An Improved Optimization Based Ada Boost Classification Based Semiconductor Crack Detection System

M. Ganga Priyadarshini, Mrs. E. M. Umaselvi M.E
[1]M.E student , [2]Assistant professor/ECE
Chandy Engineering College, Mullakkadu, Tuticorin - 628005

SYNOPSIS

In semiconductor manufacturing industry defects are determined in the die, epoxy and substrate. The defects are Cracks, fingerprints, epoxy on die, and defects due to foreign materials, such as hair, threads, fibers, dust or fluids. Small size sample set semiconductor include this detection. RGB image is an input which is capture in normal camera in the manufacturing machine. RGB image includes histogram process. Using the histogram image adaptive threshold is find. From optimum threshold range in the histogram image adaptive threshold is determined. Image segmentation is the process of segmenting the semiconductor like die, epoxy, and substrate. HIS (Hue, Saturation, Intensity) image is generated using formula to understand surface color. Then RGB image is converted into grayscale image. Using adaptive threshold value grayscale value is converted into Binary image. Extraction process have two methods. SIFT Feature extraction Gaussian smoothing is applied to compute difference of Gaussian. SIFT extraction used to find local features. Using the local feature point histogram of oriented gradient find global features. SIFT have scale and rotation variability. Using firefly algorithm to select the best features. The Firefly Algorithm (FA) is a nature - inspired algorithm which is based on the social flashing behavior of fireflies. Novel based stacking classification approach used for high classification approach. Non resampling pruning method is used to eliminate bad base learners. Ada boost classification method used in this project. To improve classification performance reduce memory and computational requirements.

I. INTRODUCTION

Automated visual inspection system used for Quality control by using cameras connected in system at timing manner. In semiconductor manufacturing industry defects are determined in the die, epoxy and substrate. The defects are Cracks, fingerprints, epoxy on die, and defects due to foreign materials .such as hair, threads, fibers, dust or fluids.

Small size sample set semiconductor include this detection.

To improve classification performance reduce memory and computational requirements. Ensampling purning method is cost effective, robust, flexible, and easy to train by operator using small sample set. Causes of defects are human interactions, low quality materials, failure of machines, unexpected events like power cut.

To run the proposed method, two different functions must be executed a large number of times. Since the time available to perform the detection of these defects may be limited, it is very important to consume the least amount of time possible. In order to reduce the overall time required for detection, an analysis of how the method accesses the input data is performed. Thus, the most efficient data structure to store the information is determined. At the end of the paper, several experiments are performed to verify that both the proposed method and the data structure used to store the information are the most suitable to solve the aforementioned problem.

II . RELATED WORK

Felzenszwalb, et.al., (2010) describe an object detection system based on mixtures of multiscale deformable part models. This system relies on new methods for discriminative training with partially labeled data. We combine a margin sensitive approach for data-mining hard negative examples with a formalism we call latent SVM. A latent SVM is a reformulation of MI-SVM in terms of latent variables. A latent SVM is semi-convex and the training problem becomes convex once latent information is specified for the positive examples.

Galar, et.al., (2012 propose a taxonomy for ensemble-based methods to address the class imbalance where each proposal can be categorized depending on the inner ensemble methodology in which it is based. In addition, we develop a thorough empirical comparison by the consideration of the most significant published approaches, within the families of the taxonomy proposed, to show whether any of them makes a difference. This comparison has shown the good behavior of the simplest approaches which combine random under sampling techniques with bagging or boosting ensembles.

Chang, et.al., (2005) propose a kernel-boundary-alignment algorithm, which considers THE training data imbalance as prior information to augment SVMs to improve class-prediction accuracy. Using a simple example, we first show that SVMs can suffer from high incidences of false negatives when the training instances of the target class are heavily outnumbered by the training instances of a non-target class. The remedy we propose is to adjust the class boundary by modifying the kernel matrix, according to the imbalanced data distribution.

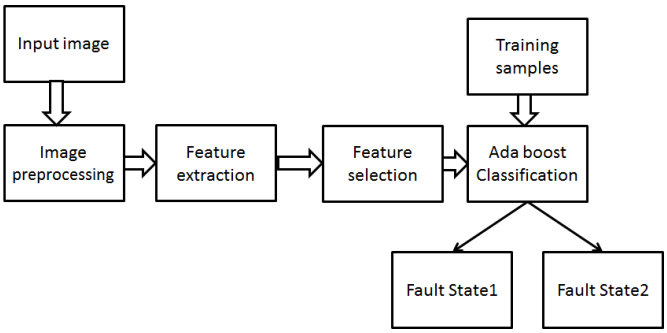
Khoshgoftaar, et.al., (2011) compares the performance of several boosting and bagging techniques in the context of learning from imbalanced and noisy binary-class data. Noise and class imbalance are two well-established data characteristics encountered in a wide range of data mining and machine learning initiatives. The learning algorithms studied in this paper, which include SMOTE Boost, RUSBoost, Exactly Balanced Bagging, and Roughly Balanced Bagging, combine boosting or bagging with data sampling to make them more effective when data are imbalanced.

Seiffert, et.al., (2010)present a new hybrid sampling/boosting algorithm, called RUSBoost, for learning from skewed training data. This algorithm provides a simpler and faster alternative to SMOTE Boost, which is another algorithm that combines boosting and data sampling. This paper evaluates the performances of RUSBoost and SMOTE Boost, as well as their individual components (random under sampling, synthetic minority oversampling technique, and AdaBoost)..

III- SYSTEM IMPLEMENTATION

RGB image is an input which is capture in normal camera in the manufacturing machine. RGB image includes histogram process. Using the histogram image adaptive threshold is find. From optimum threshold range in the histogram image adaptive threshold is determined. Then RGB image is converted into grayscale image. Using adaptive threshold value grayscale value is converted into Binary image. Extraction process have two methods. SIFT Feature extraction Gaussian smoothing is applied to compute difference of Gaussian. SIFT extraction used to find local features. Using the local feature point histogram of oriented gradient find global features. SIFT have scale and rotation variability. After feature extraction to select the best features by using firefly algorithm. Novel based stacking classification approach used for high classification approach. Non resampling pruning method is used to eliminate bad base learners. Ada boost classifier method also used in this project.

SYSTEM ARCHITEXTURE



A. PRE PROCESSING

One of the most commonly used operation in image processing is thresholding a grayscale image with a fixed value to get a binary image. For example, anything that is greater than 127 in the grayscale, can be set to 1 in the binary image and anything that is less than or equal to 127 in the grayscale image can be set to 0 in the binary image. This process is called **fixed thresholding** as our threshold value is set to 127.In **adaptive threshold** unlike fixed threshold, the threshold value at each pixel location depends on the neighboring pixel intensities. To calculate the threshold $T(x, y)$ i.e. the threshold value at pixel location (x, y)in the image, we perform the followingsteps -

1. A bxb region around the pixel location is selected. b is selected by the user.

2. The next step is to calculate the weighted average of the bxb region. OpenCV provides 2 methods to calculate this weighted average. We can either use the average (mean) of all the pixel location that lie in the bxb box or we can use a Gaussian weighted average of the pixel values that lie in the box. In the latter case, the pixel values that are near to the center of the box, will have higher weight. We will represent this value by $WA(x, y)$.

3. The next step is to find the Threshold value $T(x, y)$ by subtracting a constant parameter, let's name it param1 from the weighted average value $WA(x, y)$ calculated for each pixel in the previous step. The threshold valueT(x, y) at pixel location (x, y) is then calculated using the formula given below –

$$T(x, y) = WA(x, y) - param1$$

B. SEGMENTATION

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more

meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, suchas color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cube.

C. EXTRACTION

a) *SIFT - Scale Invariant Feature Transforms*

For any object there are many features, interesting points on the object that can be extracted to provide a "feature" description of the object. This description can then be used when attempting to locate the object in an image containing many other objects. There are many considerations when extracting these features and how to record them. SIFT image features provide a set of features of an object that are not affected by many of the complications experienced in other methods, such as object scaling and rotation.While allowing for an object to be recognised in a larger image SIFT image features also allow for objects in multiple images of the same location, taken from different positions within the environment, to be recognised. SIFT features are also very resilient to the effects of "noise" in the image.

The SIFT approach, for image feature generation, takes an image and transforms it into a "large collection of local feature vectors" Each of these feature vectors is invariant to any scaling, rotation or translation of the image. This approach shares many features with neuron responses in primate vision.

b)HOG extraction

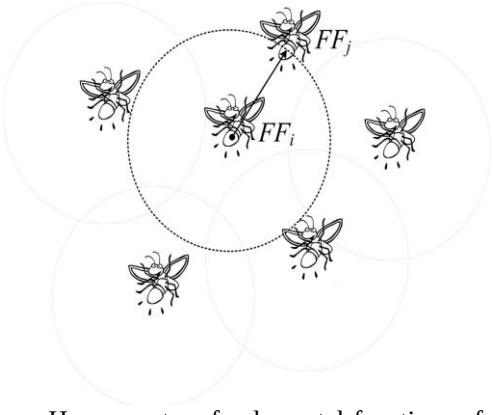
The essential thought behind the histogram of oriented gradients descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected regions called cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

The HOG descriptor has a few key advantages over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions. Moreover, as Dalal and Triggs discovered, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization permits the individual body movement of pedestrians to be ignored so long as they maintain a roughly upright position. The HOG descriptor is thus particularly suited for human detection in images.

D. FEATURE SELECTION - FIREFLY ALGORITHM

a). Behaviour of Fireflies

The flashing light of fireflies is an amazing sight in the summer sky in the tropical and temperate regions. There are about two thousand firefly species, and most fireflies produce short and rhythmic flashes. The pattern of flashes is often unique for a particular species. The flashing light is produced by a process of bioluminescence, and the true functions of such signaling systems are still debating.



However, two fundamental functions of such flashes are to attract mating partners (communication), and to attract potential prey. In addition, flashing may also serve as a protective warning mechanism. The rhythmic flash, the rate of flashing and the amount of time form part of the signal system that brings both sexes together. Females respond to a male's unique pattern of flashing in the same species, while in some species such as photuris, female fireflies can mimic the mating flashing pattern of other species so as to lure and eat the male fireflies who may mistake the flashes as a potential suitable mate.

We know that the light intensity at a particular distance r from the light source obeys the inverse square law. That is to say, the light intensity I decreases as the distance r increases in terms of $I \propto 1/r2$. Furthermore, the air absorbs light which becomes weaker and weaker as the distance increases. These two combined factors make most fireflies visible only to a limited distance, usually several

hundred meters at night, which is usually good enough for fireflies to communicate.

The flashing light can be formulated in such a way that it is associated with the objective function to be optimized, which makes it possible to formulate new optimization algorithms. In the rest of this paper, we will first outline the basic formulation of the Firefly Algorithm (FA) and then discuss the implementation as well as its analysis in detail.

b) Firefly Algorithm

Now we can idealize some of the flashing characteristics of fireflies so as to develop firefly-inspired algorithms. For simplicity in describing our new Fireflies Algorithm (FA), we now use the following three idealized rules:

1) All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex;

2) Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one than a particular firefly, it will move randomly;

3) The brightness of a firefly is affected or determined by the landscape of the objective function. For a maximization problem, the brightness can simply be proportional to the value of the objective function.

Other forms of brightness can be defined in a similar way to the fitness function in genetic algorithms.

Based on these three rules, the basic steps of the firefly algorithm (FA) can be summarized as the pseudo code shown in Fig. 1. In certain sense, there is some conceptual similarity between the firefly algorithms and the bacterial foraging algorithm (BFA). In BFA, the attraction among bacteria is based partly on their fitness and partly on their distance, while in FA, the attractiveness is linked to their objective function and monotonic decay of the attractiveness with distance. However, the agents in FA have adjustable visibility and more versatile in attractiveness variations, which usually leads to higher mobility and thus the search space is explored more efficiently.

Firefly Algorithm

Objective function $f(\mathrm{x})$, $\mathrm{x} = (x_1, ..., x_d)^T$
Generate initial population of fireflies x_i $(i = 1, 2, ..., n)$
Light intensity I_i *at* x_i *is determined by* $f(\mathrm{x}_i)$
Define light absorption coefficient γ
while *(t <MaxGeneration)*
for $i = 1 : n$ *all n fireflies*
  for $j = 1 : i$ *all n fireflies*
    if $(I_j > I_i)$, *Move firefly i towards j in d-dimension;* end if
    Attractiveness varies with distance r via $\exp[-\gamma r]$
    Evaluate new solutions and update light intensity
  end for *j*
end for *i*
Rank the fireflies and find the current best
end while
Postprocess results and visualization

Firefly algorithm

F. ADA-BOOST CLASSIFICATION

AdaBoost, short for "Adaptive Boosting", is a machine learning meta-algorithm formulated by Yoav Freund and Robert Schapire who won the Gödel Prize in 2003 for their work. It can be used in conjunction with many other types of learning algorithms to improve their performance. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. In some problems it can be less susceptible to the overfitting problem than other learning algorithms. The individual learners can be weak, but as long as the performance of each one is slightly better than random guessing (e.g., their error rate is smaller than 0.5 for binary classification), the final model can be proven to converge to a strong learner.

While every learning algorithm will tend to suit some problem types better than others, and will typically have many different parameters and configurations to be adjusted before achieving optimal performance on a dataset, AdaBoost (with decision trees as the weak learners) is often referred to as the best out-of-the-box classifier. When used with decision tree learning, information gathered at each stage of the AdaBoost algorithm about the relative 'hardness' of each training sample is fed into the tree growing algorithm such that later trees tend to focus on harder-to-classify examples.

An Adaboost classifier with the form$H(x) = \sum \alpha_t h_t(x)$ can be trained by minimizing the loss function L, i.e., byoptimizing the scalar αt and weak learner ht(x) in each iteration. Before training, every data sample xi is assigned anon-negative weight wi. After each iteration, the weights ofmisclassified samples will be heavier, which increases theseverity of misclassifying them in the following iterations.

A decision tree hTREE(x) is composed of a stump hj(x) atevery non-leaf node j. Decision trees are always trained usinga greedy procedure, recursively setting one stump at a time,starting from the root and expanding to the lower nodes. Each stump produces a binary decision, given an inputx ∈ R^K, then the stump can be parameterized with a polarityp ∈ {±1}1g, a threshold τ ∈ R, and a feature indexk ∈ {1,2,……K}

$$h_j(x) \equiv p_j \operatorname{sign}(x[k_j] - \tau_j),$$

where x[k] is the k-th dimension feature of x.The goal in each stage of stump training is to find theoptimal

parameters, which will minimize the weighted classificationerror ε

$$\varepsilon \equiv \frac{1}{Z}\sum w_i 1_{\{h(x_i)\neq y_i\}}, \qquad Z \equiv \sum w_i,$$

Given a feature k and an m-subset, the preliminary classificationerror $\varepsilon_m^{(k)}$ is defined as the smallest achievabletraining error if only the data samples in this m-subset areconsidered. That is to say, if all other samples not in thism-subset are trimmed, the preliminary classification error $\varepsilon_m^{(k)}$ is the whole classification error.

$$\varepsilon_m^{(k)} \equiv \frac{1}{Z_m}\left[\sum_{\substack{i\leq m \\ x_i[k]\leq \tau_m^{(k)}}} w_i 1_{\{y_i=+p_m^{(k)}\}} + \sum_{\substack{i\leq m \\ x_i[k]> \tau_m^{(k)}}} w_i 1_{\{y_i=-p_m^{(k)}\}}\right]$$

where $P_m^{(k)}$ and $\tau_m^{(k)}$ are both optimal preliminary parameters.We can see that using the best features of an m-subset topredict the optimal feature of the entire dataset is reasonable,but if we only use the samples in the m-subset, the training result may perform badly. So we need to find an approachwhich can reduce the samples of the m-subset as much aspossible, while having no effect on the training result.According to the properties of the upper bound of the preliminaryerror, we propose a new decision trees trainingmethod based on comparing the feature performance on subsetsof the dataset, and consequently, pruning non-effective features:

IV – RESULTS

The test images area collected from the database and the different defect of the semiconductor images is collected and tested in this work. The following figure represent the main page of this project. The following figures represents input image.

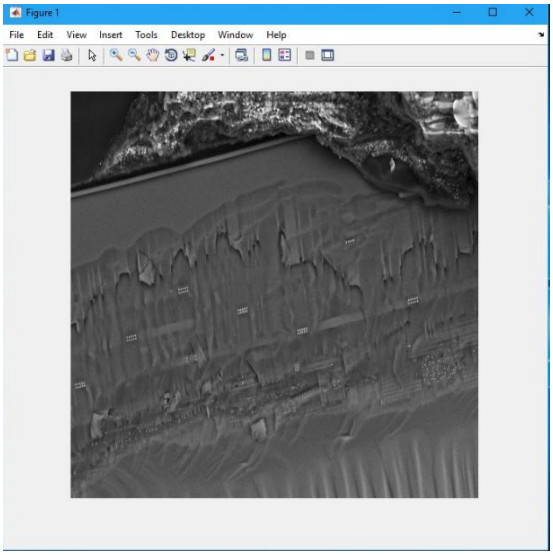


Fig 4.2 Input image

The contrast of the test image is very low because of presence default noise. This is to be suppressed and to remove the noise present in the test image. After filtering, the contrast of the test image is adjusted and shown in the following figure. Figure 4.2 represents the preprocessing the input image. It is used to convert the grayscale image.

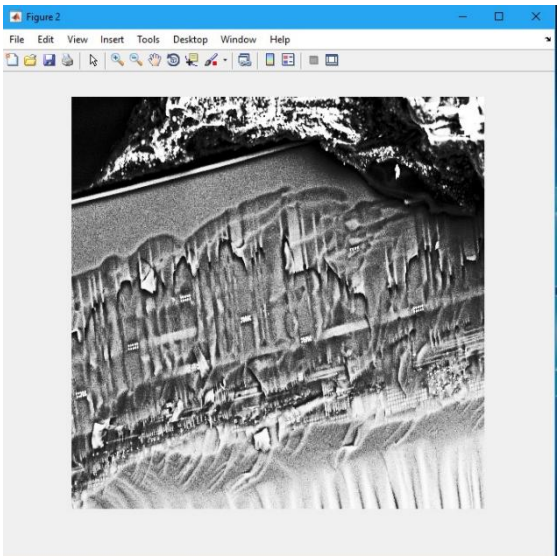


Fig 4.2 Gray scale image

The following figure 4.3 represent SIFT feature extraction. To extract the features like SIFT (Scale Invariant Feature Transforms). During SIFT extraction process to find the angle and magnitude of each pixels, figure 4.4 & 4.5 represents angle and magnitude of each pixel.

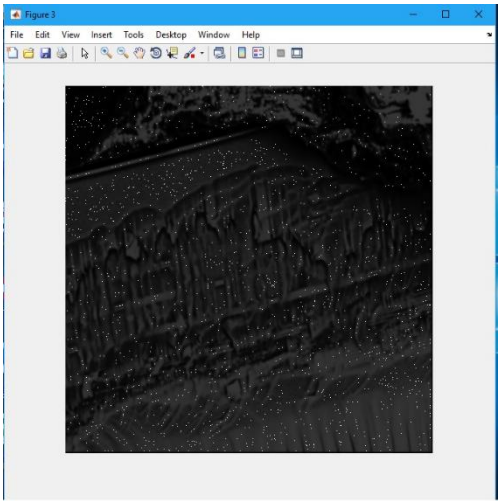


Fig 4.3 Extraction – SIFT feature

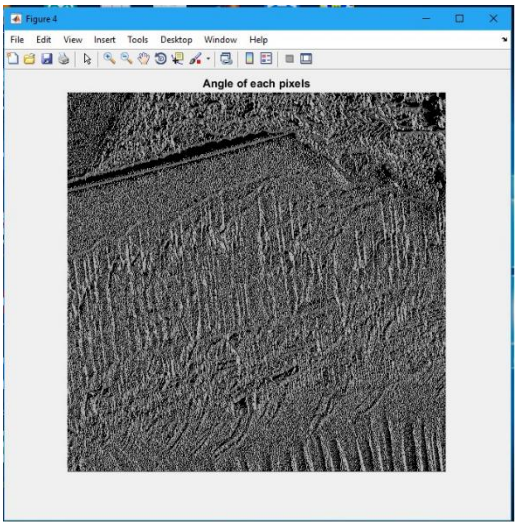


Fig 4.4Angle of each pixels

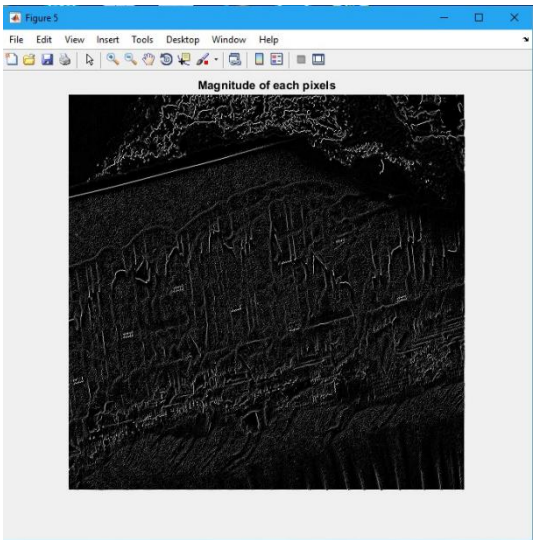


Fig 4.5 Magnitude of each pixels

The following figure 4.6 represent HOG feature extraction. HOG (histogram of oriented gradients) descriptor features. Figure 4.7, 4.8 & 4.9 represents Gabor filter results.

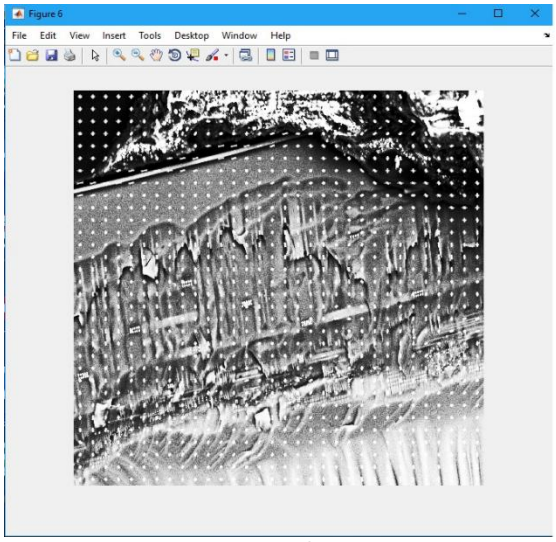


Fig 4.6 HOG feature extraction

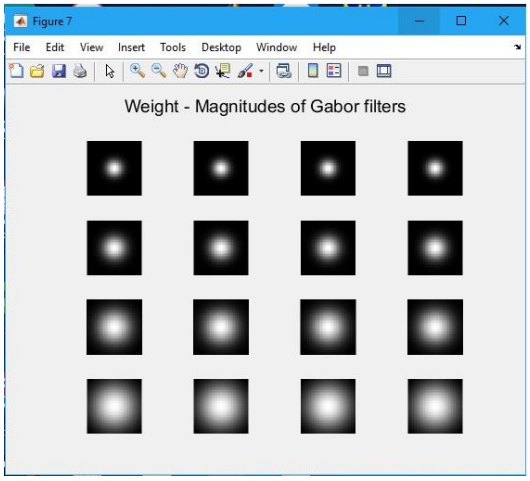


Fig 4.7 Weight – magnitudes of Gabor filters

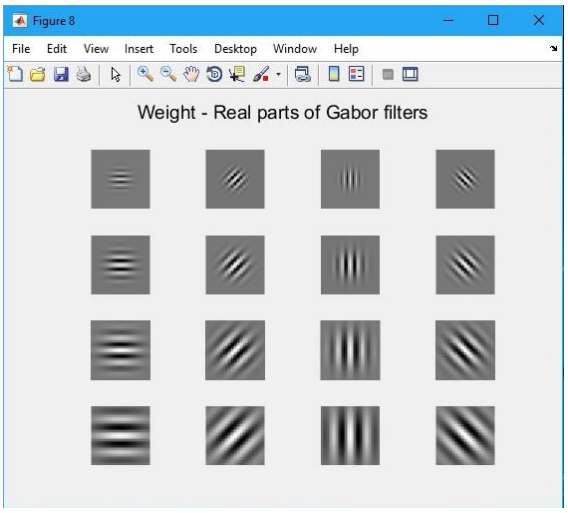


Fig 4.8 Weight – Real parts of Gabor filters

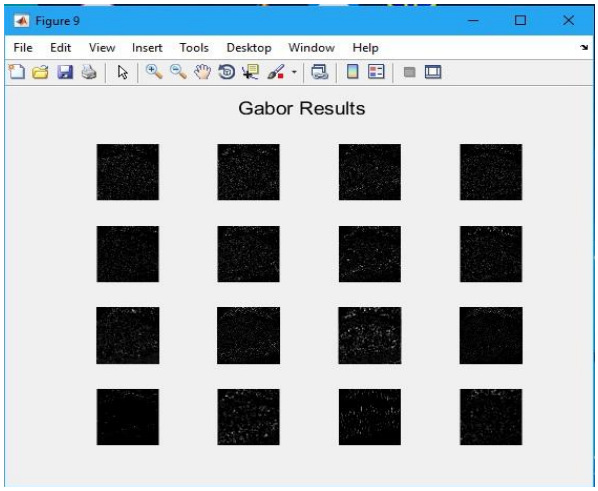


Fig 4.9 Gabor Results

The following figure 4.10 represents the classification process. To classify the semiconductor defect by using Adaboost classifier.

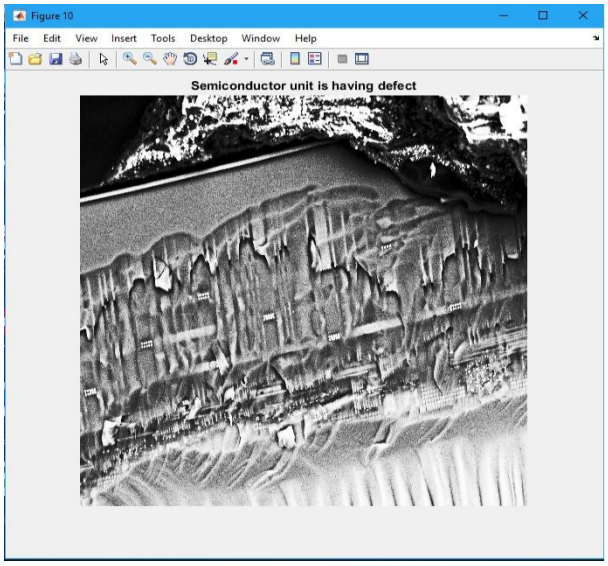


Fig 4.10 Final results

The following figure 4.11 represents the comparison of accuracy for the proposed Adaboost classifier with RF classifier and KNN classifier.

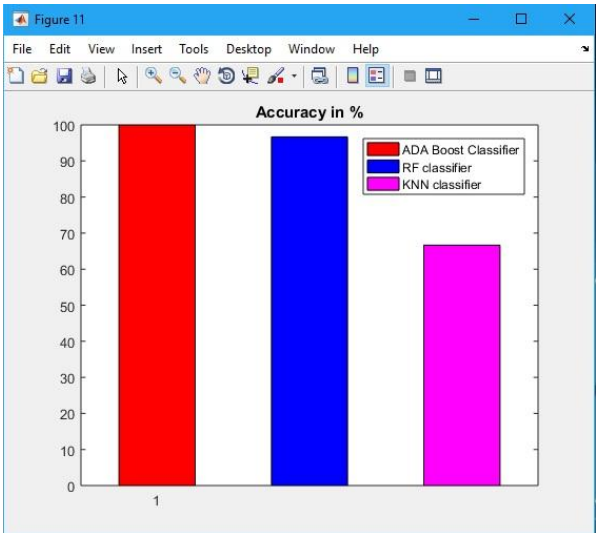


Fig 4.11 Comparison of Classification

The following figure 4.12 represents the comparison of accuracy for the proposed Adaboost classifier with optimization and without optimization.

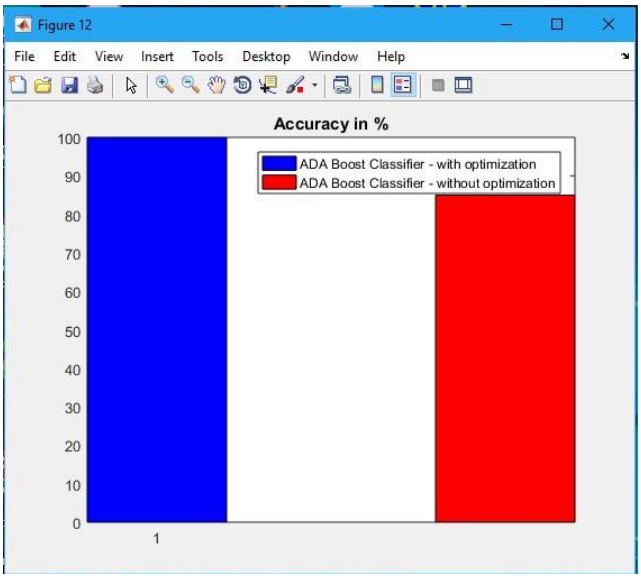


Fig 4.12 Comparison of accuracy

V. CONCLUSION

The proposed approach result than high classification accuracy as compared to existing technique by using few small training set compare to other ensemble technique like Adaboost classification algorithm. It specially deals with data imbalance problem. This system able to achieve a high qualification accuracy even a small number of samples are used for training and a high level of scalability in the sense that adding more features only improves the accuracy.

REFERENCES

[1]. C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 1–9.

[2]. K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[3]. S. Ji and J. Ye, "Linear dimensionality reduction for multi-label classification," in *Proc. 21st Int. Joint Conf. Artif. Intell. (IJCAI)*, 2009, pp. 1077–1082.

[4]. J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1794–1801.

[5]. J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality constrained Linear Coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 3360–3367.

[6]. J. Liu, S. Ji, and J. Ye, *SLEP: Sparse Learning With Efficient Projections*. Tempe, AZ, USA: Arizona State Univ., 2009. [Online]. Available: http://www.public.asu.edu/~jye02/Software/SLEP

[7]. Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 111–118.

[8]. D. A. Lisin, A. M. Mattar, M. B. Blaschko, E. G. Learned-Miller, and M. C. Benfield, "Combining local and global image features for object class recognition," in *Proc. CVPR*, San Diego, CA, USA, 2005, pp. 47–55.

[9]. W. Zhang, B. Yu, G. J. Zelinsky, and D. Samaras, "Object class recognition using multiple layer boosting with heterogeneous features," in *Proc. CVPR*, San Diego, CA, USA, Jun. 2005, pp. 323–330.

[10]. K. M. Ting and I. H. Witten, "Stacking bagged and dagged models," in *Proc. 14th Int. Conf. Mach. Learn.*, 1997, pp. 367–375.

[11]. K. Seewald, "Towards understanding stacking-studies of a general ensemble learning scheme," Ph.D. dissertation, Inst. Med. Cybern. Artif. Intell., Univ. Vienna, Vienna, Austria, 2003.

[12]. K. M. Ting and I. H. Witten, "Issues in stacked generalization," *J. Artif. Intell. Res.*, vol. 10, no. 1, pp. 271–289, 1999.

[13]. K. Seewald, "How to make stacking better and faster while also taking care of an unknown weakness," in *Proc. 19th Int. Conf. Mach. Learn.*, San Francisco, CA, USA, 2002, pp. 554–561.

[14]. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, "A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 463–484, Jul. 2012.

[15]. J. R. Quinlan, "Improved estimates for the accuracy of small disjuncts," *Mach. Learn.*, vol. 6, no. 1, pp. 93–98, 1991.

[16]. B. Zadrozny and C. Elkan, "Learning and making decisions when costs and probabilities are both unknown," in *Proc. 6th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2013, pp. 204–213.