Intelligent Systems Conference 2017
7-8 September 2017 | London, UK

Hybridization of Moth Flame Optimization and Gravitational Search Algorithm and its Application to Detection of Food Quality

Aditya Sarma, Anirudh Bhutani, Lavika Goel
Birla Institute of Technology and Science Pilani
India
asaditya1195@gmail.com, anirudh.bhutani@gmail.com, lavika.goel@pilani.bits-pilani.ac.in

***Abstract*—Gravitational search algorithm (GSA) is an optimization algorithm inspired from Newton's law of gravitation. Moth flame optimization (MFO) is another optimization algorithm, motivated by the locomotion of moths around a light source. Both of these algorithms have tried to model the search agents and altered properties like mass, gravitational constant, fitness, location, etc. in order to find the most optimal value. Optimization algorithms usually solve only a class of problems and therefore the search for a faster and more comprehensive algorithm is always on. By hybridizing MFO and GSA, the performance is expected to improve across various measures.**

This paper presents a hybrid optimization algorithm by using concepts of moth flame optimization and gravitational search algorithm and applies this hybrid algorithm to image segmentation. An optimized K-means algorithm and an optimized thresholding algorithm have been proposed. The results of the segmentation are then used to classify apples into different classes.

***Keywords*—Swarm intelligence; moth flame optimization; gravitational search algorithm; image segmentation; K-means; multilevel thresholding; automated food quality inspection**

I. Introduction

Swarm intelligence based algorithms have proven to perform well for optimization. Algorithms like, ant colony optimization, particle swarm optimization, gravitational search algorithm (GSA), biogeography based optimization and moth flame optimization (MFO) have proven to be efficient for solving optimization problems. GSA is an optimization algorithm inspired by Newton's law of gravitation. Moth flame algorithm is another optimization algorithm inspired by the moth's locomotion around a light source. Both these algorithms have their own limitations and strengths. Through creating a hybrid, the proposed algorithm aims to cover each one's limitations through the other's strengths. Additionally, it aims to reduce the convergence of the optimal value to within the error range within less number of iteration, to make the optimization more efficient.

Project food sense aims to use this improvement in order to find the degree of rottenness of various food items. This will help decrease the losses in food storage, and early detection of spoilage of food, in order to minimize monetary losses due to food and storage. This hybrid method is used in improving the results of segmentation. The hybrid optimization algorithm has been successfully incorporated in K-means and multi-level thresholding segmentation algorithms. On application of the proposed optimization algorithm to K-means clustering, the aim is to reduce the mean squared error from each data point to the centroid of the cluster. Since the initialization of the cluster centers is random in the original algorithm, the mean squared error varies with each application, even on the same dataset. Through the application of the hybrid optimization algorithm, the aim is to bring down the mean squared error and bring uniformity to the application of K-means clustering.

For multi-level thresholding, the brute force approach to finding the thresholds is a very expensive process. Finding the threshold, for multiple clusters, where each threshold could be a value anywhere between 0 and 255, the task becomes very expensive. Also, this is mostly a manual process. However, the optimization algorithm aims to bring down this complexity by using multiple search agents, each trying to find the most optimal threshold value, while communicating to each other the most optimal value. This reduces the algorithm complexity, and in turn automates the whole process of finding the right values. This technique is particularly efficient as the problem complexity increases as the dimensions or color space changes. However, this method can be extended parallel to multiple color spaces, finding the best threshold values for each color space.

Analysis is done through these improved segmentation techniques and then texture analysis techniques like grey level co-occurrence matrix, local binary pattern, etc. are explored and applied to detect the food quality. Each segment detected through the segmentation techniques mentioned above is then labeled according to its disease, and a model is created. The model is then tested using the test images. Especially with respect to food items, odor plays a significant role in identifying rottenness, like fruits releasing specific gases when rotten. Even though the current work presents a model based on vision data alone, work has started on incorporating odor data while building the model. We intend to build an automated system for classifying food items into different classes. Alpha Fox 2000 is being used for collection of odor data. The long-term aim of the project remains real time application of the proof of concepts presented in this paper. The features extracted, including the texture analysis on the vision data, and the gas sensors' readings, show great promise for our aim of reducing losses through spoilage in food storage facilities and improving food safety in messes.

II. Related Work

With respect to food safety and food quality inspection systems, a lot of research has taken place on variety of food items like apples [1], [2], bread [6], cookies [3] and vegetables [7], meat and fish to name a few. Broadly, research in this field can be classified as one of two types, 1. Sensor based techniques 2. Vision based techniques. Sensor based techniques primarily constitute odor sensors, devices like e-nose and e-tongue and recently, nano-sensors. Estimation of concentration of a specific gas is the most important factor considered when using odor based sensors. Researchers working on food contaminant detection have used nano-particle based technology like silver based nanoparticles, cerium oxide nano-particles, gold-based nano-particles [5], etc. Bio sensors [8] have also been used for food safety and food monitoring. Methods for detecting microbial contamination, pesticides, metal contaminants and mycotoxins have been explained. Vision based techniques use concepts of machine learning/computer vision techniques to extract visual features of images.

Texture, size, shape, presence of spots, etc. are some of the many factors that are taken into account when using vision based techniques. The appearance of food items, characteristics like shape, size, texture, color, etc. are useful to make a decision as to whether a food item is spoilt or not. In some scenarios, advanced techniques like multi-spectral imaging can also be used. Multi spectral imaging has been used in the past to evaluate the quality of bi colored apples [1]. On the multi-spectral images, segmentation techniques are applied and then texture related features were extracted. Statistical classifiers were trained and apples were graded into multiple classes. Apart from grading, disease identification on apples has also been performed by using vision [2]. RGB apple images were used. For texture related features, the local binary pattern features were used. Classification was performed using a standard One v/s Many SVM. Fuzzy models have also been used in this field. Features like size, shape, baked dough color, and fraction of top surface area that was chocolate chips were used to train the models. People were asked to give ratings to the cookies and accordingly four fuzzy models were developed [3]. Of those, the Mamdani inference system and Sugeno inference system gave satisfactory results. Similarly, bakery products like bread loaves were also examined. In one of the earliest research on beef, the muscle “longissimus dorsi” was located and characteristics of the longissimus dorsi were used for beef grading. Wide variety of techniques has been used in this field, both with respect to classification and feature extraction. Neural networks, fuzzy classifiers, support vector machines have thoroughly used as classifiers. With respect to feature extraction, experiments by using variety of color spaces like LAB, HSV have been done. Multispectral imaging, vibrational spectroscopy, microscopic imaging have also been used. Some more features like LBP, size, color, have also been used.

III. Optimization Algorithms

A wide range of optimization algorithms exist where each has its own method of arriving onto the optimal value. However, each algorithm solves only a class of problems. Thus, the quest for finding new optimization algorithms continues. Swarm based optimization algorithms are popular because of their inherent property to avoid local extremes. Some population based optimization algorithms include moth flame optimization [28], gravitational search algorithm [32], particle swarm optimization [30], bio-geography based optimization [29], etc.

A. Gravitational Search Algorithm

Gravitational search algorithm (GSA) is a nature inspired optimization algorithm based on Newton’s law of gravitation and Newtonian laws of motion. In this algorithm, the candidate solutions are masses spread across the hyperspace and each has an associated mass. This mass dictates the motion of the search agents, which eventually converge to the most optimal value, according the fitness function.

The algorithm starts by initializing search agents at random positions with some mass associated with each search agent. The mass of a search agent denotes how fit the agent is, i.e. the greater the mass, the greater the possibility that the mass is closer to the solutions. Gravitational search algorithm is a global algorithm, every search agent experience gravitational attraction to every other search agent in the search space, but since bigger masses move more slowly than smaller masses, as the iterations go on smaller masses will move towards the larger and more fitter masses and thus closer to the solution.

The notation for gravitational search algorithm involves three kinds of masses:

1) Active gravitational mass, M_a: M_a, is a measure of the strength of the gravitational field due to a particular object. Gravitational field of an object with small active gravitational mass is weaker than the object with more active gravitational mass.

2) Passive gravitational mass, M_p: M_p, is a measure of the strength of an object’s interaction with the gravitational field. Within the same gravitational field, an object with a smaller passive gravitational mass experiences a smaller force than an object with a larger passive gravitational mass.

3) Inertial mass, M_i: M_i is a measure of an object resistance to changing its state of motion when a force is applied. An object with large inertial mass changes its motion more slowly, and an object with small inertial mass changes it rapidly

With the above notation in mind, Newton’s force of gravity is applied.

Once the forces and accelerations on each search agent is calculated, it is time for motion. Each search agent will move in the direction of the resultant that it experiences. Gravitational search algorithm has randomization involved in the distances the search agents travel and also the direction. Each force component the agent experiences are multiplied by

978-1-5090-6435-9/17/$31.00 ©2017 IEEE

a random number and the magnitude of the distance is also multiplied by a random number, ranging from 0 to 1. It is using this randomization that gravitational search algorithm avoids local optimum values. This randomization also adds to the exploration phase of the optimization algorithm.

B. Moth Flame Optimization

Moth flame optimization is yet another nature inspired swarm optimization algorithm, based on the navigation patterns of moths around flames, also called as traverse orientation. By nature, moths maintain a fixed angle with the moon while travelling thereby moving in a straight-line path. But this gets disturbed when an external source of light, say a flame is introduced near the moth. The moth then starts maintaining a fixed angle with the flame, thereby travelling in spirals, and finally dies when it reaches the flame.

The MFO algorithm uses this spiraling of moths around flames to solve optimization problems. The algorithm starts out with 'N' number of moths and 'N' number of flames, moth traversal is guided by the flames. Fitness is associated to each flame, using a fitness function that is suitable for the problem at hand. As the algorithm continues in iterations, only the fitter flames are made to remain and the unfit flames are removed, thereby guiding the moths to the fittest flame, finally. When only one flame remains, it means that the solution has been found.

The MFO algorithm like all other optimization algorithms consists of two phases: Exploration and Exploitation. The moths that are initialized are given random co-ordinates. This randomization constitutes for the exploration phase. As the iterations of the algorithm go on, the number of flames is gradually reduced, so that finally only one remains, this is the Exploitation phase of the algorithm. As the number of flames reduces, the moths will start converging slowly. Fitness of flame depends on the specific function being optimized. Fitness of a moth is defined as the inverse of its distance to the closest flame, i.e., the closer a moth is to a flame, the fitter it is and vice versa.

IV. HYBRID OF MOTH FLAME OPTIMIZATION AND GRAVITATIONAL SEARCH ALGORITHM

The algorithm exploits the exploratory nature of the moth flame optimization (MFO), and utilizes the exploitation of the gravitational search algorithm (GSA). The intuition behind this approach to hybridize these two algorithms is that the MFO utilizes the logarithmic spiral for exploration, while the GSA uses the linear motion for locomotion. This enables the hybrid algorithm to exploit both the properties of the individual algorithm and therefore fit together. The fitness measure is mass of each search agent for GSA while the fitness measure for MFO is distance to the fittest flame. This adds the variability to exploit both the behavior in a single algorithm, thus rendering the hybrid as a better fit. MFO makes the search agent go around the fittest flame, while GSA ensures faster convergence. The flow of the algorithm is as follows (Fig. 1):

1) First, the agent population is initialized to random positions in the search domain.
2) Next, the fitness is calculated for each search agent.
3) Then the moths are associated with their serial order to the fittest flame. The first moth in the list is associated with the fittest flame, second in serial with second best flame and so on.
4) Now the search agents rotate around their associated flames in the spiral trajectory. After that, the moths attract each other and come closer to the fitter flames.

This process is repeated until the maximum number of iterations.

Locomotion for each search agent is first through the logarithmic spiral. This is the locomotion technique moths follow around a flame. This is followed by the gravitational pull of each search agent pulling each other search agents towards each other using the standard Newton's laws.

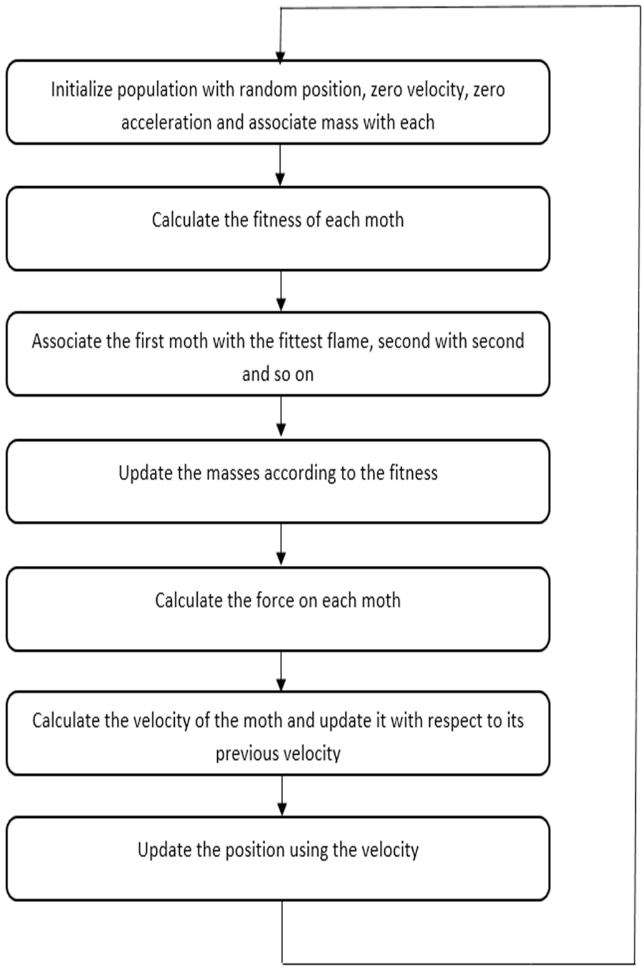


Fig. 1. Flowchart for hybrid optimization algorithm

procedure MFO_GSA_hybrid

initialize();

M = fitness();

while *iteration < max_iteration* **do**

update flame no.

978-1-5090-6435-9/17/$31.00 ©2017 IEEE

if *iteration* == 1

F = sort(M);

F` = sort(M`);

else

F = sort(M_t - 1, M);

F` = sort(M_t - 1, M);

endif

for i=1:n do

for j=1:d do

update r and t

D(i) = |F(i) – M(i)|

M(i) = D(i) * $e^{bt} * cos(2\pi t)$ + F(j)

endfor

endfor

G = (G0)*(1/*iteration*)b

for I=1:d do

Mass(i) = 1/fitness(I);

for j=1:d do

R = M_{pos}(i) - M_{pos}(j);

F = G * mass(j) / r;

endfor

M_{vel}(i) = M_{vel}(i) *rand(0,1) + F;

M_{pos}(i) = M_{pos}(i)*rand(0,1) + M_{vel}(i);

endfor

endwhile

endprocedure

PSEUDO CODE for MFO_GSA HYBRID

A. Optimizing K-means Segmentation

We have proposed an optimized K-means segmentation algorithm that improves upon the standard K-means algorithm to find better clusters.

We use the proposed hybrid optimization technique. The K-cluster centroids are treated as search agents (moths). Broadly, the optimized K-means segmentation consists of the following steps:

1) Parameter initialization: The initial K-mean points are initialized randomly from the set of pixels provided. The value of each pixel in each dimension is chosen randomly from the set of existing points, for this reason the initial K – mean points may not represent actual points in the image. But their components would be definitely being available in the image.

2) Hard Assignment of Pixel to Clusters: Given that each of the K-clusters has a mean point, every point in the set of data points is assigned to the cluster to which it is most similar. Similarity is calculated based on Euclidean distance between pixels. After this, every point is assigned to exactly one cluster C_k.

3) Parameter Re-computation: The new means of the clusters are computed based on the points assigned to the cluster in the first iteration.

Steps 2 and 3 are repeated till convergence, which typically means that no pixel shifts from one cluster to another between iterations.

procedure MFO_GSA K-means :

initialize();

for i=1:n **do**

M'[1][i] = fitness(i);

if M'[1][i] < M'[2][i] **then**

M'[2][i] = M'[1][i];

endif

endfor

for i=1:n do

for j=1:d do

update r and t

D(i) = |F(i) – M(i)|

M(i) = D(i) * $e^{bt} * cos(2\pi t)$ + F(j)

endfor

endfor

G = (G_0)*(1/*iteration*)*b;

for i=1:n **do**

Mass (i) = 1/fitness (i);

for j=1:n do

R = M_{pos}(i) - M_{pos}(j);

F = G * Mass(j) / r;

endfor

M_{vel}(i) = M_{vel}(i) *rand(0,1) + F;

M_{pos}(i) = M_{pos}(i)*rand(0,1) + M_{vel}(i);

endfor

variance = $\frac{1}{n} * \sum_{i=1}^{n}$ f_i - $f_{avg})^2$

if variance < threshold **then**

Classical_Kmeans();

978-1-5090-6435-9/17/$31.00 ©2017 IEEE

endif

endprocedure

PSEUDO CODE for OPTIMIZED K-MEANS

B. Thresholding using MFO_GSA Hybrid Optimization

In this method, the hybrid algorithm is adopted to the segmentation problem. In this method, the given image is first made noisy by adding uniformly distributed noise to the original image, i.e., $F_{ij} = f_{ij} + n_{ij}$

The basic steps of the algorithm are given as follows:

1) Initial population generation: For any pixel (i, j) the value, choose pixel value of the initial population as the value of the original pixel at (i, j) or the value of noise at (i, j). Thus, the initial population is a mixture of images with a mixture of original image values/noise values. Such a selection of the initialization helps in the exploratory phase of the optimization algorithm fitness evaluation: Fitness function is defined based on the following principle. Inter cluster similarity should be as less as possible and intra cluster similarity should be as large as possible. Thus, the fitness is defined as a ration of $S_{intercluster}/S_{intracluster}$.

2) Locomotion operations: In each iteration, the moths which represent images/potential solutions of segmentation, move due to the MFO_GSA motion, i.e., both linear as well as spiral motion. From these newly formed moth positions along with the old moth positions, the best k moths are chosen. The flames are reduced through the iterations and finally only one moth, i.e., the solution remains. This represents the segmented image.

Handling of larger images: Divide image into sub images, and final best sub images using optimization and fitness and merge them to create final segmented image.

procedure MFO_GSA Thresholding:

I = Read Image();

N = Noise_Matrix(I) // with same dimensions as I

I = I + N;

initialize();

for i=1:max_iterations **do**

while(size(population) > size(new_population)) **do**

MFO_GSA(population);

$F_i = S_{intercluster} / S_{intracluster}$;

Sort(F[])

endwhile

endfor

endprocedure

PSEUDO CODE for OPTIMIZED THRESHOLDING

V. Results and Analysis

A. Benchmark Functions

To measure the effectiveness of the optimization algorithms, the algorithms are tested against mathematical functions. The benchmark functions form the sample for evaluating the optimization algorithms. The functions are split across two types of functions: uni-modal functions and multi-modal functions. The uni-modal functions test the exploitation of the optimization functions. On the other hand, the multimodal functions test the exploration aspect of the optimization algorithms, where it is tested that the optimization function is not stuck around a local minimum. Table 1 depicts the final convergence values for Functions F1-F13 in the benchmark. Fig. 2 & 3 on the other hand represents the convergence curves of the hybrid algorithm on each of the benchmark functions as specified.

B. Optimized K-Means

Mean squared error (MSE) is calculated for standard datasets. It is clearly evident that the optimized k-means is performing better than normal k-means, as the error is less than or equal to the normal K-means. Table 2 presents the final MSE values for K-means and optimized K-means algorithms.

Function	Dim	Range
$f_1(x) = \sum_{i=1}^{n} x_i^2$	100	[−100, 100]
$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	100	[−10, 10]
$f_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	100	[−100, 100]
$f_4(x) = \max_i\{\lvert x_i \rvert, 1 \leqslant i \leqslant n\}$	100	[−100, 100]
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	100	[−30, 30]
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	100	[−100, 100]
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	100	[−1.28, 1.28]

(a)

Function	Dim	Range
$F_8(x) = \sum_{i=1}^{n} - x_i sin\left(\sqrt{\lvert x_i \rvert}\right)$	100	[−500, 500]
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10cos(2\pi x_i) + 10]$	100	[−5.12, 5.12]
$F_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	100	[−32, 32]
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	100	[−600, 600]
$F_{12}(x) = \frac{\pi}{n}\left\{10sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	100	[−50, 50]
$y_i = 1 + \frac{x_i + 1}{4}$		
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$		
$F_{13}(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\right\}$ $+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	100	[−50, 50]

(b)

Fig. 2. Benchmark Functions, F1-F13

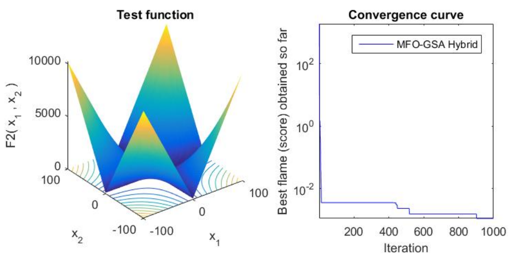
978-1-5090-6435-9/17/$31.00 ©2017 IEEE

TABLE I. COMPARING OPTIMUM VALUES OF MFO, GSA, GA AND HYBRID ALGORITHMS FOR 100 ITERATIONS

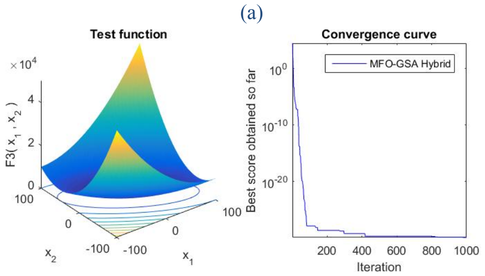
FUNCTION	MFO	HYBRID	GSA	GA
F1	6.6092e-21	5.9991e-17	1.321152	21886.03
F2	2.6592e-15	0.0010437	7.715564	56.51757
F3	7.2411e-09	2.2439e-17	736.5616	37010.29
F4	0.0043443	0.00031035	12.97988	59.14331
F5	15.7733	8.0849	77360.4184	313.21418
F6	0.12108	0.32572	2.86418	52.64496
F7	0.0046347	2.8854e-05	1.03951	20964.83
F8	-2879.4219	-2471.4722	-102.5649	-13.37504
F9	10.9445	0	25.46556	2.14891
F10	4.4409e-15	0.00011668	5.32221e-10	5.4981e-5
F11	0.22639	2.9199e-14	6.1948e-5	7.4198
F12	8.4387e-32	0.024027	1.28685e-25	0.02959
F13	1.992	2.9821	12.4944	4.5195

TABLE II. COMPARING MSE OF K-MEANS AND OPTIMIZED K-MEANS

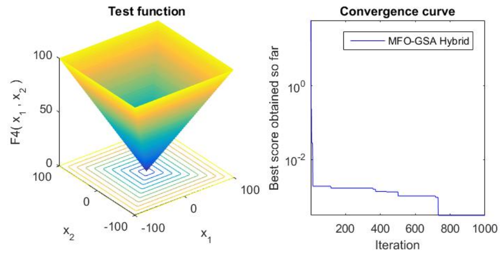
Dataset	K-Means	Optimized K-Means
Set 1	1.11	1.11
Set 2	41.59	39.54
Iris	0.54	0.52
Cancer	29.12	27.46



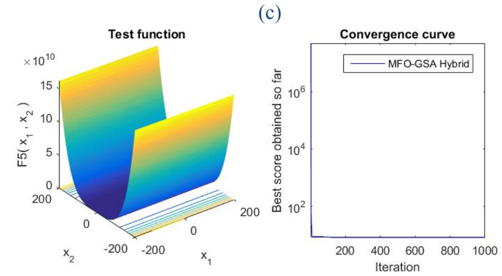
(a)



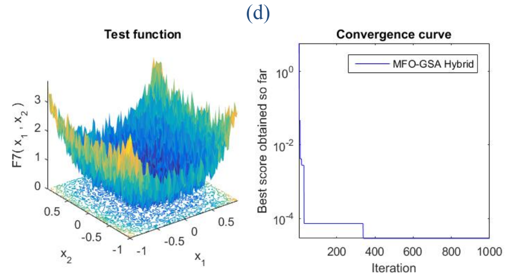
(b)



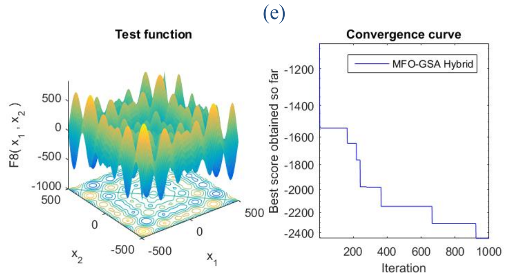
(c)



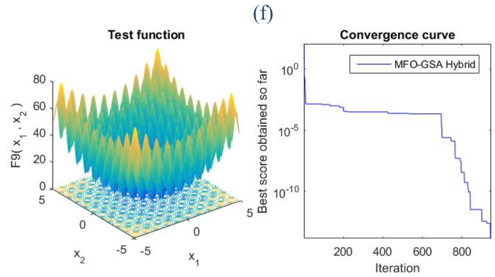
(d)



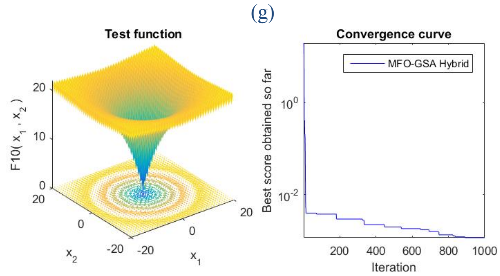
(e)



(f)

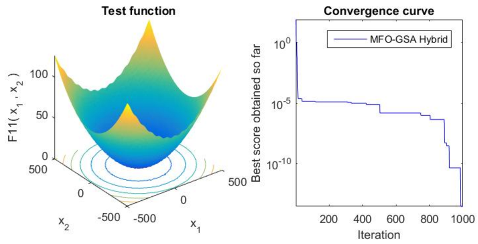


(g)

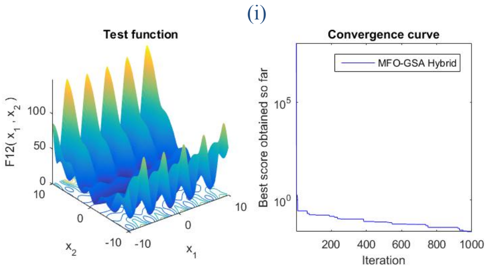


(h)

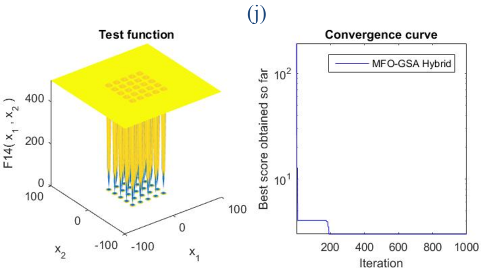
978-1-5090-6435-9/17/$31.00 ©2017 IEEE



(i)



(j)



(k)

Fig. 3. a-k : Convergence curves for benchmark Functions F2, F3, F4, F5, F7, F8, F9, F10, F11, F12, F14

The optimized K-means segmentation technique was applied on the apple dataset. The images below (Fig. 4) show the results for various types of diseased apples.

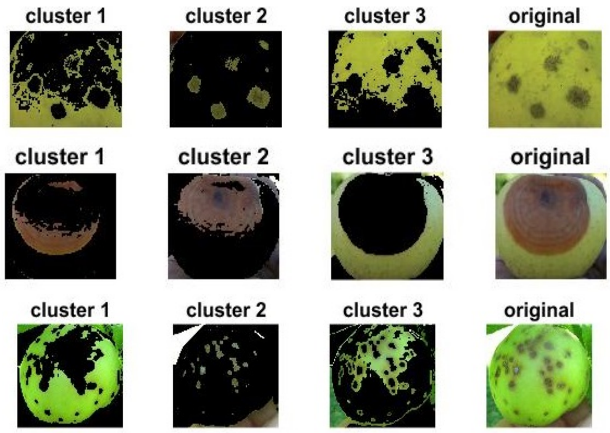


Fig. 4. Segmentation results of optimized K-means on blotch, rot and scab respectively. The diseased clusters can be identified as cluster 2 in all cases

Analysis of Multilevel Thresholding using Optimization

Multilevel thresholding using optimization algorithm has the following advantages (see Fig. 5):

1) It performs better than the K-means algorithm used above as it does not involve the factor of randomization and searches the whole search domain for the best thresholds. This reduces the error rate, and makes the algorithm more robust.

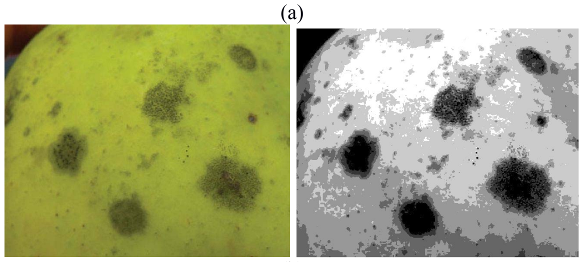
2) For grayscale images, the pixel values range from 0-255. This search space being fairly limited makes this algorithm work very fast, producing the desired results in a very small amount of time.

3) It automates the whole process of Thresholding. Thresholds are often calculated through the image and real time filters. However, that is a manual process. By automating this process, we remove the role of human, and thus improve upon the efficiency of the thresholding process.

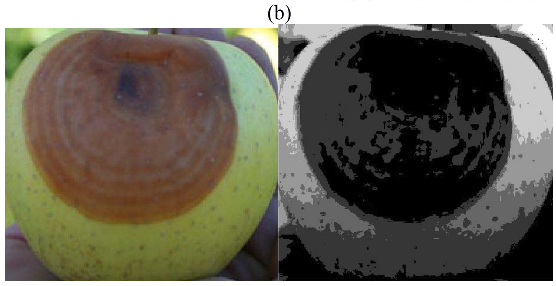
4) As the role of human is removed, the human error associated is also completely removed from the process of thresholding.

This thresholding can be expanded to any of the known color spaces, RGB, LAB, HSV, etc. Since thresholding is a very common computer vision technique, this automation results in an improvement over the current algorithms.

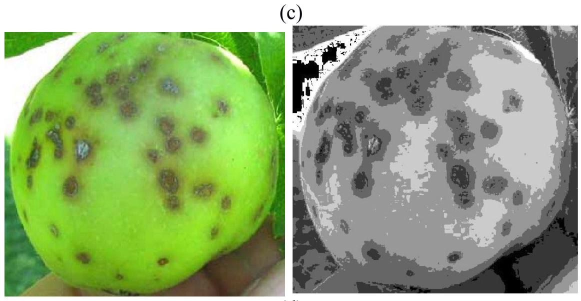
(a)



(b)



(c)



(d)

Fig. 5. a-d Segmentation: Optimized multilevel thresholding

978-1-5090-6435-9/17/$31.00 ©2017 IEEE

VI. Feature Generation

A. Local Binary Patterns

Local binary pattern (LBP) is a texture analysis technique. It assigns binary values to each pixel. These values are generated through analysis of its neighbors. It is computationally inexpensive and yet effective that it finds its application in multiple computer vision problems like facial recognition. It is a robust algorithm for it works even in different illuminations and environments.

LBP describes two dimensional textures, based on the surface in the image. One is the local spatial patterns, and the other is the grey level contrast.

B. Gray Level Co-occurance matrix

The grey level co-occurrence matrix is a distribution matrix which represents the co-occurring values in the vicinity. It signifies the relationship of a sub-image of a fixed size to its surroundings. It calculates the frequency of a pixel value occurring in its vicinity either horizontally or vertically or diagonally.

Once we have the GLCMs, one can extract information out through various statistical measures like the following:

1) *Contrast*: measures the contrast between the pixel and its surroundings.

2) *Correlation*: measures how much a pixel is correlated to its surroundings.

3) *Energy*: calculates sum of elements raised to the power of 2.

4) *Homogeneity*: calculates the closeness of the GLCM to its neighbors.

VII. Classification Results

Multi class SVM has been used for classification. The data set has been divided into the following classes: normal, scab, rot, and blotch. After segmenting using Optimized K-means and applying color based features, the following results were obtained (Table 3):

TABLE III. Confusion Matrix for Normal, Rot and Scab

Actual\Predicted	**Normal**	**Rot**	**Scab**
Normal	54	4	1
Rot	12	57	0
Scab	7	4	14

The blotch class does not have good enough features and overlaps with rot and scab. Thus, it has been excluded from the final classification.

If we analyze the multiclass model, with scab, rot and normal classes, the accuracy is 81.69%. There are a couple of reasons why the model is not as accurate: there are a few overlaps in the diseases and not a clear distinction sometimes. On doing some literature survey, it was found that the diseases can occur together, and the visual signs are often legions. For these legions to clearly depict the correct disease, it sometimes takes some time to mature.

References

[1] Nanda, Satyasai Jagannath, and Ganapati Panda. "A Survey on Nature Inspired Metaheuristic Algorithms for Partitional Clustering." Swarm and Evolutionary Computation 16 (2014): 1-18.

[2] Hatamlou, Abdolreza, Salwani Abdullah, and Hossein Nezamabadi-Pour. "Application of Gravitational Search Algorithm on Data Clustering." Rough *Sets and Knowledge Technology Lecture Notes in Computer Science* (2011): 337-46.

[3] Hatamlou, Abdolreza, Salwani Abdullah, and Hossein Nezamabadi-Pour. "A Combined Approach for Clustering Based on K-means and Gravitational Search Algorithms." *Swarm and Evolutionary Computation* 6 (2012): 47-52.

[4] Osuna-Enciso, Valentı´n, Erik Cuevas, and Humberto Sossa. "A Comparison of Nature Inspired Algorithms for Multi-threshold Image Segmentation." *Expert Systems with Applications* 40.4 (2013): 1213-219.

[5] Bülbül, Gonca, Akhtar Hayat, and Silvana Andreescu. "Portable Nanoparticle-Based Sensors for Food Safety Assessment." Sensors 15.12 (2015): 30736-0758.

[6] Duncan, Timothy V. "Applications of Nanotechnology in Food Packaging and Food Safety: Barrier Materials, Antimicrobials and Sensors." *Journal of Colloid and Interface Science* 363.1 (2011): 1-24.

[7] Yang, Tao, Huifen Huang, Fang Zhu, Qinlu Lin, Lin Zhang, and Junwen Liu. "Recent Progresses in Nanobiosensing for Food Safety Analysis." Sensors 16.7 (2016): 1118.

[8] Mutlu, Mehmet. Biosensors in Food Processing, Safety, and Quality Control. Boca Raton: CRC, 2011.

[9] Ropodi, A.i., E.z. Panagou, and G.-J.e. Nychas. "Data Mining Derived from Food Analyses Using Non-invasive/non-destructive Analytical Techniques; Determination of Food Authenticity, Quality & Safety in Tandem with Computer Science Disciplines." *Trends in Food Science & Technology* 50 (2016): 11-25.

[10] Brosnan, Tadhg, and Da-Wen Sun. "Improving Quality Inspection of Food Products by Computer Vision—a Review." *Journal of Food Engineering* 61.1 (2004): 3-16.

[11] Abdullah, Mohd Zaid, Sabina Abdul Aziz, and Abdul Manan Mohamed. "Quality Inspection Of Bakery Products Using A Color-Based Machine Vision System." Journal of Food Quality J Food Quality 23.1 (2000): 39-50.

[12] Du, Cheng-Jin, and Da-Wen Sun. "Learning Techniques Used in Computer Vision for Food Quality Evaluation: A Review." Journal of Food Engineering 72.1 (2006): 39-55.

[13] Ma, Ji, Da-Wen Sun, Jia-Huan Qu, Dan Liu, Hongbin Pu, Wen-Hong Gao, and Xin-An Zeng. "Applications of Computer Vision for Assessing Quality of Agri-food Products: A Review of Recent Research Advances." *Critical Reviews in Food Science and Nutrition* 56.1 (2014): 113-27.

[14] Ding, Zhuqing, Ruoyu Zhang, and Za Kan. "Quality and Safety Inspection of Food and Agricultural Products by LabVIEW IMAQ Vision." Food Analytical Methods Food Anal. Methods 8.2 (2014): 290-301.

[15] Miraglia, M., H.j.p. Marvin, G.a. Kleter, P. Battilani, C. Brera, E. Coni, F. Cubadda, L. Croci, B. De Santis, S. Dekkers, L. Filippi, R.w.a. Hutjes, M.y. Noordam, M. Pisante, G. Piva, A. Prandini, L. Toti, G.j. Van Den Born, and A. Vespermann. "Climate Change and Food Safety: An Emerging Issue with Special Focus on Europe." *Food and Chemical Toxicology* 47.5 (2009): 1009-021.

[16] In Brief: Achieving a Transparent, Actionable Framework for Public–Private Partnerships for Food and Nutrition Research. *Journal of the Academy of Nutrition and Dietetics* 115.8 (2015): 1318-319

[17] Goel, Lavika, Daya Gupta, V. K. Panchal, and Ajith Abraham. "Taxonomy of Nature Inspired Computational Intelligence: A Remote Sensing Perspective." *2012 Fourth World Congress on Nature and Biologically Inspired Computing* (NaBIC) (2012)

[18] Babak E. Bejnordi ; Geert Litjens ; Meyke Hermsen ; Nico Karssemeijer and Jeroen A. W. M. van der Laak. "A multi-scale superpixel classification approach to the detection of regions of interest in whole slide histopathology images", *Proc. SPIE 9420, Medical Imaging 2015: Digital Pathology*, 94200H (March 19, 2015)

978-1-5090-6435-9/17/$31.00 ©2017 IEEE

[19] Hong-Min Zhu and Chi-Man Pun. "An Adaptive Superpixel Based Hand Gesture Tracking and Recognition System".

[20] Zhao, Mingru, Hengliang Tang, Jian Guo, and Yuan Sun. "Data Clustering Using Particle Swarm Optimization." *Lecture Notes in Electrical Engineering Future Information Technology* (2014): 607-12.

[21] Li, Haiyang, Hongzhou He, and Yongge Wen. "Dynamic Particle Swarm Optimization and K-means Clustering Algorithm for Image Segmentation." Optik - *International Journal for Light and Electron Optics* 126.24 (2015): 4817-822.

[22] Koonsanit, Kitti. "Determination of the Initialization Number of Clusters InK-means Clustering Application Using Co-OccurrenceStatistics Techniques for Multispectral Satellite Imagery." *International Journal of Information and Electronics Engineering IJIEE* (2012).

[23] Kumar, Tara Sai, Mahesh Chandra M, and Sreenivasa Murthy P. "Colour Based Image Segmentation Using Fuzzy C-Means Clustering." *Int. Jour. Intel. Elec. Sys. International Journal on Intelligent Electronic Systems* 5.2 (2011): 47-51.

[24] Dubey, Shiv Ram, and Anand Singh Jalal. "Adapted Approach for Fruit Disease Identification Using Images." *International Journal of Computer Vision and Image Processing* 2.3 (2012): 44-58

[25] Zhang, Y.j. "A Survey on Evaluation Methods for Image Segmentation." Pattern Recognition 29.8 (1996): 1335-346.

[26] Ahmadyfard, Alireza, and Hamidreza Modares. "Combining PSO and K-means to Enhance Data Clustering.*" 2008 International Symposium on Telecommunications (2008)*

[27] Narwade, Jyoti, and Binod Kumar. "Local and Global Color Histogram Feature for Color Content-Based Image Retrieval System.*" Proceedings of the International Congress on Information and Communication Technology Advances in Intelligent Systems and Computing* (2016): 293-300.

[28] Mirjalili, Seyedali. "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm." Knowledge-Based Systems 89 (2015): 228-49.

[29] Simon, D. "Biogeography-Based Optimization." IEEE Transactions on Evolutionary Computation 12.6 (2008): 702-13.

[30] Kennedy, James. "Particle Swarm optimization" Encyclopedia of machine learning. Springer US, 2011. 760-766.

[31] Huang, Di, Caifeng Shan, Mohsen Ardabilian, Yunhong Wang, and Liming Chen. "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 41.6 (2011): 765-81.

[32] Rashedi, Esmat, Hosein Nezamabadi-Pour, and Saeid Saryazdi. "GSA: A Gravitational Search Algorithm." *Information Sciences 179.13* (2009): 2232-248

978-1-5090-6435-9/17/$31.00 ©2017 IEEE