International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

GPU Accelerated Code Optimization: Leaf Disease Detection

K. Meghana[1], Rucha Gurav[2], Khushboo Parab[3], Preeti Godabole[4]
[1, 2, 3]Student, Computer Engineering, University of Mumbai, Navi Mumbai, Maharashtra, India-400706
[4]Lecturer, Computer Engineering, University of Mumbai, Navi Mumbai, Maharashtra, India-400706

Abstract— *Modern GPU's highly parallel structure makes manipulating computer graphics and image processing, more efficient and faster than general. Our project is based on Code Optimization which makes use of GPU acceleration. Code optimization can be achieved by parallel execution using GPU. GPU-accelerated computing offloads compute-intensive portions of the application to the GPU, while the rest of the code still runs on the CPU. We propose and experimentally evaluate a solution for faster detection and classification of plant leaf diseases using GPUs. Our project solution is an improvement to the existing system as it gives more faster and accurate solutions. To demonstrate the effects of optimizations over time, we at first experimented on CPU (using Matlab) and then we will experiment it on GPUs using OpenCV to understand the improvement in performance. Identification of plant diseases is beneficial in monitoring large fields of crops, and automatically detect the symptoms of diseases at early states only. Therefore, for great realistic significance one need to look for fast, automatic, less expensive methods to detect plant diseases. By making use of OpenCV libraries in mobile's GPU through an Application, it helps to decrease the execution time of detection of disease level and also provides prevention method (name of disease and pesticides to be used) in English and Marathi and the percentage of the infected area. The language flexibility is provided for better understanding of farmers. The dataset of leaf disease has taken from Kaggle site. The mobile application identifies four diseases: Rust, Tar spot, Linden Leaf and Phyllosticta.*

Keywords— *Parallel processing: Plant leaves diseases: Preventions: Symptoms.*

I. INTRODUCTION

A graphics processing unit (GPU) is a specialized electronic circuit which is designed to manipulate and alter memory to accelerate the creation of images in a frame buffer for intended output to a display device. In embedded systems, mobile phones, personal computers, and game consoles GPUs are used .Modern GPUs highly parallel structure makes manipulating computer graphics and image processing, more efficient than general.

Image processing technology in the agricultural research has made significant development. The goal is to perform automatic detection and classification of leaf diseases using image processing technique. Plant disease is a crucial issue which causes noteworthy reduction in the quality, growth and quantity of plant production. In India 60 percent of the people, depend on agriculture for their livelihood. Hence it become very important to detect diseases in leaves or other plants if occur. Recognition and classification of diseases in plants using digital image method is extremely effective in providing symptoms of characteristic diseases at its early stages. It is very essential to identify leaf disease at an early stage and suggest solution so to minimize the loss [2].

The plant diseases are a major threat to losses of agricultural production. Plant disease is an important parameter to detect the disease level and thus can be used to classify them and prevent them by providing appropriate treatments and spraying correct volume of pesticides on plant. The rapid, accurate diagnosis of disease severity will help to reduce yield losses. Traditionally, plant disease severity was used to be detected using help of trained experts which consumed a lot of time in visual inspection. The expensive cost and low efficiency of human disease assessment hinders the rapid development of modern agriculture [5]. With the population of digital cameras and the advances in computer vision, the automated disease diagnosis models are highly demanded by precision agriculture, high- throughput plant phenotype and so forth. The accurate Disease detection and classification of the plant leaf image is very important for the successful cultivation of cropping and this can be done using image processing. The optimum results can be obtained using new technologies with very less computational efforts. The results obtained helps farmers in making effective decision, so it provides support system in diagnosis phase. This scheme even detects prevention methods for the diseases [1].

In order to help the people in an effective way, execution time for detecting the diseases in leaves was carried out using GPU along with the CPU. A part of code was executed in Matlab and Android Studio differently. A Leaf detection application was made in Android Studio named as "Plant Leaf Disease Detection" which makes use of Open CV libraries and works on mobile's GPU. The images are given as input to the application and processes on it to show the prevention methods.

The processing of images passes across many stages including input, conversion of images into a specific format, extraction of features and then finally classification of diseases. It can identify four different types of diseases: *Rust, Tar spot, Linden Leaf and Phyllosticta.* A comparison is made between the execution time of both the CPU processing Matlab and the GPU processing application. Thus, the results proved that working with GPU provides much faster output than the CPU.

II. METHODOLOGY

GPU's allows the work to process parallely in an efficient way as compared to CPU. An application was designed which makes use of its mobile's GPU for parallel processing.

K. Meghana, Rucha Gurav, Khushboo Parab, and Preeti Godabole, "GPU accelerated code optimization: Leaf disease detection," *International Research Journal of Advanced Engineering and Science*, Volume 3, Issue 2, pp. 165-168, 2018.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

OpenCV libraries are also included for easy functioning. The images are given as input to the application and processes on it to show the prevention methods. Images can either be captured or can be browsed for input in .jpg format. After processing it identifies the infected areas and accordingly inform the disease name, percentage of infected area, execution time and treatment description. Fig 1 shows the proposed system for the project.

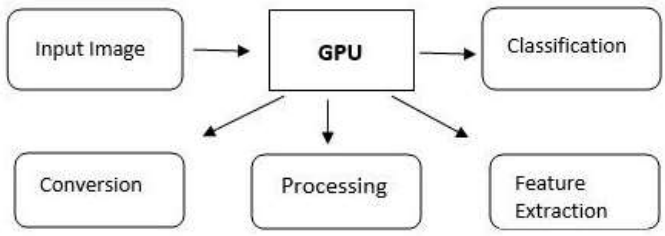

Fig. 1. Proposed system.

Fig. 2 Describes the different stages for detecting the disease [4].

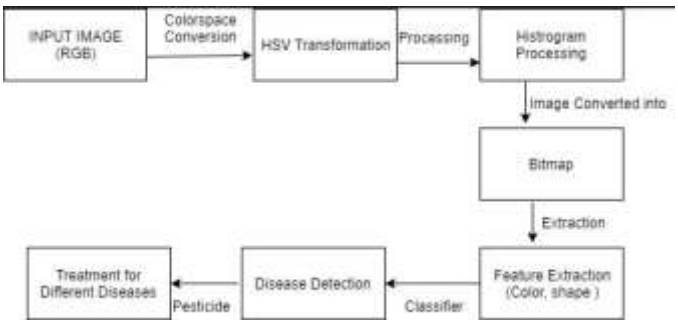

Fig. 2. Flow diagram.

1. Collection of Dataset

Collection of dataset is the first step of the whole processing. The infected leaf images are captured through the camera, this image is in RGB form, color transformation structure for the leaf image is created. structure is applied [1].

2. Conversion

Input image should be analyzed for performing following steps: RGB to HSV Resizing of image according to need Image Colorspace Conversion.

3. Processing

It consists of two steps. Firstly, it splits the image into its R, G and B planes using the function split .Calculate the Histogram of each 1-channel plane by calling the function calcHist. An image histogram is a chart that represents the distribution (frequency of occurrence) of pixels within an image. Detecting objects and borderlines in images are also an important part of plant Processing. Secondly, image gets converted into bitmap form for final result.

4. Feature Extraction

In feature extraction method, various attributes of the segmented image are extracted. Features like colour, shape, and texture are extracted and are compared with the original plant leaf image.

5. Classification of Plant Diseases

Classification technique is used for detecting the type of leaf disease. Based on classification leaves are mainly affected with fungal, bacterial and viral.

Main diseases detected on crops
RUST- Found in wheat, bajra coffee.
Symptoms: Begins as small circular to oval yellow spots. As the disease progresses, the spots develop into orange-colored pustules.

Fig. 3. Rust affected leaf.

TAR SPOT- Found in corn, maple trees.
Symptoms: Looks like a spot of tar, developing black oval or circular lesions on leaf.

Fig. 4. Tar spot affected leaf.

LINDEN LEAF- Brownish circular spots on the leaves. The brown spots will have a darker ring around the outer border and can cause premature leaf drop.

Fig. 5. Linden affected leaf.

K. Meghana, Rucha Gurav, Khushboo Parab, and Preeti Godabole, "GPU accelerated code optimization: Leaf disease detection," *International Research Journal of Advanced Engineering and Science*, Volume 3, Issue 2, pp. 165-168, 2018.

PHYLLOSTICTA- Has spot outlined in black, surrounding a tan or gray area of dead tissue, with black fruiting body at the center.

Fig. 6. Phyllosticta affected leaf.

III. IMPLEMENTATION

This proposed system will first take an input image, which is in RGB form. It detects the infected part of the disease. Data set contains images, which need to be processed and through feature extraction diseases are detected.

- Input of the image through App's camera or by browsing.

Fig. 7. Capturing of image.

- It shows Disease Measure, Disease Severity and Disease Name.

Fig. 8. Identification of disease using App in English.

- Language Flexibility (in Marathi)

Fig. 9. Identification of disease using App in Marathi.

- Treatment description is produced according to the Disease detected.

Fig. 10. Description of the treatment.

Time calculation
a. Matlab:

$$Total\ Time = \sum_{i=0}^{t\ tasks} time\ of\ task\ i \qquad \text{(i)}$$

$$\text{time of task} = \frac{work\ of\ task}{\text{rate of work of task}} \qquad \text{(ii)}$$

Using equation (i), total time is calculated as the sum of the time of all sub tasks.
Using equation (ii), time of each task can be calculated as work by the rate of work of the task.

b. OpenCV:
double difMs = d2.getTime()-
today.getTime(); double difSec = difMs /
1000; double min = difSec / 60;
double sec = difSec
double difMs - It is the difference between the ending time of the process and starting time (capturing of the image).

K. Meghana, Rucha Gurav, Khushboo Parab, and Preeti Godabole, "GPU accelerated code optimization: Leaf disease detection," *International Research Journal of Advanced Engineering and Science*, Volume 3, Issue 2, pp. 165-168, 2018.

IV. RESULTS

```
Trial>> tic;
Affected_Area = (A1/A2);
if Affected_Area < 0.1
  Affected_Area = Affected_Area+0.15;
end
sprintf('Affected Area is: %g%%',(Affected_Area*100))
toc;
ans =
  'Affected Area is: 52.2305%'

Elapsed time is 0.005371 seconds.
Trial>> toc;
Elapsed time is 0.0711942 seconds.
Trial>>
```

Fig. 11. Matlab time calculation.

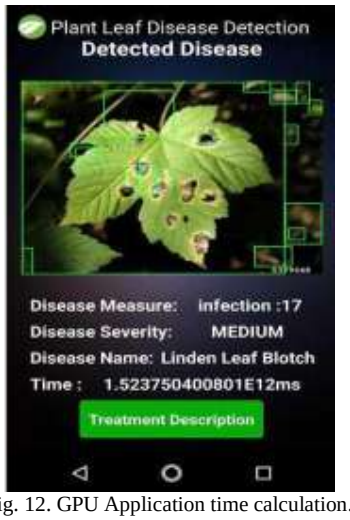

Fig. 12. GPU Application time calculation.

Improvement:

$$\frac{4.3 - 1.52}{4.3} \times 100$$

$$\frac{2.78}{4.3} \times 100$$

$$= 64\ \%$$

Fig. 13. Improvement calculation.

Execution time of Matlab = 0.0711 seconds
= 4.3 milliseconds
Execution time of Application= 1.523 milliseconds.

Execution time was calculated for both Matlab and Application. On comparing the timing, there was an improvement of 64% compared to Matlab. Thus, GPU processing application runs more faster than CPU processing Matlab.

V. CONCLUSION

In our paper, we would like to conclude that this is an efficient and accurate technique for detection of plant diseases. The output produced from the app includes infected percentage area, time execution, and pesticide which needs to be sprayed on it. In this proposed system few species of plants were tested on 3 types of leaves. The result which obtained using GPU (Disease Detection Application) was much lesser than the CPU time execution (Matlab time execution).It gives us an improvement of 80%. It provides the prevention method in two languages (English and Marathi) according to the disease identified. This results obtained helps farmer in making effective decisions at early stages in order to protect them from getting rotten. This scheme computes very fast and can be easily handled by anyone. The Profile GPU Rendering tool displays, as a scrolling histogram, a visual representation of how much time it takes to render the frames of a UI window [6].

VI. ACKNOWLEDGMENT

We have a great pleasure to express my gratitude to all those who have contributed and motivated during my project work. We would like to express deepest gratitude to our Principal Dr. Vikram Patil. for providing us this opportunity. We would like to thank Dr. Rizwana Shaikh, Head of the Department of Computer Engineering for all the guidance and support given to us. We would also like to thank Prof. Preeti Godabole, our Project Guide and Prof. Prachi Shahane, our Project in charge for constant guidance support. We would also like to thank all the faculty members non-teaching staff who have helped us directly or indirectly to complete our project report. With this we would also like to thank our family and friends for their support. The successful completion of our project within the stimulated time frame is the result of collective efforts of our group.

REFERENCES

[1] S. Megha and C. R. Niveditha, "Image processing system for plant disease identification by using FCM-Clustering technique," *International Journal of Advance Research, Ideas and Innovations in Technology*, vol. 3, issue 2, pp. 445-449, 2017.

[2] P. R. Rothe and R. V. Kshirsagar, "Cotton leaf disease identification using pattern recognition techniques," *International Conference on Pervasive Computing (ICPC)*, Pune, 2015, pp. 1-6, 2015.

[3] Sonal P Patel and Mr. Arun Kumar Dewangan, "A comparative study on various plant leaf diseases detection and classification," *International Journal of Scientific Research Engineering & Technology*, vol. 6, issue 3, pp. 220-224, 2017.

[4] Priti Badar, Suchitra.C, "Disease detection of pomegranate plant using image processing," *International Journal of Computer Science and Mobile Computing*, vol. 5, issue 5, pp. 328-334, 2016.

[5] Y. Sanjana and Ashwath Sivasamy, "Plant disease detection using image processing techniques," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 4, issue 6, pp. 295-334, 2015.