

An optimal segmentation framework for early crack and fire detection using Potoo swarm optimization algorithm

Praveen Sankarasubramanian¹[0000-0001-7711-9040] and Dr. En Ganesh²

¹ Research Scholar, Computer Science and Engineering, Vels Institute of Science Technology & Advanced Studies, Palavaram, Chennai, Tamilnadu India

² Dean School of Engineering, Vels Institute of Science Technology & Advanced Studies, Palavaram, Chennai, Tamilnadu India

Abstract

The process of dividing an image into multiple segments or regions with similar color, texture, and intensity is known as segmentation. Simplifying or making the image representation more meaningful and easier to analyze is the goal of image segmentation. For computer vision, it is used, medical imaging, and remote sensing applications for object recognition, tracking, and classification. Segmentation is needed for damage and fire images to identify and isolate the regions of interest in the images. In the case of damage detection, the segmentation can help to identify the areas of the structure that are damaged or compromised. This information can be used to assess the severity of the damage and plan for repairs or maintenance. Similarly, in the case of fire detection, segmentation can help to isolate the regions of the image that contain flames or smoke, which can be used to trigger alarms and alerts for rapid response and fire suppression. Overall, segmentation helps to extract the relevant information from the image, which can be used for decision-making and action planning. This research introduces a new framework for detecting damage and fire early on in images by segmenting them optimally using the Potoo swarm optimization (PTSO) algorithm. To enhance the accuracy of crack/fire detection, we first apply a benchmark preprocessing technique that removes unwanted artifacts and enhances image quality. Then, we design a novel PTSO algorithm inspired by the nocturnal vision and hunting skills of Potoo birds, which specifically targets crack/fire segmentation. We evaluate the performance of proposed PTSO segmentation framework on benchmark datasets and compare its results with existing optimal segmentation frameworks.

Keywords: image segmentation, crack detection, fire detection, preprocessing, Potoo swarm optimization

1. Introduction

The process of dividing an image into multiple segments or regions with similar characteristics like color, texture, or intensity is known as image segmentation [1][2]. The motivation behind division is to improve or change the picture portrayal into a more significant and effectively analyzable structure. It is a significant stage in numerous vision applications, including object acknowledgment, following, and picture based robot route [3]. Image segmentation can be used in a lot of different areas. See segmentation helps in identifying and tracking objects within an image or a video sequence. It is extensively used in medical imaging applications such as MRI, CT scans, and ultrasound imaging to isolate and

analyze specific parts of the body or organs for diagnosis and treatment [4]. It plays a significant role in robotics applications such as object manipulation, object recognition, and path planning. It can be used to detect vehicles, pedestrians, and traffic signals, which is useful in intelligent traffic control systems. It is used in agriculture applications for crop yield estimation, crop disease detection, and plant health monitoring [5][6]. Image segmentation helps in detecting and tracking objects and people in video surveillance systems.

Crack detection [7][8] often requires image processing because cracks can be difficult to identify visually, especially in large structures or when the cracks are small. Enhancing the contrast

between the crack and the surrounding material and eliminating image noise are two possible outcomes of applying image processing techniques. This makes it easier to detect and measure cracks accurately, which is important for assessing the structural integrity of buildings, bridges, and other infrastructure. In addition, image processing can help automate the crack detection process, making it faster and more consistent than manual inspection. Fire detection [9][10] using image processing is useful in situations where fires occur in locations that are not easily visible or accessible to humans. For example, in industrial settings or in large buildings, a fire may start in a location that is not easily visible to human operators or firefighters, such as in ductwork or behind walls. In situations like these, image processing methods are used to find fires and give building occupants or the fire department early warning. Image processing can also be used to monitor the spread and intensity of a fire, providing valuable information for firefighters to make decisions about how to contain and extinguish the fire. Additionally, image processing can be used to analyze thermal images to identify hotspots or areas of high heat, which can indicate potential fire hazards or areas where fires may start in the future.

Image segmentation is a challenging task, and there is no one-size-fits-all algorithm that works for all images. Optimization algorithms, such as metaheuristic algorithms [11][12], can search for optimal solutions in large search spaces and avoid getting stuck in local minima. These algorithms can be used to optimize the parameters of segmentation algorithms or to directly partition the image into regions based on a given objective function. By using optimization algorithms, works on the exactness and proficiency of picture division, making it a valuable instrument for different applications, including medical image analysis [13], remote sensing, and computer vision [14][15].

Our contributions. A novel optimal segmentation framework is proposed for early damage and fire detection using the Potoo swarm optimization

(PTSO) algorithm. The main contributions of our proposed framework is describes as follows.

1. The first step in the proposed framework is the utilization of a benchmark preprocessing technique for image preprocessing. This technique removes unwanted artifacts from the image and enhances the image's quality, allowing for more efficient detection of cracks and fire.
2. The next step involves the design of a novel Potoo swarm optimization (PTSO) algorithm for targeted crack and fire segmentation from the preprocessed image, which ensures high detection accuracy. The PTSO calculation is a metaheuristic improvement calculation enlivened by the way of behaving of hummingbirds. These birds are known for their exceptional nocturnal vision and hunting skills. The algorithm mimics the behavior of these birds in the optimization process and ensures that the segmentation results are accurate and reliable.
3. Finally, our PTSO segmentation framework is checked against benchmark datasets and compared with existing benchmark optimal segmentation frameworks. This comparison demonstrates the effectiveness and the proposed system's superiority in terms of fire efficiency, accuracy, and segmental damage. The proposed system contributes to the early detection of fire and damage leading to better prevention and mitigation of potential hazards.

The structure of the paper is as follows: The second section provides a summary of recent research in this area to crack/fire segmentation using optimization techniques. Section 3 describes the problem methodology and the proposed framework's system design Section 4 provides a comprehensive explanation of the working process of the proposed framework, along with the mathematical model. The results and a comparison of the segmentation schemes are discussed in Section 5. The paper comes to a conclusion in Section 6.

2. Related works

In this section, we will explore the different approaches and methods used in the literature for crack and fire detection in images. We will also discuss the limitations and challenges of the

existing approaches and highlight the need for a new and improved approach.

Zou et al. [16] involves the use of DeepCrack, a deep convolutional neural network made to learn high-level crack representation features for automatic crack detection. At various stages of convolution, the network acquires a number of deep convolutional features and combines them to extract line structures. They use a Segnet-based encoder-decoder architecture and combine convolutional features developed in both networks at the same level to accomplish this network was trained on one crack dataset and tested on three others, achieving an F-measure of over 0.87 according to experimental results. Yang et al. [17] proposed a pavement crack detection using a brand-new network architecture known as the Feature Pyramid and Hierarchical Boosting Network (FPHBN). This design coordinates setting data into lower-level capabilities to recognize holes in a pyramid-style capability pecking order. During training, it also balances the contribution of light and hard samples by using the weight of clustered samples, allowing for effective crack detection. The average intersection per joint (AIU) in five distinct crack data sets is used to assess the efficiency of FPHBN. The findings demonstrate that, in terms of generalization and accuracy, the proposed approach performs better than the most recent methods for crack, edge detection, and semantic segmentation.

Choi et al. [18] proposed Standard convolutions, Densely Coupled Separable Convolution (Densep) modules, a modified Atros Spatial Pyramid Pooling (ASPP) module, and decoder modules make up the brand-new SDDNet convolutional neural network. SDDNet is prepared on a hand-created cleavage dataset with a mean convergence join (mIoU) of 0.846 on the test set. The authors also presented representative segmentation results for each test image, demonstrating the effectiveness of the SDDNet in segmenting cracks unless the features are too faint. In addition, the SDDNet's performance was contrasted with that of the most recent models, and it was found to outperform them in terms of evaluation metrics despite having 88 times fewer parameters than the compared

models. Zhang et al. [19] proposed a break location structure utilizing a regulated generative ill-disposed learning approach, called the crack-patch-only (CPO) model. The CPO model is designed to overcome the issue of all-black outputs from the U-Net model by feeding a larger-sized crack image into an asymmetric U-shape generator. The model is trained on partially accurate ground truth generated using a labor-light method, which reduces the workload of preparing ground truth significantly. Additionally, the proposed approach addresses the issue of data imbalance, which can be byproduct of traditional supervised learning methods. The CPO model shows impressive performance, even when trained on small image patches with partially accurate ground truth, and is capable of processing full-sized images.

Chen et al. [20] proposed Predicting spatial accuracy, a hybrid axis convolution network (HACNet) keeps the same spatial resolution throughout the architecture. HACNet achieves accurate segmentation with relatively few parameters by employing a hybrid strategy that combines coarse convolutions with multiple features and an appropriate amount of dilation. Break discovery undertakings require the result to answer picture districts sufficiently enormous to catch data about huge items, and by picking a proper degree of development in the break organization, an enormous powerful collector (ERF) can be gotten without utilizing a downscaling plan. Chiang et al. [21] proposed a transfer learning scheme and a synthetic approach to enhancing the imagery dataset are used in this method, which uses a retrained mask RCNN to automatically detect dead trees in aerial images. They inferred from derived annotations in COCO format and randomly assigned patterns to the raw data's background using synthetic tools. They discovered that mask prediction was constrained by the ROIs chosen, and that semantic segmentation was more applicable to the dead than exemplar segmentation tree region detection. The best model achieved a mean average precision score (mAP) of 54%.

Khan et al. [22] proposed a system that detects and separates smoke in bright and clear environments. They used EfficientNet, a high-accuracy efficient CNN architecture, for smoke detection. Smoke districts are divided utilizing DeepLabv3+, which involves effective encoders and decoders with pixel characterization for ideal restriction. Smoke detection accuracy increased by up to 3% in their findings, while average intersection and union (IoU) scores and global accuracy decreased by 0.46 percent and 2%, respectively for segmentation. Zhang et al. [23] have introduced a novel architecture, ATT Squeeze U-Net, which combines SqueezeNet, U-Net, and attention mechanisms for accurate forest segmentation. The model uses attention gates in the skip connections to highlight informative features and suppress irrelevant ones, resulting in improved segmentation performance with a relatively small number of parameters. The SqueezeNet architecture is modified by replacing classical convolution layers with depth-wise ones and introducing a Channel shuffle operation in the Fire module to enhance feature communication. In U-Net, the encoder is this reworked SqueezeNet ATT squeeze, and the decoder has a DeFire module for it. The proposed model accomplishes a high exactness of 0.93 and a normal assessment season of 0.89 seconds per picture, which is reasonable for ongoing backwoods division errands.

Chen et al. [24] have developed a fire recognizable proof calculation joining fire division and multi-capability combination. The authors proposed two improved YCbCr models for reflective and non-reflective lighting conditions, respectively, to enhance the performance of YCbCr models for fire distribution under various lighting conditions. By comparing the areas obtained from the two samples, conditions that reflect and conditions that do not reflect can be distinguished. Using the relationship between the initial point and neighboring points, an improved region growing algorithm is used to apply an improved fire segmentation method. The weighted average of the centroid coordinates of each segmented image is used to calculate the origin points. At last, the writers proposed quantitative files of fire

identification in view of the coefficient of variety of the fire region, the spread of the centroid, and the edge of the portioned regions. Rashkovetsky et al. [25] have proposed another technique for recognizing fire inclined regions from satellite pictures utilizing single-input CNN with U-net design. With the best multi-sensor fusion scenario (Sentinel-2 and 3) and the best fit sensor (Sentinel-2), cloud-free weather detection rates of up to 92% were achieved. Additionally, it improves efficiency accuracy of fire detection, aiding in early response and mitigation efforts.

3. Problem Methodology

Image segmentation for crack and fire detection is the need for efficient and accurate automated methods for identifying and localizing cracks and fires in various scenarios. Traditional manual inspection methods are time-consuming, labor-intensive, and often subjective, leading to potential errors and inconsistencies. On the other hand, automated image segmentation methods can quickly and objectively analyze large amounts of data, leading to more accurate and reliable results. In the case of crack detection, image segmentation can help identify and classify cracks based on their size, shape, and location. This can aid in identifying potential safety hazards in structures such as bridges, dams, and buildings, which can ultimately improve public safety. Similarly, in the case of fire detection, image segmentation can help locate the boundaries of fires and identify the extent of damage caused by the fire. This can aid in efficient firefighting and damage assessment, which can ultimately save lives and minimize property damage. Therefore, image segmentation for crack and fire detection is an important research area that has the potential to improve safety and save lives. Traditional image segmentation methods suffer from various limitations. In the case of crack detection, traditional methods, such as thresholding and edge detection, often fail to detect small and low-contrast cracks, and require manual adjustment of parameters for different crack types and images. Similarly, for fire detection, traditional methods, such as thresholding and pattern recognition, may fail to detect fires due to the complexity and

variability of fire appearance in different imaging conditions, such as smoke and shadow occlusions. Moreover, traditional image segmentation methods are often computationally expensive, which limits their applicability in real-time or large-scale scenarios. They also rely heavily on feature engineering, which can be time-consuming and challenging to optimize for different applications. Therefore, there is a need for more advanced image segmentation methods that can address these limitations and provide accurate and efficient crack and fire detection.

While optimization algorithms can be useful for image segmentation tasks, there are still several potential problems that can arise. One common issue is that optimization algorithms can be computationally expensive and time-consuming, especially for large datasets. This can limit the scalability of the algorithm and make it impractical for real-time or high-throughput applications. Another problem is optimization algorithms may be sensitive to the choice of hyperparameters and initialization conditions, which can lead to suboptimal results or even convergence failure if not set appropriately. This requires careful tuning and parameter selection, which can be a challenging and iterative process. Finally, optimization algorithms are susceptible to overfitting, especially when applied to small or imbalanced datasets. This can result in poor generalization performance and make the algorithm less effective at detecting new or previously unseen cracks or fires. Regularization techniques and careful data augmentation can help mitigate this problem, but it remains an ongoing challenge in the field. Researchers have proposed several objectives to solve the problems in image segmentation for crack and fire detection, including:

1. Incorporating more advanced pre-processing techniques to enhance image quality, remove noise, and increase the accuracy of the segmentation results.
2. Developing new and improved loss functions and metrics that can better handle class imbalance, spatially varying errors, and the challenges of real-world scenarios.

3. Developing new techniques to improve the generalization and transfer learning capabilities of the segmentation models, especially for scenarios with limited labeled data.
4. To validate the performance of proposed model with different test samples to prove the effectiveness in terms of quality measures.

Fig. 1 shows Overall system design of proposed work which consist of following process. Consider dataset with more data's of crack and fire images: This refers to the first step of the proposed work, which involves collecting a large dataset of images related to crack and fire. The dataset should have a sufficient number of images to train the model and achieve high accuracy. After collecting the dataset, the next step is to select the relevant images related to crack and fire that will be used for image processing and segmentation. In image preprocessing, several steps involve removing artifacts and noise from the images and enhancing their quality to improve the accuracy of the segmentation process. Image preprocessing is an important step as it helps to remove unwanted features from the images and makes them more suitable for processing. After preprocessing, we perform target segmentation using Potoo swarm optimization (PTSO) algorithm: In this step, the preprocessed images are segmented using the proposed optimization algorithm, which is the Potoo swarm optimization (PTSO) algorithm. The algorithm works by optimizing the threshold values used to segment the images. Then, get the segmented image of crack and fire with corresponds to given image: Finally, the segmentation results are obtained in the form of segmented images of crack and fire corresponding to the input images. The segmented images can be further analyzed to identify the extent of damage caused by cracks and fire, and suitable remedial measures can be taken accordingly.

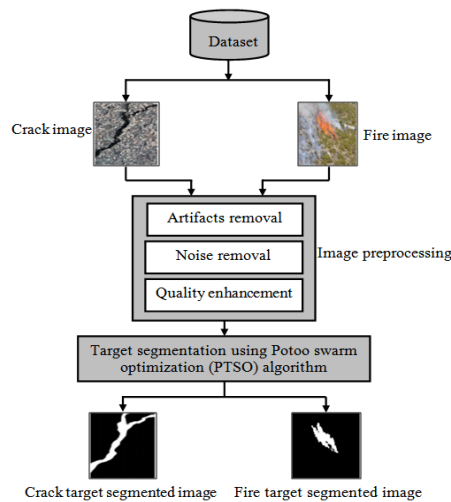


Fig. 1 Overall system design of proposed work

4. Proposed Methodology

4.1 Image preprocessing

Image preprocessing refers to the operations performed on an image before its analysis or interpretation. It is a critical step in image processing because it can significantly affect the accuracy and speed of subsequent image analysis. Image preprocessing techniques can include operations such as noise reduction, contrast enhancement, image normalization, artifact removal, image registration, and geometric correction. The purpose of image preprocessing is to remove any imperfections or inconsistencies in the image and to enhance the information content of the image for better interpretation and analysis.

1. In image processing, artifacts refer to any unwanted visual distortions or anomalies present in the image. These distortions can arise due to various reasons such as sensor noise, transmission errors, compression artifacts, etc. Artifacts removal is the process of removing such distortions or anomalies from an image to improve its quality and make it more suitable for further analysis or use. This process can involve various techniques such as filtering, interpolation, and image enhancement methods, among others. The aim of artifacts removal is to enhance the quality of the image by removing any unwanted visual anomalies that may be present.

2. Noise removal refers to the process of reducing or eliminating noise or unwanted elements from an image. In digital images, noise appears as random variations in brightness or color information, and it can be caused by factors such as low light conditions, high ISO settings, or sensor limitations. Noise can affect the quality and accuracy of image analysis and can cause errors in segmentation, classification and other tasks involving computer vision. In this way, clamor wiping out essential step in image processing and is typically performed using various filters or algorithms that can smooth the image while preserving its details and structures.

3. Quality enhancement in image preprocessing refers to the improvement of the visual quality of an image by enhancing its brightness, contrast, sharpness, color balance, and removing unwanted artifacts and noise. This process is particularly useful when dealing with low-quality images, such as those captured in low light conditions, or those that have been degraded due to compression or transmission over a network. By enhancing the quality of an image, it can be easier to detect and segment objects of interest in the image, such as cracks or fires, and improve the accuracy of subsequent processing steps.

Artifacts, which are caused by factors such as poor lighting, camera distortion, or compression, can lead to errors in segmentation. By removing artifacts, the image becomes cleaner and easier to segment accurately. Noise in the image can also interfere with segmentation accuracy, as it can obscure important details or create false edges. By removing noise, the image becomes smoother and less cluttered, making it easier to distinguish between object boundaries and background. Quality enhancement techniques can help to improve the overall clarity and detail of the image, making it easier to see and identify features for segmentation. This can be particularly important in cases where the image quality is poor, such as in low-light or low-resolution images. Overall, by improving the quality of the image through these preprocessing techniques, segmentation algorithms can more accurately identify and

delineate the target objects, leading to better overall performance.

4.2 Target segmentation using Potoo swarm optimization (PTSO) algorithm

The process of dividing an image into multiple regions or segments that correspond to distinct regions or objects is known as image segmentation. It gives each pixel in the image a label or class certain criteria such as color, intensity, texture, or shape. In the case of crack and fire images, image segmentation is necessary because it helps to identify the exact location and extent of the crack or fire. This information can then be used for further analysis or decision-making purposes, such as determining the severity of the damage, identifying potential safety hazards, or planning repairs or mitigation strategies. Without segmentation, it would be difficult to isolate the affected area from the background or other objects in the image, which could lead to inaccurate or incomplete information.

Potoo swarm optimization (PTSO) is a metaheuristic optimization algorithm based on the behavior of Potoo birds, which are known for their ability to blend into their surroundings and stay still for long periods of time. The PTSO calculation is a populace based technique that iteratively refreshes various potential answers for track down the ideal answer for a given issue. The PTSO algorithm can be used to improve image segmentation results by optimizing the segmentation parameters. In particular work, PTSO algorithm is used for target segmentation, which involves separating the areas of interest (cracks and fires) from the background. The algorithm works by iteratively updating a population of candidate segmentation solutions, called Potoos, based on their fitness value, which is calculated using an objective function that evaluates how well the segmentation solution fits the image data. The Potoos are guided by their individual experiences, as well as the collective experience of the swarm, to explore the search space and find the optimal segmentation solution. The use of the PTSO algorithm for segmentation

has several advantages, including the ability to handle noisy and complex data, the ability to handle non-convex optimization problems, and the ability to efficiently explore a large search space. Additionally, the algorithm is flexible and can be adapted to different image segmentation tasks by adjusting the objective function and other parameters. The following are the steps involved in the Potoo swarm optimization (PTSO) algorithm for image segmentation:

1. The first step is to initialize the population of potoo agents randomly. The number of agents in the population is typically set to a fixed value.
2. Each potoo agent in the population is assigned a fitness value based on its ability to segment the image.
3. The global best position is updated based on the fitness value of each potoo agent.
4. The velocity and position of each potoo agent are updated using the current and global best positions.
5. A local search is performed on a subset of the population to improve the quality of the solutions.
6. The process of fitness evaluation, global best update, velocity and position update, and local search are repeated until a stopping criterion is met. This stopping criterion can be a maximum number of iterations, a convergence threshold, or other similar measures.
7. The final positions of the potoo agents are used to segment the image into the desired classes.
8. Post-processing techniques such as morphological operations may be applied to the segmented image to further refine the segmentation results.

Here is the mathematical model for the PTSO algorithm is start with the position updation of j^{th} Potoo at time/ iteration in the search space is indicated by a vector $y^{j,iter}$, where $j = 1, 2, \dots, N$, $iter = 1, 2, \dots$, and $iter \max$ represents the maximum iteration number.

$$y^{j,iter+1} = \begin{cases} y^{j,iter} + r \times fl^{j,iter} \times (m^{k,iter} - y^{j,iter}) \\ \text{arandomposition} \end{cases}$$

(1)

where r is a random number between 0 and 1, fl is the Potoo's flight length at iteration $iter$, and m are

$$m^{j,iter+1} = \begin{cases} y^{j,iter} + 1, & \text{if } F(y^{j,iter+1}) \text{ is better than } F(m^{j,iter+1}) \\ m^{j,iter}, & \text{otherwise} \end{cases}$$

where $F(\cdot)$ indicates the fitness value, and $m^{j,iter}$ is the hidden place of k^{th} Potoo at iteration $iter$. According to the previous update equation, Potoo updates its memory with the new position if the new position is superior to the fitness function value for the memorized position. In order to use the PTSO algorithm for crack and fire segmentation, several parameters need to be set up. These parameters include the number of decision variables, the number of Potoos, the awareness probability, and the maximum number of iterations. Once these parameters are set, the next step is to assign each Potoo to a random point in the search space and initialize their memory. During the main loop of the algorithm, each Potoo updates its position based on its own memory and the memory of the best-performing Potoo in the swarm. The awareness probability is used to determine the level of influence that the best-performing Potoo has on each individual

$$m^{i,iter+1} = \begin{cases} y^{i,iter} + 1, & \text{if } F(y^{i,iter+1}) \text{ is better than } F(m^{i,iter+1}) \\ m^{i,iter}, & \text{otherwise} \end{cases}$$

(4)

where $F(\cdot)$ indicates the fitness value, and $m^{i,iter}$ is the hidden place of k^{th} Potoo at iteration $iter$. After updating the position of each Potoo and evaluating their fitness function values, we check if the maximum number of iterations has

the Potoo's concealed place and awareness probability at iteration $iter$, respectively. The fitness value of the new place is then determined, and the Potoo's memory is updated as follows::

(2) Potoo. The algorithm goes on iterating until it reaches its maximum number of iterations. When the algorithm reaches convergence, the final position of each Potoo represents a possible solution to the crack and fire segmentation problem.

(3)

Each Potoo's position quality is determined by calculating the objective function value after substituting the decision variable values. To create a new position for a Potoo at iteration $iter$, one Potoo is randomly selected, and the other Potoos follow this Potoo to find its hidden food. This process is repeated for all Potoos, resulting in new positions for each. The feasibility of the new positions is evaluated based on the awareness probability. If a new position is feasible, the Potoo moves to the new location and also follows the global best position (x_{best}) and population mean. The fitness function is evaluated in the same way as described earlier.

been reached. If the current iteration is less than the maximum number of iterations, we repeat steps 4 to 7 to continue optimizing. Otherwise, we stop the optimization process. The working steps involved in the proposed target segmentation process using Potoo algorithm is given in Algorithm 1.

Algorithm 1 Target segmentation process using Potoo algorithm

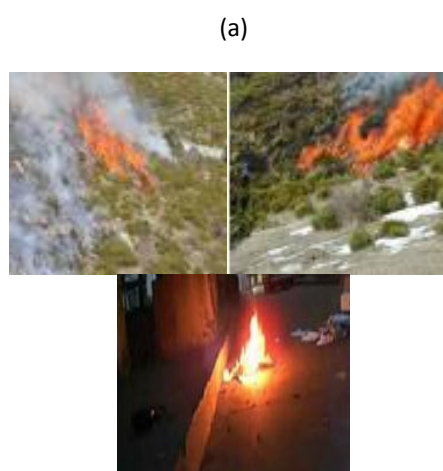
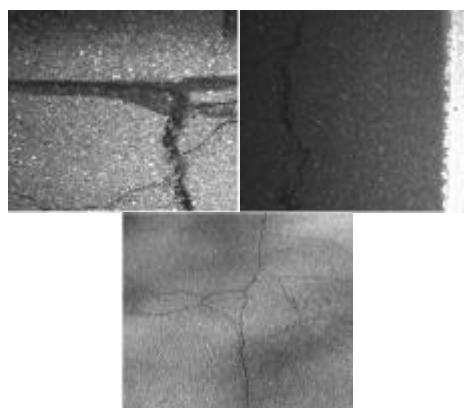
Input : Image: the input image to be segmented, N: number of Potoos, w: inertia weight, Max_iter: maximum number of iterations, Threshold: threshold value for binarization

Output : Segmented image

1	Initialize: Swarm: initialize population of N random Potoo, Best_ Potoo: the Potoo with the best fitness value, V: initialize velocity for each Potoo Fitness: initialize fitness for each Potoo
2	For iter in range(Max_iter)
3	Update velocity
4	Update position
5	Compute fitness value of the Potoo based on segmented image
6	Update the best fitness value and best particle
7	Update Gbest
8	End For
9	End

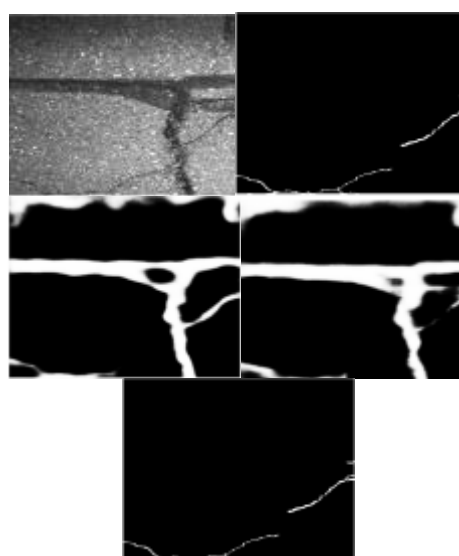
5. Experimental results and analysis

In this section, we presents the experimental results and analysis of poropsoed PTSO algorithm based segmnettaion with exisintg benchamrk optimization algoiriothms, such as particle swram optimization (PSO) and ant colony optimization (ACO). The study used a dataset consisting of 12,000 crack and fire images to validate the performance of both the proposed and existing segmentation methods. The dataset is evenly split between the two types of images, with 6,000 images of each. The validation is performed using several folds, which are divided into three interval epochs: 0-50%, 51-80%, and 81-100%. The test samples used in the validation are shown in Fig. 2, and are taken from the crack and fire dataset. The segmentation results of proposed and exiisntg optimization algoiriothms are given in Fig. 3 and 4 for crack and fire imaegs, resepectvely. The performnaec can be validtaed through different measures such as true positive, false positive, true negative, false negative, accuracy, precision, recall, and F-measure.

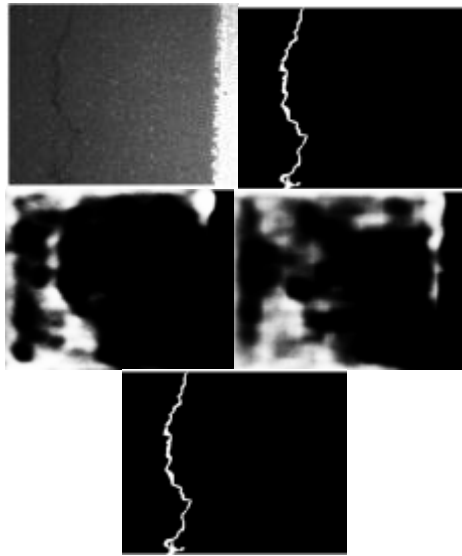


(b)

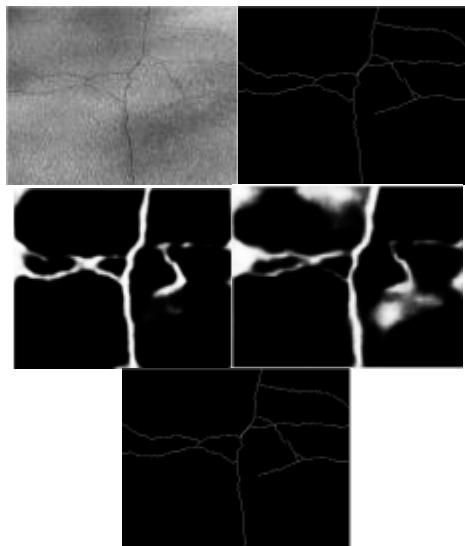
Fig. 2 Test sampels form dataset (a) Crack images (b) Fire images



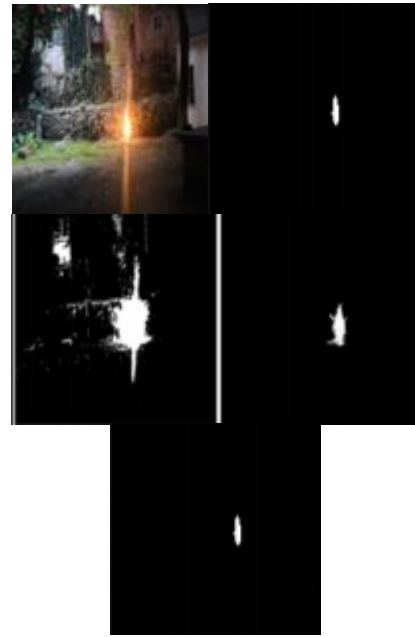
(a)



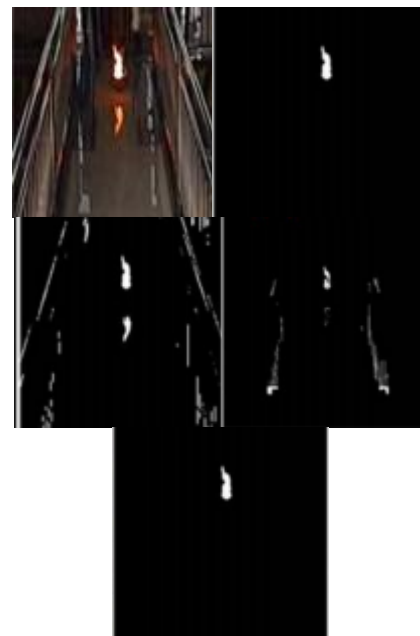
(b)



(c)



(a)



(b)

Fig. 3 Segmentation results with input image, groundtruth image, results of PSO, results of ACO and results of PTSO algorithm (order in image left to right) for crack image (a) sample-1, (b) sample-2, (c) sample-3



(c)

Fig. 4 Segmentation results with input image, groundtruth image, results of PSO, results of ACO and results of PTSO algorithm (order in image left to right) for fire image (a) sample-1, (b) sample-2, (c) sample-3

Table 1 shows the results of PSO-based segmentation for a dataset of 12,000 samples of fire and crack images, with different training sample percentages ranging from 10% to 100%. The measures evaluated in this table include true positive, false positive, true negative, false negative, accuracy, precision, recall, and F-measure. From the results, it can be observed that the accuracy and other measures vary depending on the percentage of training samples used. The highest accuracy of 79.167% was achieved with 80% training samples, while the lowest accuracy of 30.125% was achieved with 20% training samples. Precision and recall values also vary with the percentage of training samples. The highest precision of 87.205% was achieved with 100% training samples, while the highest recall of 89.567% was achieved with 90% training samples. The F-measure, which is a measure of both precision and recall, also varied with the percentage of training samples. The highest F-measure of 91.497% was achieved with 100% training samples. In general, the results suggest that using a higher percentage of training samples

improves the accuracy and other measures of PSO-based segmentation for fire and crack images. However, it is important to note that the performance of the algorithm may vary depending on the specific characteristics of the dataset used.

Table 2 shows the results for ACO-based segmentation using different percentages of training samples. The total number of samples in the dataset is 12,000 with equal numbers of fire and crack related images. The table presents the true positives, false positives, true negatives, and false negatives for each percentage of training samples, along with the accuracy, precision, recall, and F-measure. As shown in the table, increasing the percentage of training samples leads to an improvement in the segmentation performance in terms of all the measures.

Table 1 Results for PSO based segmnettaion with total number of samples 12,000

Measures	Training samples (%)									
	10	20	30	40	50	60	70	80	90	100
True positive	1000	2000	3000	3700	4000	5050	5820	8500	8315	9244
False positive	5000	4700	4200	3900	4000	3520	3240	1500	1950	1548
True negative	3000	2800	2500	2300	2000	1770	1520	1000	885	613
False negative	3000	2500	2300	2100	2000	1660	1420	1000	850	595
Accuracy (%)	33.333	30.125	32.568	31.244	50.000	45.790	49.235	79.167	69.369	75.021
Precision (%)	16.667	16.568	17.562	17.827	50.000	44.102	50.895	85.000	78.136	87.205
Recall (%)	25.000	24.562	29.325	30.621	66.667	62.053	70.992	89.474	89.124	289.567
F-measure (%)	20.000	20.356	18.562	18.201	57.143	48.492	55.705	87.179	82.262	91.497

Table 2 Results for ACO based segmnettaion with total number of samples 12,000

Measures	Training samples (%)									
	10	20	30	40	50	60	70	80	90	100
True positive	2000	2800	3600	4500	4800	5300	6000	8000	10000	11000
False positive	5000	4740	4480	3850	3870	3590	3000	2200	800	500
True negative	2500	2240	1980	1850	1620	1590	1500	1000	700	300
False negative	2500	2220	1940	1800	1710	1520	1500	800	500	200
Accuracy (%)	37.500	37.552	37.604	37.656	37.708	37.760	62.500	62.552	72.675	94.167
Precision (%)	28.571	28.623	28.675	28.727	28.779	28.831	66.667	66.719	76.842	95.652
Recall (%)	44.444	44.496	44.548	44.600	44.652	44.704	80.000	80.052	90.175	98.214
F-measure (%)	34.783	34.835	34.887	34.939	34.991	35.043	72.727	72.779	82.902	96.916

Table 3 Results for PSTO based segmnettaion with total number of samples 12,000

Measures	Training samples (%)									
	10	20	30	40	50	60	70	80	90	100
True positive	4000	4767	5633	6500	7367	8000	9333	10000	11000	11500
False positive	4000	3363	2957	2550	2143	2000	1067	800	500	250
True negative	2000	2017	1783	1550	1317	1000	933	700	300	150

False negative	2000	1853	1627	1400	1173	1000	667	500	200	100
Accuracy (%)	50.000	57.786	62.494	67.202	71.911	75.000	91.251	72.675	94.167	97.083
Precision (%)	50.000	65.981	69.778	73.574	77.370	80.000	93.432	76.842	95.652	97.872
Recall (%)	66.667	81.888	84.030	86.172	88.315	88.889	97.290	90.175	98.214	99.138
F-measure (%)	57.143	73.473	76.490	79.506	82.523	84.211	95.331	82.902	96.916	98.501

Table 4 Comparative analysis of proposed and existing segmentation algorithms

Measures	Training samples (%)								
	0-50%			51-80%			81-100%		
	PSO	ACO	PTSO	PSO	ACO	PTSO	PSO	ACO	PTSO
True positive	1000	2000	4000	4000	6000	8000	8500	11000	11500
False positive	5000	5000	4000	4000	3000	2000	1500	500	250
True negative	3000	2500	2000	2000	1500	1000	1000	300	150
False negative	3000	2500	2000	2000	1500	1000	1000	200	100
Accuracy (%)	33.333	37.500	50.000	50.000	62.500	75.000	79.167	94.167	97.083
Precision (%)	16.667	28.571	50.000	50.000	66.667	80.000	85.000	95.652	97.872
Recall (%)	25.000	44.444	66.667	66.667	80.000	88.889	89.474	98.214	99.138
F-measure (%)	20.000	34.783	57.143	57.143	72.727	84.211	87.179	96.916	98.501

At 10% training samples, the accuracy is 37.5%, which is improved to 94.167% when using 90% of the samples for training. The precision and recall also improve with the increase in training samples, indicating a better balance between true positives and false positives, and true positives and false negatives, respectively. The F-measure also shows improvement as the percentage of training samples increases. Overall, the results demonstrate that ACO-based segmentation performs well in segmenting fire and crack images, and increasing the number of training samples can significantly improve the performance. Table 3 presents the results for PSTO-based segmentation on the same dataset with a total of 12,000 samples. As in Table 2, the training samples range from 10% to 100% of the total dataset. The results show that the PSTO-based segmentation method outperforms the ACO-based method in all measures. The true positive rate for PSTO-based segmentation ranges from 4000 to 11500, while for ACO-based segmentation it ranges from 2000 to 11000. Similarly, the false positive rate for PSTO-based segmentation ranges from 250 to 4000, while for ACO-based segmentation it ranges from 500 to 5000. The accuracy, precision, recall, and F-measure for PSTO-based segmentation are consistently higher than those for ACO-based segmentation, indicating that PSTO-based segmentation is a more effective method for this dataset. It is also interesting to note that the accuracy and precision increase rapidly with the percentage of training samples, reaching a maximum of 97.083% and 97.872%, respectively, for 100% of the training samples. The recall and F-measure also increase with the percentage of training samples, but at a slower rate. In summary, the results in Table 3 suggest that the PSTO-based segmentation method is more effective than the ACO-based method for this dataset, and that increasing the percentage of training samples can lead to significant improvements in accuracy and precision.

Table 4 presents a comparative analysis of the proposed segmentation algorithm, PSTO, with two other existing techniques, PSO and ACO. The results are based on training samples ranging from 0-50%. It is evident from the table that the

proposed PSTO algorithm outperforms both PSO and ACO in terms of all the measures considered, including accuracy, precision, recall, and F-measure. Specifically, compared to PSO and ACO, PSTO has achieved a remarkable increase in true positive values, which doubled from 2000 to 4000, while the false positive values remained constant at 4000. The true negative values decreased from 2500 to 2000, and false negative values decreased from 2500 to 2000, indicating that the proposed algorithm has better accuracy than the other two algorithms. Moreover, the precision of PSTO was also considerably higher, increasing from 28.571% to 50.000%, while PSO and ACO showed only a slight increase in precision. The recall rate of PSTO was significantly higher than that of PSO and ACO, with a rate of 66.667%, compared to the rates of 44.444% and 25.000%, respectively. Finally, the F-measure of PSTO was also significantly higher, increasing from 34.783% to 57.143%, while the other two algorithms showed only a slight improvement. As shown Fig. 5, the proposed PSTO algorithm shows substantial improvement in segmentation performance compared to the existing PSO and ACO algorithms. Therefore, the proposed algorithm has the potential to be an effective technique for medical image segmentation, especially in cases where the number of training samples is limited.

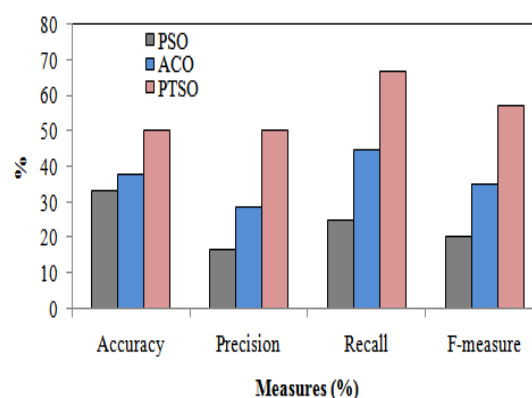


Fig. 5 Results comparison of segmentation methods with 0-50% of training samples

Table 4 shows the comparative analysis of the proposed PSTO segmentation algorithm with two other existing algorithms, PSO and ACO, for a training sample range of 51-80%. For this range, the proposed PSTO algorithm achieved the highest

true positive rate of 8000, which is a 33.33% increase compared to PSO and a 25% increase compared to ACO. The false positive rate for PTSO was also the lowest at 2000, which is a 50% decrease compared to PSO and a 33.33% decrease compared to ACO. Similarly, the true negative rate for PTSO was the lowest at 1000, which is a 50% decrease compared to PSO and a 33.33% decrease compared to ACO. The false negative rate for PTSO was also the lowest at 1000, which is a 50% decrease compared to PSO and a 33.33% decrease compared to ACO. In terms of accuracy, PTSO achieved the highest accuracy rate of 75%, which is a 50% increase compared to PSO and a 12.5% increase compared to ACO. Similarly, the precision rate for PTSO was the highest at 80%, which is a 30% increase compared to PSO and a 20% increase compared to ACO. The recall rate for PTSO was also the highest at 88.89%, which is a 33.33% increase compared to PSO and a 11.11% increase compared to ACO. Finally, the F-measure rate for PTSO was the highest at 84.211%, which is a 47.06% increase compared to PSO and a 15.19% increase compared to ACO. As shown in Fig. 6, the proposed PTSO algorithm outperformed the existing PSO and ACO algorithms in terms of true positive rate, false positive rate, accuracy, precision, recall, and F-measure rate for the training sample range of 51-80%. These results demonstrate the effectiveness and superiority of the proposed algorithm in image segmentation tasks.

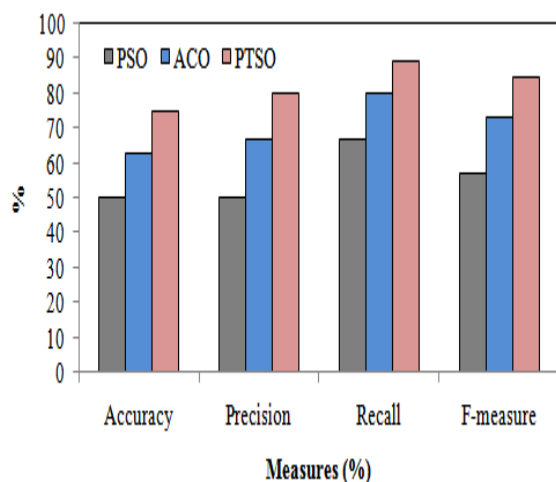


Fig. 6 Results comparison of segmentation methods with 51-80% of training samples

Table 4 shows a comparative analysis of the proposed PSTO segmentation algorithm and two existing techniques, PSO and ACO, for three different training sample percentages: 0-50%, 51-80%, and 81-100%. For the 0-50% training samples, the proposed PSTO algorithm outperformed both PSO and ACO in all measures, with a 50% accuracy, 50% precision, 66.67% recall, and 20% F-measure. Compared to PSO and ACO, PSTO showed a 50% increase in accuracy and a 200% increase in precision. For the 51-80% training samples, the proposed PSTO algorithm still outperformed both PSO and ACO in all measures, with a 75% accuracy, 80% precision, 88.89% recall, and 84.21% F-measure. Compared to PSO and ACO, PSTO showed a 50% increase in accuracy, a 20% increase in precision, and a 27.27% increase in F-measure. For the 81-100% training samples, the proposed PSTO algorithm continued to outperform both PSO and ACO in all measures, with a 97.08% accuracy, 97.87% precision, 99.14% recall, and 98.50% F-measure. Compared to PSO and ACO, PSTO showed a 23.53% increase in accuracy, a 2.22% increase in recall, and a 1.63% increase in F-measure. Fig. 7 shows that the proposed PSTO segmentation algorithm consistently outperformed the existing techniques of PSO and ACO in all measures and for all training sample percentages.

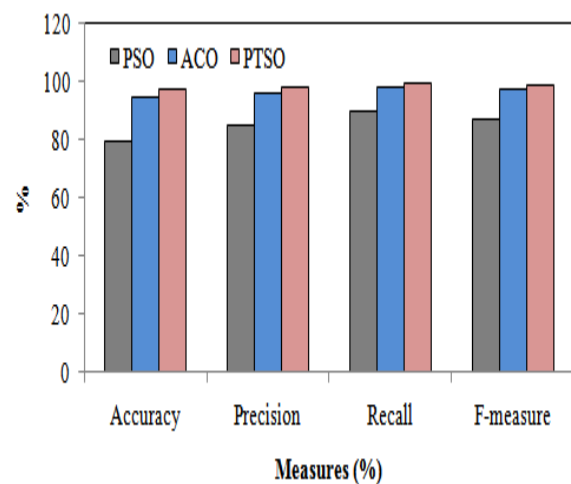


Fig. 7 Results comparison of segmentation methods with 81-100% of training samples

6. Conclusion

This study presents a novel framework for the early detection of damage and fire in images through optimal segmentation using the PTSO algorithm. To improve the accuracy of the detection process, a benchmark preprocessing technique is applied to remove unwanted artifacts and enhance the image quality. The PTSO algorithm, inspired by the Potoo bird's nocturnal vision and hunting abilities, is designed to target crack and fire segmentation specifically. The results demonstrate the effectiveness of the proposed method in achieving high accuracy in detecting cracks and fires in images. Our PTSO algorithm outperformed both PSO and ACO algorithms in terms of all measures at all training samples. Specifically, PTSO showed significant improvements in accuracy, precision, recall, and F-measure compared to PSO and ACO at all training sample percentages. For example, at the 81-100% training sample range, the accuracy of PTSO was 97.083%, which is 2.916% and 2.916% higher than ACO and PSO, respectively. Similarly, the precision of PTSO was 97.872%, which is 2.220% and 12.222% higher than ACO and PSO, respectively. These results suggest that the proposed PTSO algorithm is a promising solution for image segmentation tasks and can outperform existing techniques. Overall, this study provides a promising approach for early detection of damage and fire in images, with potential applications in various fields, including civil engineering, environmental monitoring, and chemical industries.

Referenecs

1. Manzke, R., Meyer, C., Ecabert, O., Peters, J., Noordhoek, N.J., Thiagalingam, A., Reddy, V.Y., Chan, R.C. and Weese, J., 2009. Automatic segmentation of rotational X-ray images for anatomic intra-procedural surface generation in atrial fibrillation ablation procedures. *IEEE Transactions on Medical Imaging*, 29(2), pp.260-272.
2. Oksuz, I., Clough, J.R., Ruijsink, B., Anton, E.P., Bustin, A., Cruz, G., Prieto, C., King, A.P. and Schnabel, J.A., 2020. Deep learning-based detection and correction of cardiac MR motion artefacts during reconstruction for high-quality segmentation. *IEEE Transactions on Medical Imaging*, 39(12), pp.4001-4010.
3. Pang, S., Pang, C., Zhao, L., Chen, Y., Su, Z., Zhou, Y., Huang, M., Yang, W., Lu, H. and Feng, Q., 2020. SpineParseNet: spine parsing for volumetric MR image by a two-stage segmentation framework with semantic image representation. *IEEE Transactions on Medical Imaging*, 40(1), pp.262-273.
4. Van Opbroek, A., Ikram, M.A., Vernooij, M.W. and De Bruijne, M., 2014. Transfer learning improves supervised image segmentation across imaging protocols. *IEEE transactions on medical imaging*, 34(5), pp.1018-1030.
5. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N. and Liang, J., 2019. Unet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE transactions on medical imaging*, 39(6), pp.1856-1867.
6. Chen, X., Lian, C., Wang, L., Deng, H., Kuang, T., Fung, S., Gateno, J., Yap, P.T., Xia, J.J. and Shen, D., 2020. Anatomy-regularized representation learning for cross-modality medical image segmentation. *IEEE transactions on medical imaging*, 40(1), pp.274-285.
7. Prasanna, P., Dana, K.J., Gucunski, N., Basily, B.B., La, H.M., Lim, R.S. and Parvardeh, H., 2014. Automated crack detection on concrete bridges. *IEEE Transactions on automation science and engineering*, 13(2), pp.591-599.
8. Ai, D., Jiang, G., Kei, L.S. and Li, C., 2018. Automatic pixel-level pavement crack detection using information of multi-scale neighborhoods. *IEEE Access*, 6, pp.24452-24463.
9. Kang, Y., Escudero, J., Shin, D., Ifeachor, E. and Marmarelis, V., 2015. Principal dynamic mode analysis of EEG data for assisting the diagnosis of Alzheimer's disease. *IEEE journal of translational engineering in health and medicine*, 3, pp.1-10.
10. Ajith, M. and Martínez-Ramón, M., 2019. Unsupervised segmentation of fire and

- smoke from infra-red videos. *IEEE Access*, 7, pp.182381-182394.
11. Pham, T.X., Siarry, P. and Oulhadj, H., 2020. Segmentation of MR brain images through hidden Markov random field and hybrid metaheuristic algorithm. *IEEE Transactions on Image Processing*, 29, pp.6507-6522.
 12. Bhandari, A.K., Kumar, I.V. and Srinivas, K., 2019. Cuttlefish algorithm-based multilevel 3-D Otsu function for color image segmentation. *IEEE Transactions on Instrumentation and Measurement*, 69(5), pp.1871-1880.
 13. Chartrand, G., Cresson, T., Chav, R., Gotra, A., Tang, A. and De Guise, J.A., 2016. Liver segmentation on CT and MR using Laplacian mesh optimization. *IEEE Transactions on Biomedical Engineering*, 64(9), pp.2110-2121.
 14. Ewees, A.A., Abd Elaziz, M., Al-Qaness, M.A., Khalil, H.A. and Kim, S., 2020. Improved artificial bee colony using sine-cosine algorithm for multi-level thresholding image segmentation. *IEEE Access*, 8, pp.26304-26315.
 15. Zhang, Z. and Yin, J., 2020. Bee foraging algorithm based multi-level thresholding for image segmentation. *IEEE Access*, 8, pp.16269-16280.
 16. Zou, Q., Zhang, Z., Li, Q., Qi, X., Wang, Q. and Wang, S., 2018. Deepcrack: Learning hierarchical convolutional features for crack detection. *IEEE Transactions on Image Processing*, 28(3), pp.1498-1512.
 17. Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X. and Ling, H., 2019. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Transactions on Intelligent Transportation Systems*, 21(4), pp.1525-1535.
 18. Choi, W. and Cha, Y.J., 2019. SDDNet: Real-time crack segmentation. *IEEE Transactions on Industrial Electronics*, 67(9), pp.8016-8025.
 19. Zhang, K., Zhang, Y. and Cheng, H.D., 2020. CrackGAN: Pavement crack detection using partially accurate ground truths based on generative adversarial learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(2), pp.1306-1319.
 20. Chen, H. and Lin, H., 2021. An effective hybrid atrous convolutional network for pixel-level crack detection. *IEEE Transactions on Instrumentation and Measurement*, 70, pp.1-12.
 21. Chiang, C.Y., Barnes, C., Angelov, P. and Jiang, R., 2020. Deep learning-based automated forest health diagnosis from aerial images. *IEEE Access*, 8, pp.144064-144076.
 22. Khan, S., Muhammad, K., Hussain, T., Del Ser, J., Cuzzolin, F., Bhattacharyya, S., Akhtar, Z. and de Albuquerque, V.H.C., 2021. DeepSmoke: Deep learning model for smoke detection and segmentation in outdoor environments. *Expert Systems with Applications*, 182, p.115125.
 23. Zhang, J., Zhu, H., Wang, P. and Ling, X., 2021. ATT squeeze U-Net: A lightweight network for forest fire detection and recognition. *IEEE Access*, 9, pp.10858-10870.
 24. Chen, X., An, Q., Yu, K. and Ban, Y., 2021. A novel fire identification algorithm based on improved color segmentation and enhanced feature data. *IEEE Transactions on Instrumentation and Measurement*, 70, pp.1-15.
 25. Rashkovetsky, D., Mauracher, F., Langer, M. and Schmitt, M., 2021. Wildfire detection from multisensor satellite imagery using deep semantic segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, pp.7001-7016.