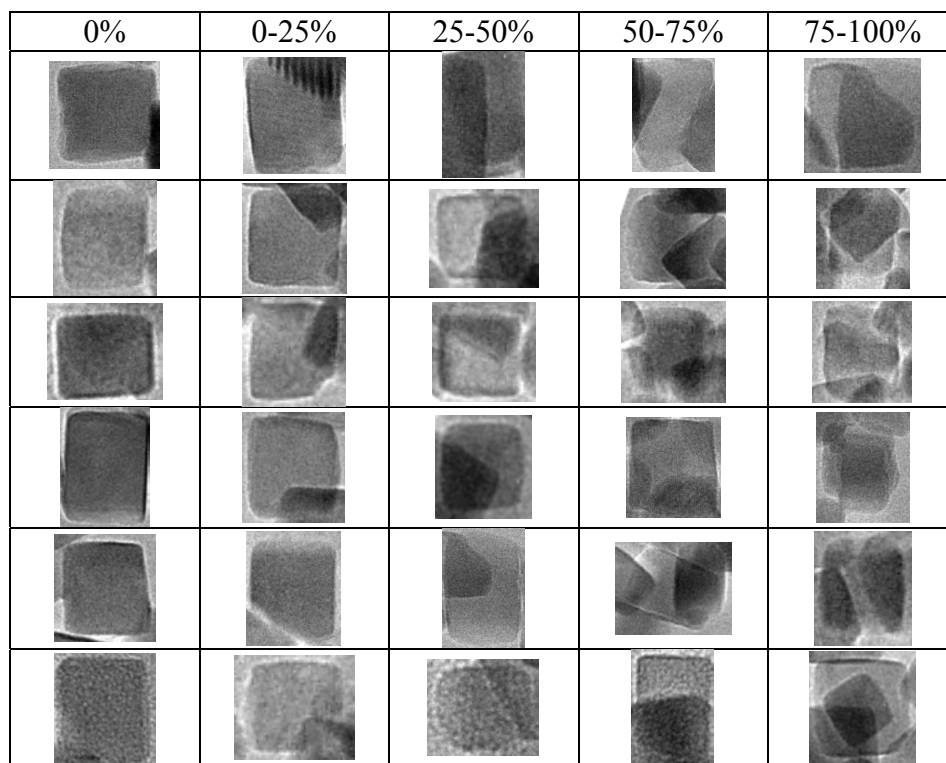


Figure 6: Five subsets of positive samples according to their overlapped ratios.

TABLE 2: NUMBER OF POSITIVE SAMPLES IN EACH SUBSET

Image	0%	0-25%	25-50%	50-75%	75-100%
cube_a	24	25	14	28	48
cube_e	44	25	8	9	25
cube_b	6	13	11	20	26

3.3 NEGATIVE SAMPLES

A large archive of negative samples were collected by cropping a number of arbitrary pictures of different scenes (Figure 7). All negative samples are of the same size that is marginally bigger than that of positive samples (but were later transformed into the same sizes as that positive sample during the actual training process). Another option is to use the actual background samples in the TEM images. The main consideration of choosing random negative samples is to increase the distribution diversity of negative samples which might increase the robustness of the final classifier and detection performance.

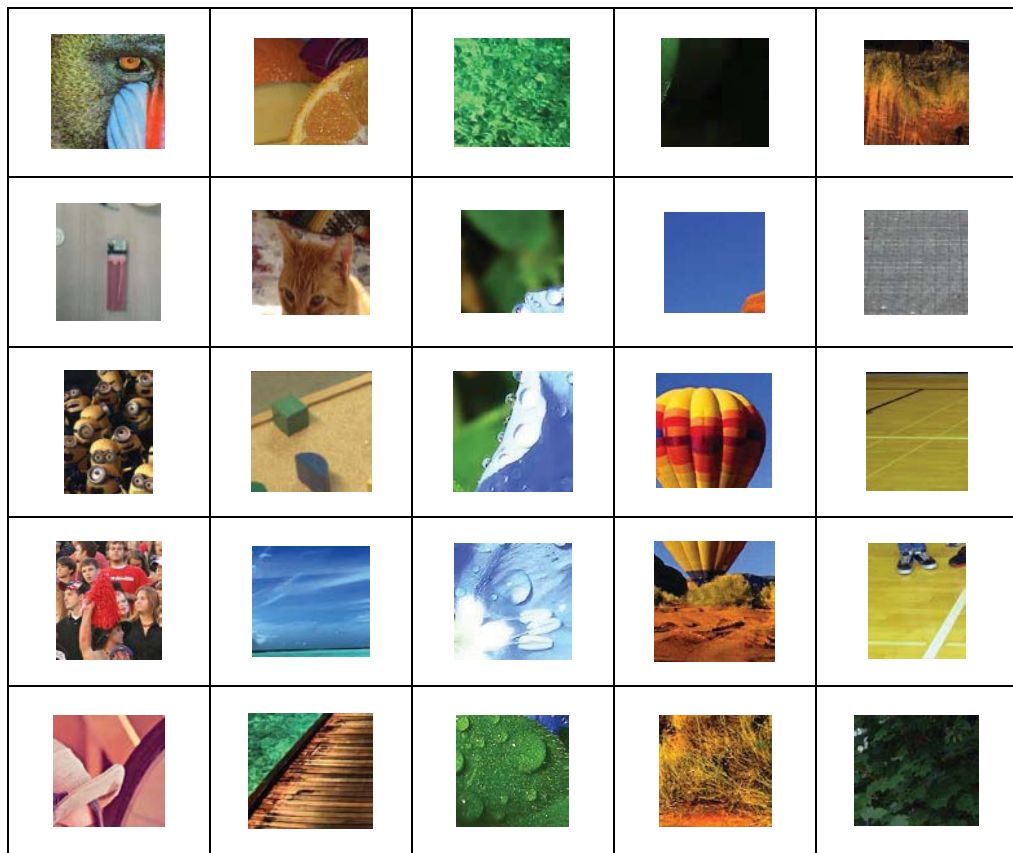


Figure 7: Negative samples used in the training.

4. EXPERIMENTS

Ten rounds of experiments have been designed and carried out to assess the efficacy of the proposed separate training method. In all of the experiment, the testing image is always cube_b. When it came to the training sets, the first five experiments were carried out using the overlapped cubes from cube_a. The remaining five experiments were carried out using the overlapped cubes from cube_a and cube_e combined. All experiments used the OpenCV software installed in a Linux system. The training procedure had four steps:

1. Generating two sample lists and selecting negative samples needed for a training.
 2. Generating the respective positive samples.
 3. Combining individual training files.
 4. Training the cascade classifier.
1. Generating sample lists:

Positive and negative image lists are generated as an outcome in two separate files. Depending upon the test complexity, the negative samples are selected that may be used in the training process.

```
find ./posImgDir -name "*.jpg" | sort -V -f > posImgList.txt
find ./negImgDir -name "*.jpg" | sort -V -f > allNegImgList.txt

#====Select a subset from allNegImgList.txt====
numNeg=222
k=0
while read varLine
do
    echo "$varLine" >> negImgList.txt
    ((k++))
    if [ $k -ge $numNeg ]
    then
        break
    fi
done < allNegImgList.txt
```

2. Generating positive samples:

All of the original positive samples were already aligned in horizontal and vertical directions during the cropping process. All of the cubes in each image were then separately grouped based on their respective overlapping percentages. A rotation angle was specified along the z axis to increase the diversity of the sample space. In other words, the actual positive sample may have various degrees of rotation depending upon a specific parameter setup. For example, if a dataset contains 80 original positive images and a factor of 8 rotation degrees is desired, then a total of $80 \times 8 = 640$ training samples can be generated. The sample window size (h, w) and the size of samples were also set at this step.

```
perl ./bin/step2.pl\  
posImgList.txt\  
negImgList.txt\  
vecSampleDir\  
640\  
"opencv_createsamples\  
-bgcolor 0\  
-bgthresh 0\  
-maxxangle 0.005\  
-maxyangle 0.005\  
-maxzangle 3.141\  
-maxidev 3\  
-w 18\  
-h 18"
```

3. Merging individual training files:

The vec files of each individual sample generated from the previous steps can then be merged into a single vec file that has the vector format defined in the OpenCV packages.

```
find ./vecSampleDir/posImgDir -name '*.vec' | sort -V -f >  
./vecSampleDir/vecList.txt  
./bin/mergevec ./vecSampleDir/vecList.txt  
./vecSampleDir/allPositiveSamples.vec
```

4. Training the cascade classifier:

To train a good cascade classifier, certain important parameters need to be specified, For example, how many positive samples and negative samples should be considered and whether a sample set of balanced or skewed positive-negation ratio needs to be maintained, how many stages are adequate for an acceptable detection result, the window size and its impact on the training speed and detection rate, the minimum acceptable hit rate, the maximum allowable false alarm rate, the number and types of Harr filters, as well as the weight trim rate, etc. It is also worth mentioning that the number of positive samples specified in this step should be around 70-80% of the number of positive samples generated during the first step to avoid potential problem of running out of samples. The use of non-default values of a few parameters such as the feature types (HAAR vs LBP) can also be considered.

```
opencv_traincascade -data trainedClassifier\  
                    -vec ./vecSampleDir/allPositiveSamples.vec\  
                    -bg  negImgList.txt\  
                    -numPos          512\  
                    -numNeg          222\  
                    -numStages       20\  
                    -precalcValBufSize 512\  
                    -precalcIdxBufSize 512\  
                    -stageType       BOOST\  
                    -featureType     HAAR\  
                    -w                20\  
                    -h                20\  
                    -bt               GAB\  
                    -minHitRate       0.996\  
                    -maxFalseAlarmRate 0.500\  
                    -weightTrimRate   0.950\  
                    -maxDepth         1\  
                    -maxWeakCount     100\  
                    -mode             ALL
```


4.1 TRAINING WITH SUBSETS FROM CUBE_A

Given five subsets of positive sample from cube_a, each containing 0%, 0-25%, 25-50%, 50-75% and 75-100% overlapped objects exclusively, five training studies were conducted in which each subset mentioned above was used for the training phase and the entire image of cube_b was used for the detection phase. The test results and more detailed discussions of the experiments are given in the analysis section below.

4.2 TRAINING WITH COMBINED SUBSETS OF CUBE_A AND CUBE_E

To investigate whether merging subsets from multiple image sources would boost the classifier's performance, a series of training and testing studies using the subsets from both cube_a and cube_e were carried out. Each subset from cube_a was combined with a subset of cube_e that has the same overlapping ratio to form a new subset. The results were five new subset of larger number of samples: 0% (a + e), 0-25% (a + e), 25-50% (a + e), 50-75% (a + e) and 75-100% (a + e), respectively. This type fusion of different image source provide more insights into the behavior of the classifier as well as the effectiveness of the separate training approach.

Using the five combined subsets from cube_a and cube_e, similar five detection studies were conducted, again using each combined subset as the training set and cube_b as the detection target. Note that the samples of cube_b were also sorted into five subsets based on their overlapping ratios. As a result, 25 possible subset-to-subset detection results were obtained and in-depth analysis of the results are given in the next section.

5. ANALYSIS

5.1 DETECTION USING THE SUBSETS OF CUBE_A

In this experiment, the training set included five subsets of cube_a (0%, 0-25%, 25-50%, 50-75% and 75-100%) and the resulting five different classifiers were used to conduct the detection tests on cube_b image. This experiment setup generated 25 possible subset-to-subset results. In Table 3 and Figure 8, the results of using five training-testing subset pairs that have the same overlapping ratios are listed (in other words, 0% subset of cube_a was used for training and 0% subset of cube_b was used for testing). The sets of 0% overlapping ratios shows the best performance while the set of 25-50% overlapping ratio gives the worst detection rate. Note that the performance is measured in terms of the detection of overlapped cubes.

TABLE 3: RESULTS OF USING SUBSETS OF CUBE_A FOR TRAINING.

Subsets of cube_a overlap ratio (num. of samples)	Num. of total detected cubes in cube_b	Num. of actual overlapped cubes in cube_b	Num. of detected overlapped cubes in cube_b	Num. of missed overlapped cubes in cube_b
0% (22)	18	6	4	2
0-25% (28)	23	13	7	6
25-50% (11)	7	11	0	11
50-75% (39)	31	20	7	13
75-100% (47)	38	26	10	16

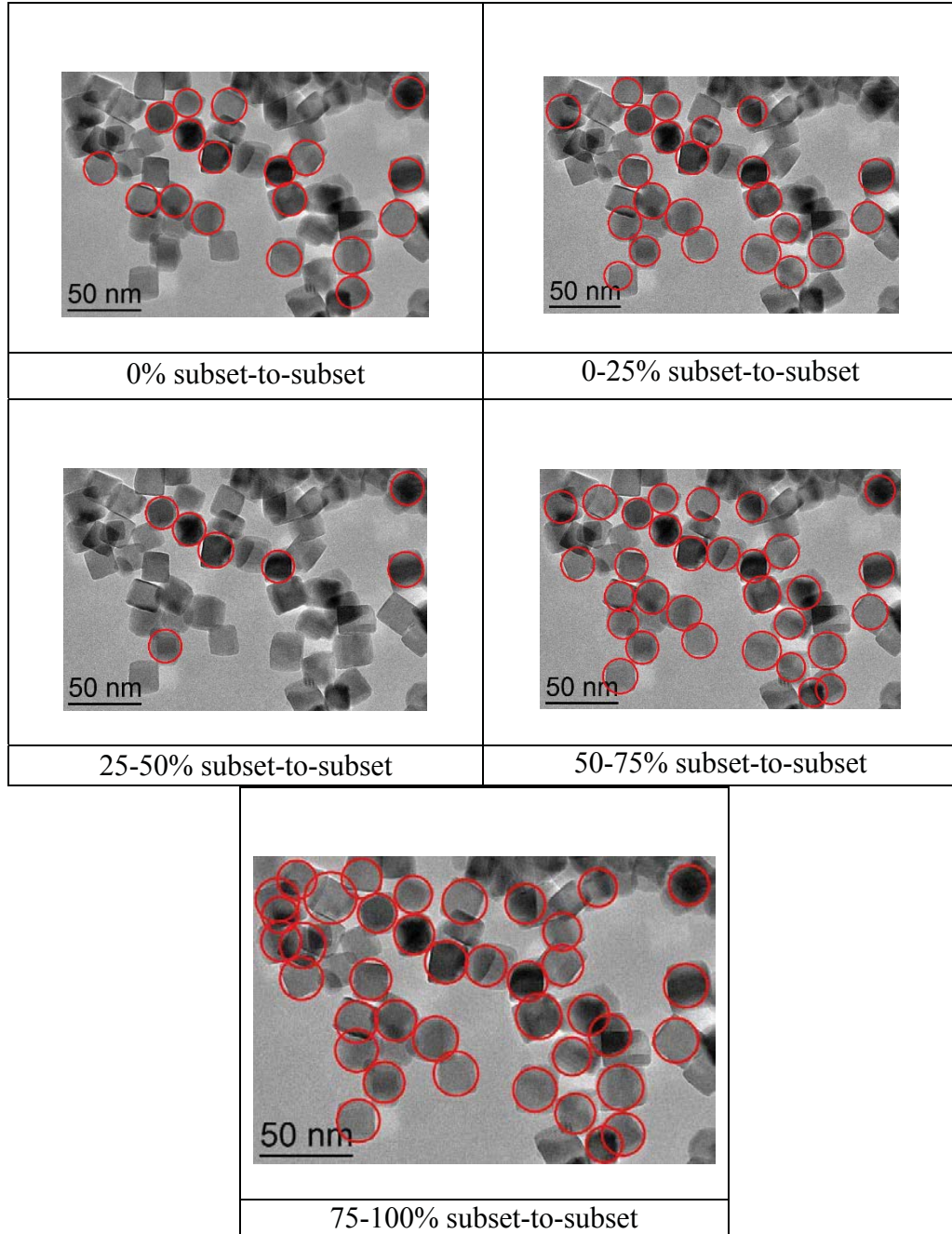


Figure 8: Detection results using the five subsets of cube_a for training.

5.2 DETECTION USING THE COMBINED SUBSETS OF CUBE_A AND CUBE_E

In this experiment, the combined subsets from cube_a and cube_e were used to train five different classifiers that were then applied to the detection task of cube_b. A total of 139 positive samples from cube_a and 112 samples from cube_e were combined to form the corresponding subsets. The goal was to find out whether there would be any additional Haar-like feature selected by the boosting process that eventually increases the detection rate. The test results were given in Table 4 and Figure 9.

It is clear that the fusion of subsets of positive samples from two data sources (cube_a and cube_e) has led to a large detection rate improvement, albeit it is only for the cases of subset-to-subset comparison based on the numbers of missed overlapped cubes. Another important observation is that the improvement is broad with respect to all subsets studies. To further analyze the potential of the proposed separate training method, a more comprehensive comparative study is presented in following section.

TABLE 4: DETECTION USING SUBSETS OF CUBE_A AND CUBE_E.

Subsets of cube_a and cube_e overlap ratio (num. of samples)	Num. of total detected cubes in cube_b	Num. of overlapped cubes in cube_b	Num. of detected overlapped cubes in cube_b	Num. of missed overlapped cubes in cube_b
0% (68)	41	6	6	0
0-25% (50)	38	13	10	3
25-50% (22)	19	11	4	7
50-75% (37)	43	20	13	7
75-100% (73)	40	26	16	10

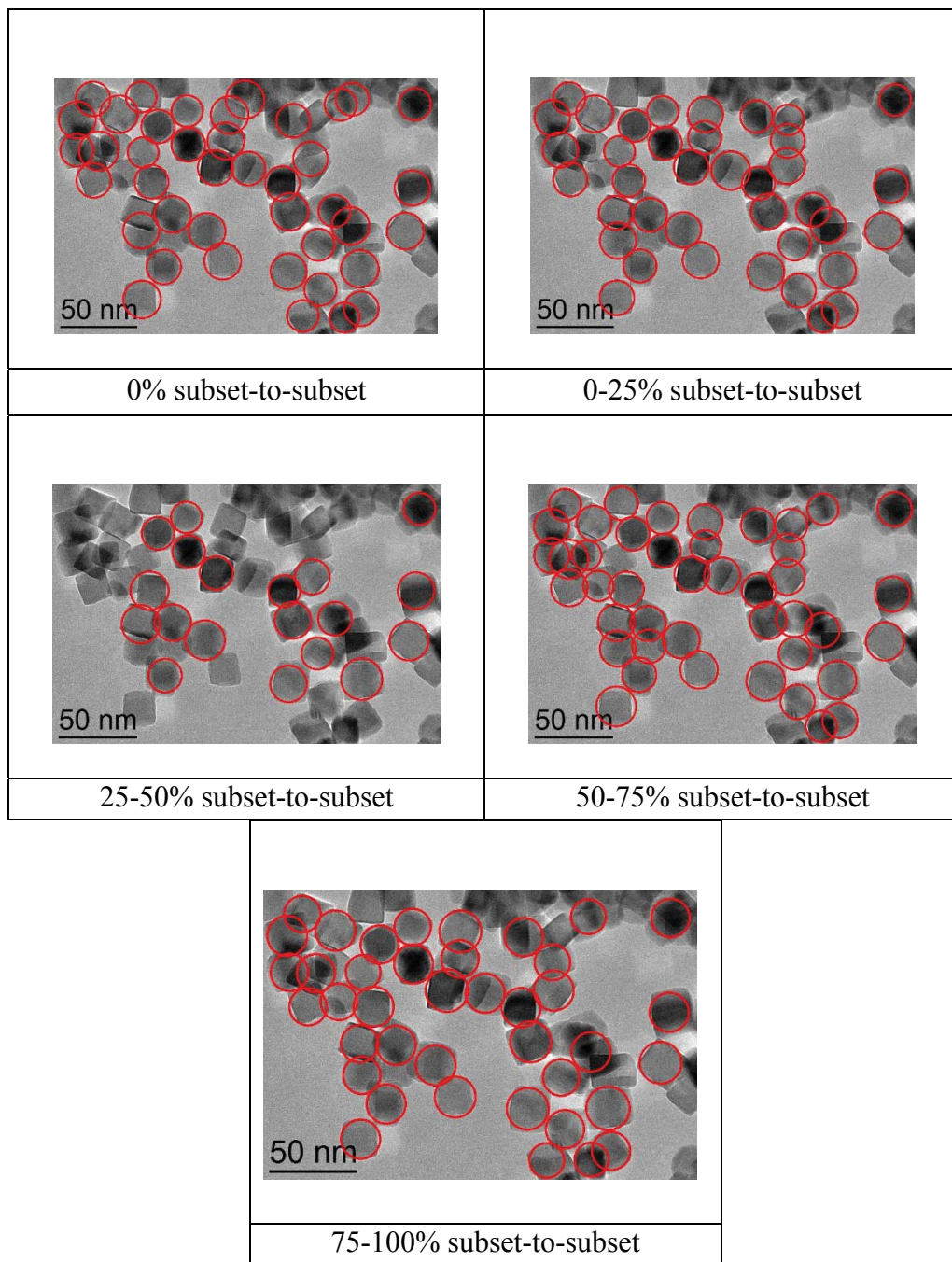


Figure 9: Results using the combined five subsets of cube_a and cube_e for training.

5.3 COMPARATIVE ANALYSIS

For each testing subset of cube_b, a super training subset of cube_a was chosen that yielded the best detection rate (see Table 5). A series of comparative analyses can then be made with respect to the corresponding super training subset. This approach might reveal the connections between the degree of sample overlapping and a classifier's behavior and performance. The results may also help us design a better separate training strategy using various subsets that are different than the ones used in this study. For example, the selection of most discriminative subsets or features, the number of subsets, as well as the overall effectiveness of the divide-and-conquer strategy via subgrouping.

TABLE 5: SUPER TRAINING SUBSETS (CUBE_A + CUBE_E)

Training subsets (a + e)	Total num. of detected cubes in cube_b.	Detection result on five subsets of cube_b				
		0%	0-25%	25-50%	50-75%	75-100%
0% (68)	41	6/6 (100%)	10/13 (76.92%)	7/11 (63.63%)	14/20 (70%) (super subset)	13/26 (50%)
0-25% (50)	38	6/6 (100%)	10/13 (76.92%)	7/11 (63.63%)	12/20 (60%)	8/26 (30.76%)
25-50% (22)	19	6/6 (100%)	5/13 (38.46%)	4/11 (36.36%)	4/20 (20%)	8/26 (30.76%)
50-75% (37)	43	6/6 (100%) (super subset)	12/13 (92.30%) (super subset)	9/11 (81.81%) (super subset)	13/20 (65%)	18/26 (69.23%) (super subset)
75-100% (73)	40	6/6 (100%)	10/13 (76.92%)	8/11 (72.72%)	13/20 (65%)	16/26 (61.53%)

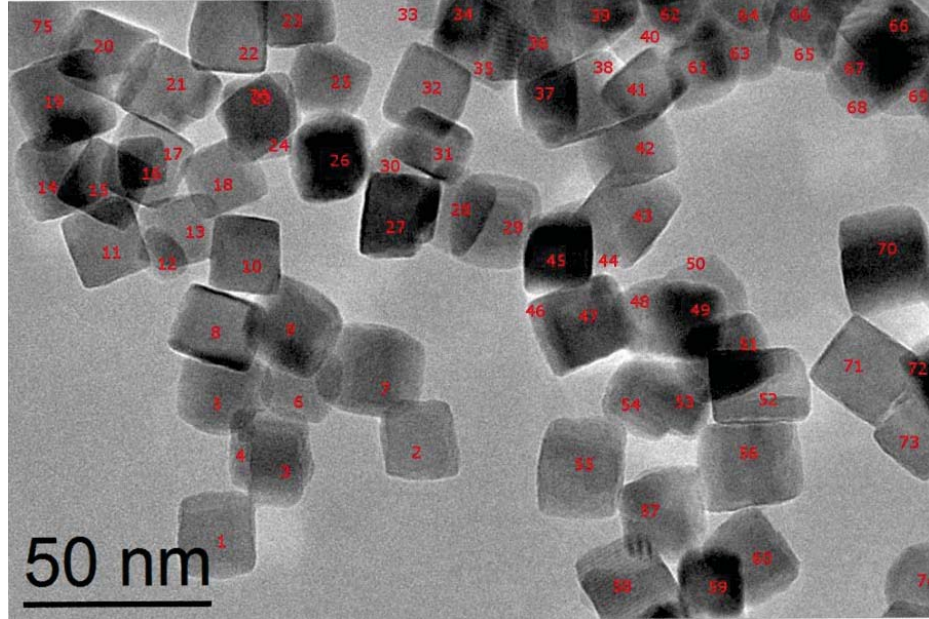


Figure 10: Labelled particles in the image of cube_b.

For four of the testing subsets of cube_b, the subset of 50-75% overlapping ratio in (a+e) was chosen as the super training subset, while for the testing subset of 50-75% overlapping ratio of cube_b, the subset of 0% overlapping ratio of (a+e) was chosen as the super training set. To facilitate the discussion of classifier's performance, the test image with labeled cubes was provided in Figure 10. The following discussion was made with respect to each testing subset of cube_b.

1) Result on 0% testing subset of cube_b:

For this group, the super training subset is chosen to be 50-75% overlapped samples in (a+e), which includes 37 positive samples. The number of 0% overlapped samples in cube_b is 6, with their labels as: 2, 10, 26, 55, 70, and 71. In fact, all training subsets enabled the detection of 6 objects. So the selection of super set is arbitrary and the improvement is minimal (see Table 6).

TABLE 6: RESULTS ON SUBSETS OF 0% OVERLAP RATIO IN CUBE_B

Training subset (a+e)	Actual num. of 0% overlapped cubes in cube_b	Detected	Missed	Improved
0% (68)	6	6	0	0
0-25% (50)	6	6	0	0
25-50% (22)	6	6	0	0
75-100% (73)	6	6	0	0

2) Result on 0-25% testing subset of cube_b:

In this group, the super training subset is 50-75% overlapped samples of cube (a+e). The sample size remained the same as that of the previous comparison. In the testing image of cube_b, the number of 0-25% overlapped samples is 13, with their labels as: 1, 11, 22, 23, 27, 32, 41, 43, 45, 46, 47, 60 and 74. The super subset detected 12 of them and the missing object has a label of 74. All other four training subsets didn't show any improvement (Table 7).

TABLE 7: RESULTS ON SUBSETS OF 0-25% OVERLAP RATIO IN CUBE_B

Training subset (a+e)	Actual num. of 0-25% overlapped cubes in cube_b	Detected	Missed	Improved
0% (68)	13	10	3	0
0-25% (50)	13	10	3	0
25-50% (22)	13	5	8	0
75-100% (73)	13	10	3	0

3) Result on 25-50% testing subset of cube_b:

For this group, the super training subset is also the 50-75% overlapped samples of (a+e). In the testing image of cube_b, the number of 50-75% cubes is 11, with their labels as: 7, 8, 9, 16, 21, 31, 33, 49, 52, 57 and 59. The super subset detected 9 of these 11 cubes. The labels of two missed cubes are 49 and 52. It is interesting to note that all other four training subsets detected cube 49, even though they all have lower overall detection rates than that of the super subset. This example clearly demonstrate the rationale of using subsets to train different classifiers that could detect the hard samples that otherwise would be missed by a single super subset or the entire sample population.

TABLE 8: RESULTS ON SUBSETS OF 25-50% OVERLAP RATIO IN CUBE_B

Training subset (a+e)	Actual num. of 25-50% overlapped cubes in cube_b	Detected	Missed	Improved
0% (68)	11	7	4	1 (label: 49)
0-25% (50)	11	7	4	1 (label: 49)
25-50% (22)	11	4	7	1 (label: 49)
75-100% (73)	11	8	3	1 (label: 49)

4) Result on 50-75% testing subset of cube_b:

In this group, the super training subset is 0% overlapped samples of (a+e) and the sample size of this training subset is 68. In the testing image of cube_b, the number of 50-75% overlapped samples is 20, with their labels as: 14, 15, 18, 19, 20, 25, 29, 34, 37, 39, 42, 48, 54, 58, 61, 63, 65, 66, 75, and 76. The super subset detected 14 of 20 these cubes and missed the following ones: 37, 39, 42, 48, 65 and 75. One important observation is that three training subsets (0-25%, 50-75% and 75-100%) were able to detect the cubes that

were missed by the super subset, and only the subset of 25-50% did not show any change. The subset of 50-75% gave the highest number of new detections: 37, 42 and 48.

TABLE 9: RESULTS ON SUBSETS OF 50-75% OVERLAP RATIO IN CUBE_B

Training subset (a+e)	Actual num. of 50-75% overlapped cubes in cube_b	Detected	Missed	Improved
0-25% (68)	20	12	8	2 (label: 37,42)
25-50% (50)	20	4	16	0
50-75% (37)	20	13	7	3 (label: 37,42,48)
75-100% (73)	20	13	7	2 (label: 37,42)

5) Result on 75-100% testing subset of cube_b:

Finally in this group, the super training subset is 50-75% samples of cube (a+e). The sample size remained as 37. The testing image of cube_b has 26 cubes of 75-100% overlapped ratios with their labels as: 3, 4, 5, 6, 12, 13, 17, 24, 28, 30, 35, 36, 38, 40, 44, 50, 53, 56, 62, 64, 67, 68, 69, 72, and 73. The super subset detected 18 of them and missed the following ones: 35, 36, 40, 50, 62, 64, 72 and 73. It was observed that the two subsets of 0% and 75-100% overlapping ratios showed new detections: 35 and 64.

TABLE 10: RESULTS ON SUBSETS OF 100% OVERLAP RATIO IN CUBE_B

Training subset (a+e)	Actual num. of 50-75% overlapped cubes in cube_b	Detected	Missed	Improved
0% (68)	26	13	13	2 (label: 64,35)
0-25% (50)	26	8	18	0
25-50% (22)	26	8	18	0
75-100% (73)	26	17	9	1 (label: 64)

Overall, as shown in Table 11 and Table 12, the fusion of the detection result of using all training subsets showed a large increase of new detected cubes with respect to that of using the super subset from 59 to 65, with 6 being new detections. Given the total number of cubes in the image of cube_b, the improvement of the detection rate of using the proposed separate training strategy is from 77.63% to 85.52%.

TABLE 11: IMPROVEMENT USING SEPARATE TRAINING OF SUBSETS

Training subsets	Detection by super subset	Combined detection by other subsets
0%	6	6
0-25%	12	12
25-50%	9	10
50-75%	14	17
75-100%	18	20
Total	59	65

TABLE 12: OVERALL RESULTS

Total number of cubes in cube_b	76
Original number of detections	59
Original detection rate	77.63%
Number of new detections	6
Number of detection by separate training	65
New detection rate	85.52%
Improvement of detection rate	7.89%

6. CONCLUSIONS

This thesis proposed a method of separate training using sample subsets that aims to improve the overall detection rate of severely overlapped nanoparticles in TEM images. Five subsets of positive samples were manually collected from the original data set based on their overlapping ratios. In addition, all cubes in the testing image were also grouped into five subsets according to the same overlapping ratios. The results of five experiments using the classifiers trained with different subsets indicated that the fusion of test outcomes help improve the overall detection rate from 77.63% to 85.52%. The main findings of this study are:

- Regrouping the original samples into subsets based on their overlapping ratios seems an effective fusion strategy for detecting overlapped nanoparticles.
- Experiments using five separate training subsets showed a large increase of detection rate of 7.89%. More importantly, the improvement was obtained on the more difficult samples that were missed by the best single training subset.
- The exact mechanism of how the separate training with sample subsets improved the detection rate is still not clear, especially why the improvement was observed only on certain types of subsets. A more comprehensive study using a much larger number of subsets is therefore needed.

7. REFERENCES

- [1] J. R. Jinschek and S. Helveg, “Image resolution and sensitivity in an environmental transmission electron microscope,” *Micron*, vol. 43, no. 11, pp. 1156–1168, Nov. 2012.
- [2] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
- [3] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154.
- [4] Z. Wang and X. Yang, “Maximum Principle in the Unbounded Domain of Heisenberg Type Group,” in *Advanced Intelligent Computing Theories and Applications*, D.-S. Huang and K. Han, Eds. Springer International Publishing, 2015, pp. 25–33.
- [5] F. Scarselli, A. C. Tsoi, M. Gori, and M. Hagenbuchner, “Graphical-Based Learning Environments for Pattern Recognition,” in *Structural, Syntactic, and Statistical Pattern Recognition*, A. Fred, T. M. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, Eds. Springer Berlin Heidelberg, 2004, pp. 42–56.
- [6] K. Nandakumar, A. Ross, and A. K. Jain, “Incorporating Ancillary Information in Multibiometric Systems,” in *Handbook of Biometrics*, A. K. Jain, P. Flynn, and A. A. Ross, Eds. Springer US, 2008, pp. 335–355.
- [7] Z. Xu, C. Zhang, S. Zhang, W. Song, and B. Yang, “Efficient Attribute Reduction Based on Discernibility Matrix,” in *Rough Sets and Knowledge Technology*, J. Yao, P. Lingras, W.-Z. Wu, M. Szczuka, N. J. Cercone, and D. Ślęzak, Eds. Springer Berlin Heidelberg, 2007, pp. 13–21.

- [8] A. Ross and A. K. Jain, “Biometrics, Overview,” in *Encyclopedia of Biometrics*, S. Z. Li and A. Jain, Eds. Springer US, 2009, pp. 168–172.
- [9] M. Botta, “Resampling vs Reweighting in Boosting a Relational Weak Learner,” in *AI*IA 2001: Advances in Artificial Intelligence*, F. Esposito, Ed. Springer Berlin Heidelberg, 2001, pp. 70–80.
- [10] A. López-Chau, F. García-Lamont, and J. Cervantes, “Classification on Imbalanced Data Sets, Taking Advantage of Errors to Improve Performance,” in *Advanced Intelligent Computing Theories and Applications*, D.-S. Huang and K. Han, Eds. Springer International Publishing, 2015, pp. 72–78.
- [11] Y.-Q. Zhang, J. C. Rajapakse, and Wiley InterScience (Online service), *Machine learning in bioinformatics*. Hoboken, N.J.: Wiley, 2009.