AUTO-ENCODER BASED CONVOLUTIONAL NEURAL NETWORK (AECNN) FOR MULTISPECIES FRUIT FLOWER DETECTION

Dr. M. Nester Jeyakumar & Dr. Jasmine Samraj

Assistant Professor, Department of Computer Science, Loyola College, Chennai, TN, India
Associate Professor, Department of Computer Science,
Quaid-E-Millath Government College for Women (A), Chennai, TN, India
nesterjk@gmail.com, jasminesamraj@gmail.com

ABSTRACT: On trees, process of counting and detecting fruits count defines the task of crop estimation. At various locations, manual fruits and vegetables counting is performed in recent days. Excessive amount of labor requirement and consumption of more time are the major drawbacks of this manual counting. Bloom intensity which corresponds to flowers count present in orchard guides critical crop management decisions in the production of fruits. Even with high importance, human visual inspection is used for estimating bloom intensity. Hand engineered methods forms base for the automated computer vision system used for identifying fruits. Under specific conditions only, their performance is defined and it is a limited one. Major drawbacks of existing methods are, fruit flower images are selected without the reduction of dimensionality, in predictions, objects orientation and positions are not encoded by CNN.For detection of fruit flower, Auto-Encoder based Convolutional Neural Network (AECNN) and Animal Migration Optimization (AMO) algorithms are proposed for solving these issues. There are three main stages in this work. They are, training of network for deep Fully Convolutional Network (FCN), pre-processing, reduction of dimension and segmentation. On commercial GPU, high resolution image evaluation procedure are defined with deep FCN. GPU memory space is needed for fully convolutional computations and based on image resolution, there will be an exponential increase. From image, noises are reduced or removed using an Additive white Gaussian Noise (AWGN) technique which is introduced next. At last, small patches are formed by splitting high resolution image, AMO is used for reducing dimensionality, fine-tuned AECNN is used for evaluating every patch and final segmentation is performed by applying refinement algorithm. Three datasets of peach and apple flowers are used in experimentation. Metrics like Intersection-over-Union (IoU), F-Score(F_1), Recall(R), Precision(P) are used for measuring the results.

KEYWORDS: Auto-Encoder based Convolutional Neural Network (AECNN), Additive white Gaussian Noise (AWGN), Fully Convolutional Network (FCN), Animal Migration Optimization (AMO).

1. INTRODUCTION

The harvest of the executives' framework ought to be consolidated with mechanized yield expectation innovation to create great outcomes and get greater efficiency with less sums. The expectation should be possible by utilizing yield gauges which gives significant data. The yield gauges are additionally used to make remedy maps that can be used by tree escalated applications. The profitability and proficiency of the plant science can be improved with the assistance of innovative advancements in detecting, mechanical technology and registering calculations [1].

These days, in organic product horticultural creation, the greater part of the cultivating works depend on the physical work of natural product grower. An extraordinary amount of straightforward and monotonous works not just devours time and vitality and builds creation costs yet additionally carries more vulnerabilities to farming creation. Organic product ranchers may commit errors in the judgment of cultivating because of the absence of information and experience. An inappropriate judgment will straightforwardly prompt an inappropriate execution of cultivating, which could seriously affect crop yield [2].

During early growing season, in orchard, presented flowers count defines bloom intensity. In thinning and pruning process, guidance given by information of bloom intensity and climate are very crucial. Fruit taste, coloration, size and load are directly affected by this. Packing houses may be benefitted by bloom intensity's accurate estimation. Postharvest handling and storage processes optimization are contributed by early crop-load estimation.

In orchards, for estimating bloom intensity, visual inspection is used as a dominant method, but it is labor intensive as well as time consuming method which is prone to errors [3]. In orchard, spatial variability information are not provided by it, but it knows well about precision agriculture practices benefits [4]. So, in order to solve this problem, various automatic computer vision systems have been implemented, but they relies on hand-engineered features [5]. So, only in relatively controlled environments, their performance is acceptable.

In computer vision, primary goal in image understanding. In various applications like surveillance, security, autonomous navigation, object tracking, it is crucial one. In image, computation of objects is a highly important aspect in image understanding [6]. Various image processing methods and sensor types are used in machine vision system for introducing highly accurate and reduced labor intensive methods to estimate bloom intensity [7]. A CNN extracted feature based apple flower detection method is proposed which is inspired by effectiveness of Convolutional Neural Network (CNN) in various computer vision tasks.

In agriculture applications like plant identification from leaf vein patterns, crop classification and quantification of fruit, CNN architectures are adapted in recent works [8]. In those works, apple flowers are detected using combination of classification network and super-pixel-based region proposals. Intrinsic to super-pixels segmentation accuracy and architecture of network are the limitations of those methods.

Different segmentation algorithms are used for segmenting image into background and foreground regions. For further processing, eliminate the background regions. Pixel intensities are used for performing segmentation. In order to differentiate from foreground, color of background should be selected accordingly. For operations, any background color can be chosen by user, but results are effected accordingly. So, for enhancing efficiency, background color can be selected using automatic process [9].

For detection of fruit flower, Auto-Encoder based Convolutional Neural Network (AECNN) and Animal Migration Optimization (AMO) algorithms are proposed. There are three main stages in this work. They are, training of network for deep Fully Convolutional Network (FCN), pre-processing, reduction of dimension and segmentation. On commercial GPU, high resolution image evaluation procedure are defined with deep FCN. GPU memory space is needed for fully convolutional computations and based on image resolution, there will be an exponential increase. From image, noises are reduced or removed using an Additive white Gaussian Noise (AWGN) technique which is introduced next. At last, small patches are formed by splitting high resolution image, AMO is used for reducing dimensionality, fine-tuned AECNN is used for evaluating every patch and final segmentation is performed by applying refinement algorithm.

2. LITERATURE REVIEW

Dias et al [10]detected apple flowers using a deep learning techniques based novel Convolutional Neural Network (CNN) algorithm. For computer vision applications, state of the art are represented using this technique. Flower sensitiveness is given to Pre-trained CNN by fine-tuning it. In contrast to existing techniques, morphological and color information are combined effectively by proposed CNN using extracted hierarchical features, which leads to performance enhancement to all considered cases. Existing techniques are color analysis based and their application to conditions with illumination or occlusion level change are limited. Four various datasets namely, Peach, AppleC, AppleB, AppleA are used in experimentation. Even with various species of flowers and illumination, better performance is exhibited by proposed CNN-based model with accurate identification. Around 80% of precision and recall values are produced for new datasets without training.

Lin and Chen [11] implemented a reliable, fast as well as accurate strawberry flower detection system called Faster Region-based Convolutional Neural Network (Faster R-CNN), which automatically estimate harvesting and strawberry flower yield. In outdoor fields, to enhance strawberry flowers detection accuracy, developed a state-of-the-art deep-level object detection framework of R-CNN. A set of 400 strawberry flower images is used for training the network and 100 strawberry flower images are used for testing it. On multiple scales, three object detection methods based on region including Fast R-CNN, R-CNN, Faster R-CNN are implemented for feature identification and for representing instances of strawberry flower. Better performance is achieved using Faster R-CNN method as shown in experimentation when compared with Fast R-CNN and R-CNN and consumes less time. With overlapping of strawberry flowers or under shadow or with occlusion by foliage, better performance can be shown by Faster RCNN. For current manual yield estimate counting of flowers or fruits, viable solution can be provided using this automatic yield estimation. Manual estimation requires more time and it is an expensive one and for big fields, it cannot be applied.

Koirala et al [12]estimated mango fruits flowering using a two-stage deep learning framework. With a single and a two-stage deep learning framework (YOLO and R^2CNN), undertaken the panicles classification into three developmental stages via rotated or upright bounding boxes. Models R^2CNN-upright, R^2CNN(-rotated), YOLOv3-rotated, MangoYOLO-rotated, MangoYOLO(-upright) are used in image set validation and counting total panicle. The R^2 for machine vision is used for counting panicles in tree using various camera and involving various cultivar and orchard images. Well generalization is exhibited by this model and there is no consistency in using rotated over upright bounding boxes. Better results in counting total panicle can be achieved using YOLOv3-rotated model and classification of panicle stage can be done effectively using R^2CNN-upright model. For demonstrating practical

applications, for 994 trees orchard, weekly panicle count is made with the application of peak detection routine to document multiple flowering events.

Horton et al [13] detected patch blossoms on trees by implementing an image processing algorithm. From southwestern part of Idaho, from commercial and experimental peach orchards, acquired peach(Prunus persica) trees aerial images via off-the-shelf Unmanned Aerial System (UAS) which is equipped with multispectral camera (near-infrared, green, blue). Three bands contrast stretching is included in this image enhancement algorithm for improving patch blossoms detection using image and thresholding segmentation technique. For orchard management, it can be used as a better monitoring tools with 84.3% average detection rate.

Hočevar et al [14] detected apple fruit flower using an automated tree Flower Clusters (FC) measurement system. In apple fruit production, for high quantity and quality, it is needed to have Tree-specific management practice related to individual tree based on its physiological condition. In a high density apple orchard, for estimating individual trees FC count, Hue, Saturation, Luminance (HSL) images analysis based apple flowering abundance detection is used. Still camera is used for acquiring images in day time and industrial color camera is used in night time. The HSL thresholding is included in FC estimation algorithm with parameter optimization. Hypothetical spraying was done by on/off criterion>100 FC per tree, if industrial camera is used to acquire image acquisition in daytime and identified around 10% incorrect execution. Industrial camera or still camera is used for producing better performance in FC counting.

Dias et al [15] identified flowers using an automated method. This method is robust against uncontrolled environments and various flower species can be identified. End-to-end residual CNN forms the base for this proposed method and in semantic segmentation, state-of-the-art is represented by it. Apple flower images single dataset is used for fine-tuning this network for enhancing sensitivity of this method to flowers. In order distinguish individual flower instances in a better manner, refinement method is employed as coarse segmentations are produced by CNN. Broad applicability and robustness of proposed method is demonstrated using experimentation on pear flowers, peach and apple images which are acquired in various conditions. It does not require ant dataset specific training or pre-processing.

Sa et al [16]detected fruits using a Faster Region-based CNN (Faster R-CNN) technique. For fruit detection task, through transfer learning, this model is adopted and it uses two modalities for obtaining images. They are Near-Infrared (NIR) and Colour (RGB). In order to combine multi-modal (RGB and NIR) information, late and early fusion techniques are explored. For new fruits, this technique can be deployed in a faster manner and produces better accurate results. Bounding box annotation is needed in this and it does not require pixel-level annotation. Annotating bounding boxes is an order of magnitude quicker for performing. Seven fruits are detected using this model and it is retained. For annotating and training new model, four hours is needed by this entire process.

Chen et al [26] predicted yield with minimum time cost and labor using Region-based Convolutional Neural Network (R-CNN) automatic strawberry flower detection system. In this system, two variety near-ground images namely, Radiance and Sensation are captured using a small Unmanned Aerial Vehicle (UAV) developed by DJI Technology Co., Ltd., Shenzhen, China which is equipped with an Red, Green, Blue (RGB) camera at two various heights of 2 and 3 m with orthoimages of a $402\ m^2$ strawberry field. The Pix4D software is used to process orthoimages automatically. For detection of deep learning, it is split into sequential pieces. For counting and detecting immature strawberries, matures strawberries and flowers count, a state-of-the-art deep neural network model called faster Region-based Convolutional Neural Network (R-CNN) is adapted in this system. At 2 m height, for all detected objects mean Average Precision (mAP) is 0.83and at 3 m height, it is about 0.72. Strawberry flowers accurate count can be computed using this system and using this count, future yield can be forecasted as well as distribution maps can be constructed which tends to assist the farmers for observing strawberry fields growth cycle.

3. PROPOSED METHODOLOGY

This work Figure 1 proposes an Auto-Encoder based Convolutional Neural Network (AECNN) algorithm to detect fruit flower. There are three main stages in this work. They are, training of network for deep Fully Convolutional Network (FCN), pre-processing, reduction of dimension and segmentation. On commercial GPU, high resolution image evaluation procedure are defined with deep FCN. GPU memory space is needed for fully convolutional computations and based on image resolution, there will be an exponential increase. From image, noises are reduced or removed using an Additive white Gaussian Noise (AWGN) technique which is introduced next. At last, small patches are formed by splitting high resolution image, AMO is used for reducing dimensionality,

fine-tuned AECNN is used for evaluating every patch and final segmentation is performed by applying refinement algorithm.

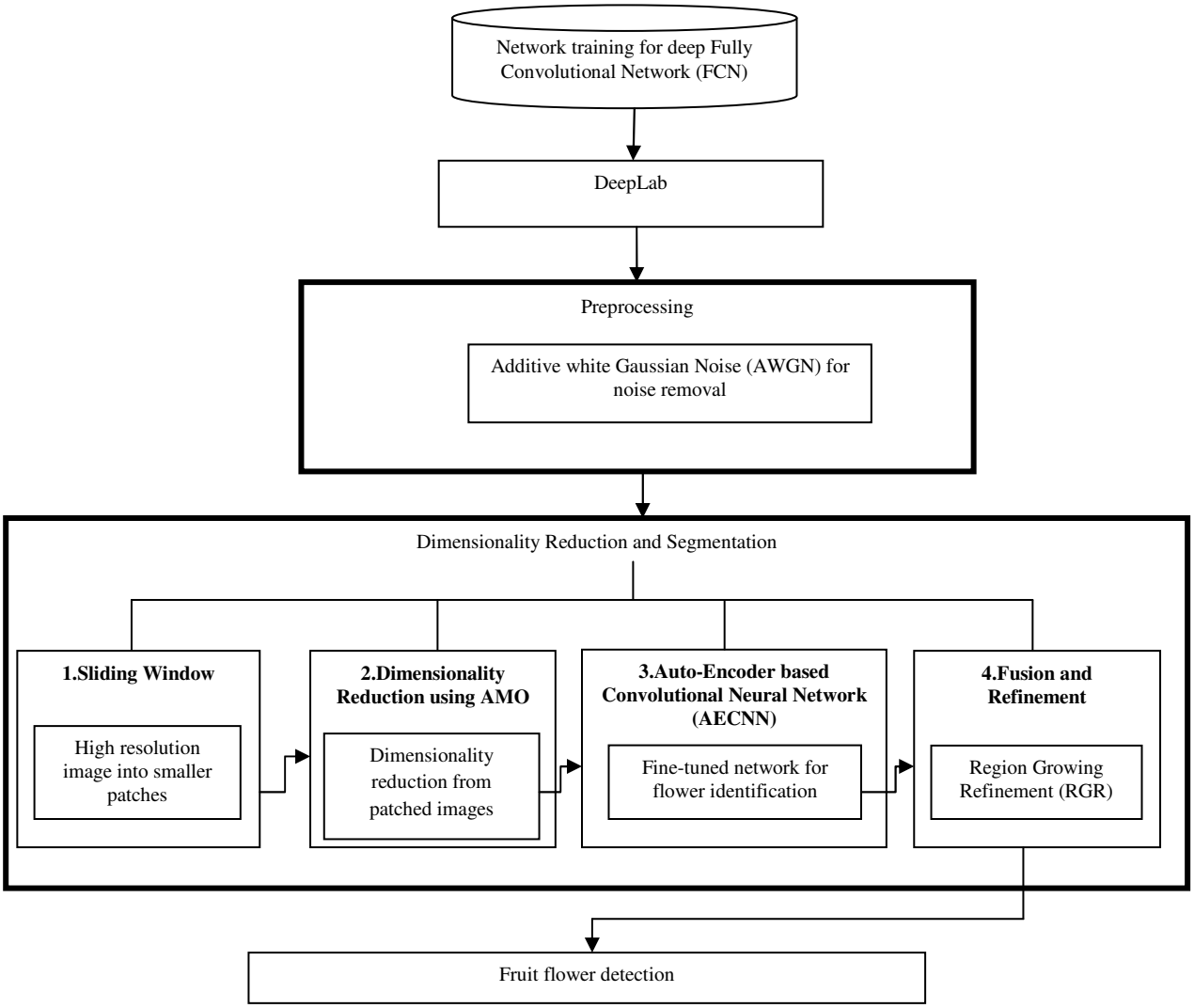


Figure 1: PROPOSED ARCHITECTURE OF FRUIT FLOWER DETECTION

A. Network training for deep Fully Convolutional Network (FCN)

For segmentation of binary flower, this work used DeepLab. In this fine-tuning and network surgery procedures are performed. In trees, undesired branch pruning corresponds to surgery procedure. For interested class segmentation, responsible branches with connections and weights are preserved out of all original branches. Segmentation of fruit flower branch with sigmoid classification unit are used for preserving architecture.

Unbounded the transferred flower branch generated score and saturation of sigmoid happens in an easy manner, without model's original softmax layer induced normalization. Model tuning with two-branches is adopted for alleviating the difficulties in learning which are produced by poor initialization. This is adopted based on the fact that, foreground branch generated predictions can be normalized properly by allowing the network to learn representation of background via second branch. For flower segmentation, major source for misclassification is nearby leaves as indicated in experimentation.

With the application of pre-trained model to training dataset, highest activations are presented for class leaf predictions. Because of this, for initializing two-branch flower segmentation network, branch together with one associated with flowers are adopted. Training set which is shown in Table 1 is used for fine-tuning adapted architecture. There are 100 apple tree images in training set. For 10,000 iterations, this procedure is performed for

this experimentation using Caffe framework [17], with 10^{-4} as initial learning rate and it has $10^{-4} \times \left(1 - \frac{i}{10000}\right)^{0.9}$,polynomial decay. Where, iteration number is represented as i. Procedure of model pre-training is used in model fine-tuning, where every training portrait is evaluated at $(0.5, 0.75, 1: 0, 1.25, 1.5)$ times its original resolution.

TABLE 1: DATASETS SPECIFICATIONS

Dataset	No. images	Weather	Background panel	Camera model	Resolution	Camera support
AppleA	100 (train)+30 (val)	Sunny	No	Canon EOS 60D	5184×3456	Hand-held
AppleB	18	Sunny	Yes	GoPro HERO5	2704×1520	Utility vehicle
Peach	24	Overcast	No	GoPro HERO5	2704×1520	Hand-held

The 100 training images are split into 321×321 pixels portraits, which leads to 52,644 training portraits in total after augmentation using original network parameterization.

B. Preprocessing

In images, original pixel values are modified to various intensity values by noises. From an image, reduction or removal of noises are done using the process called noise removal algorithms. In information theory, basic noise model used is Additive white Gaussian Noise (AWGN) for mimicking various random processes effect in nature [18]. Specific characteristics are donated by modifiers,

- **Additive:** Noises intrinsic to information system are added with this kindof noises.
- **White:** For information system, across the frequency band, it has uniform power distribution. Its behavior is looks like white color noise, where in visible spectrum, at all frequencies, uniform emission is produced by color white noise.
- **Gaussian** : In time domain, it exhibits a normal distribution with zero average time domain value.

A type of statistical noise is a Gaussian noise with Probability Density Function (PDF) similar to Probability Density Function (PDF) of normal distribution. It is also called as Gaussian distribution. In image, every pixels original value is changed by Gaussian noise with small variation. Noise's normal distribution is shown by plot called histogram, it is a plot between pixel value's distortion amount to their frequency of occurrence. When compared with other noise model, Gaussian noise model is mostly preferred, as various noises sum corresponds to a Gaussian distribution as stated in central limit theorem.

From digital images, removal of Gaussian noise is done by image pixel smoothening. In image, presented noise intensity can be reduced using this method. Noises are produced during acquisition and it is not desirable one and at every point, high-quality images pixel value blurring at edges [19]. Gaussian random variable's PDF is expressed as,

$$P(x) = \frac{1}{\left(\sigma\sqrt{2\pi}\right) * \frac{e^{(x-\mu)^2}}{2\sigma^{2-\infty<0<\infty}}}$$

Where, in image, Gaussian distribution is represented as P(x), mean is represented as μ and standard deviation is represented as σ.

An image composed of pixels or also termed as picture element is called as digital image. Every pixel is having a finite discrete quantity, which represents its gray level or intensity and it is an output of two-dimensional functions given with spatial coordinates x on x axis and y on y axis as input. It is classified as raster or vector type based on its fixed resolution.

Spatial filter can be used for reducing Gaussian noise. But high frequencies are blocked by this filter. Because of that, in image smoothing, fine-scaled image details and edges may be blurred as a result of it. Gaussian smoothing, median filtering, mean or average filtering are the conventional spatial filtering methods used for removal of noise.

Mean Filtering

A simple sliding window spatial filter where, average value of all the window pixel is used for replacing center pixel value is termed as mean filter. Square shape of kernel or window is used in general case, but it may take any kind of window. From an image, short tailed noise like Gaussian type and uniform noise can be removed using arithmetic mean filter operation. Because of this image may be blurred. Within a local image region, all pixels average is used for defining arithmetic mean filter [20].

In a rectangular subimage window with $m \times n$ size, assume coordinate set is represented as S_{xy}, which is centered as point (x, y). In an area defined by S_{xy}, corrupted image $g(x, y)$'s average value is computed using arithmetic mean filtering process. At any point (x, y), restored image value will equals arithmetic mean of pixels in the region shown by S.

$$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

Median Filtering

A nonlinear signal processing method is median filtering which is statistic based. Median value of neighborhood is used for replacing digital sequence or images noisy value. Based on gray levels, mask pixels are ranked. Noisy value is replaced by storing group's median value. $g(x, y)$ $\text{med}\{ f(x - i, y - j), i, j \in W\}$ is the output of median filtering, where, original image is represented as $f(x, y)$, output image is represented as $g(x, y)$, two-dimensional mask is represented as W, size of the mask is $n \times n$(with n as an odd value) like $3 \times 3, 5 \times 5$. Cross, circular, square, linear shaped mask can be used [21]. For image having random noise, mathematical analysis of median filtering is a complex one because of its non-linear nature. Under normal distribution, image having zero mean, median filters noise variance can be approximated as,

$$\sigma_{med}^2 = \frac{1}{4nf^2(n)} \approx \frac{\sigma_i^2}{n + \frac{\pi}{2} - 1} \cdot \frac{\pi}{2}$$

Where, input noise power is represented as σ_i^2, median filtering mask size is represented as n, noise density function is represented as $f(\bar{n})$. Average filtering noise variance is represented as,

$$\sigma_0^2 = \frac{1}{n} \sigma_i^2$$

In reduction of random noise, better performance is exhibited by median filtering when compared with average filtering performance. But less effective performance is exhibited on impulse noise of narrow pulse with less than n/2 pulse width. If average filtering algorithm is combined with median filtering algorithm, then its performance can be enhanced and based on noise density, mask size is varied adaptively [22].

Gaussian Smoothing

Blurring produced on image due to Gaussian function is termed as Gaussian blur or it is also termed as Gaussian smoothing. In Graphics software, it is a most commonly used effect for reducing image noise. This blurring method's visual effect is a smooth blur. Image viewing through translucent screen resembles it and it is different from out-of-focus lens produced bokeh effect or object shadow in usual illumination.

In computer vision algorithms, for pre-processing stage, Gaussian smoothening is used. At various scales, image structures are enhanced using this. Gaussian's Fourier transform will produce another Gaussian. High frequency components of images are reduced by applying this Gaussian blur. It is a low pass filter [23]. Image blurring filter type corresponds to Gaussian blur, where transformation is computed using Gaussian function, which is applied to every image pixel. In statistics, normal distribution is expressed by it.

The Gaussian function expression in one dimension is expressed as,

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Product of Gaussian function produces expression in two dimension, with one function in every dimension as,

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Where, in horizontal axis, distance from origin is represented as x, in vertical axis, distance from origin is represented as y and Gaussian distributions standard deviation is represented as σ. Concentric circle contours surface having Gaussian distribution from center point is produced by applying this expression in two dimensions. Convolution matrix are constructed using this distribution values and original image is applied with this.

Right side figure visualizes the process of convolution. Weighted average of neighborhood pixel is assigned as a new value of every pixel. Large weight is assigned to original value of pixel and based on the distance from original pixel, small weights are assigned to neighboring pixels.

C. Dimensionality Reduction and Segmentation

There are four major steps in segmentation of fruit flower. They are, formation of smaller patches by dividing high resolution image in sliding window procedure, AMO based dimensionality reduction, fine-tuned AECNN based evaluation of every patch, computation of final segmentation mask by applying refinement algorithm on computed scoremaps.

1. Sliding Window

Resolution of images in dataset ranges from 2704×1520 to 5184×3456 pixels. Resampling artifacts are avoided using sliding window technique. Formed a set P of n portraits $p^{(i)} \in P$ by spitting every input image. Size of every portrait is 321×321 pixels i.e. $p^{(i)} \in R^{r \times r}$ with r = 321. Artificial boundaries are introduced by cropping of non-overlapping portraits from original image, quality of detection is compromised by this. So, for this, every portrait made overlapped in this technique. Every immediate neighbor area percentage is represented as s. In this experimentation, value of s is 10%. Discarded the results respective to overlapping pixels, if fusion of scoremaps happens. For subsequent portrait pair, this process is illustrated in figure 2. For every portrait, obtained score is depicted as heatmap. Lower scores are represented with blue and red is used for representing high scores.

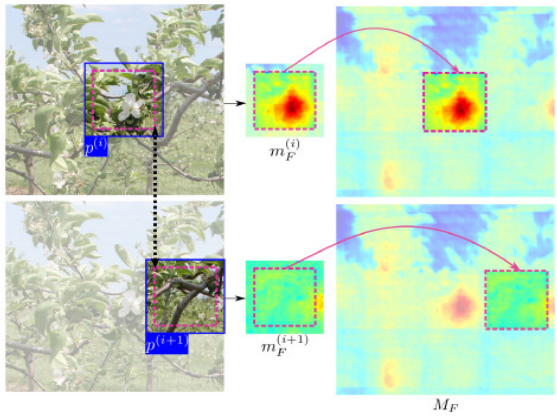


FIGURE 2: BEST VIEWED IN COLOR

2. Dimensionality Reduction using AMO

Animal migration behavior is inspired in a new heuristic optimization algorithm called Animal Migration Optimization (AMO). Animal migration is a ubiquitous phenomenon which exist in majority of animal groups including crustaceans, insects, amphibians, retiles, fish mammals, birds, etc. There are two steps in Animal migration algorithm. They are, process of animal migration and animal update. Movement of animal groups to a new position from current position is simulated in migration process. Update of animals are simulated using probabilistic technique in population updating process [24].

Animal Migration Process

Three basic rules must be obeyed by animals in animals migration process. They are, collisions avoidance with neighbors, movement in same direction of neighbors and maintain close distance with neighbors. Topological ring is used for defining individuals local neighborhood concept, which is demonstrated in figure 3. For every individuals dimensions, length of neighborhood is set as five for simplicity.

On indices set of vectors, neighborhood topology is defined and it is static one. For animal with index i, $\mathrm{i}-2, \mathrm{i}-1, \mathrm{i}, \mathrm{i}+1, \mathrm{i}+2$gives the indices of its neighborhood animals if animal index is 1, neighborhood with animal having indices $\mathrm{NP}-1, \mathrm{NP}, 1, 2, 3$, and so forth. Randomly one neighbor is selected and based on its neighbors, individual's patched images count are updated as expressed below,

$$X_{i,G+1} = X_{i,G} + \delta.(X_{neiborhood,G} - X_{i,G})$$

Where, neighborhood patched images current number is represented as $X_{neiborhood,G}$, random number generator produces δ and Gaussian distribution controls it, ith individual's patched images current number is represented as $\mathrm{X}_{\mathrm{i,G}}$and ith individual's patched images new number is represented as $\mathrm{X}_{\mathrm{i,G+1}}$.

Population Updating Process

Leaving and joining of animals in new population are simulated by this algorithm in population updating process. Few new animals having probability P_a is used for replacing individuals. Based on accuracy of segmentation quality, this probability is used. In a descending order, segmented accuracy are sorted. Individuals probability having best segmented accuracy is $1/\mathrm{NP}$ as indicated by it and individuals with worst segmentation accuracy is 1. Algorithm 1 demonstrates this process. In this , randomly selected integers are represented as $r_1, r_2 \in [1, \ldots, \mathrm{NP}]$ and $\mathrm{r}_1 \neq \mathrm{r}_2 \neq \mathrm{i}$.Newly produced patched images$\mathrm{X}_{\mathrm{i,G+1}}$ are evaluated and comparison with $\mathrm{X}_{\mathrm{i,G,}}$ is done for selecting individual having better objective segmented accuracy.

$$X_i = \begin{cases} X_{i,G} \; if \; f(X_{i,G}) \; is \; better \; than \; f(X_{i,G+1}), \\ X_{i,G+1} \; otherwise \end{cases}$$

ALGORITHM 1: POPULATION UPDATING PROCESS

(1) For i=1 to NP do
(2) For j=1 to D do
(3) If rand $> \mathrm{P_a}$ then
(4) $X_{i,G+1} = X_{r1,G} + rand.(X_{best,G} - X_{i,G}) + rand.(X_{r2,G} - X_{i,G})$
(5) End if
(6) End For
(7) End For

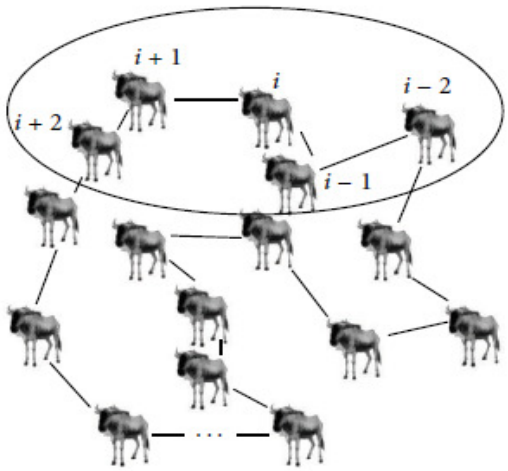


FIGURE 3: THE CONCEPT OF THE NEIGHBORHOOD OF AN ANIMAL

New Migration Process

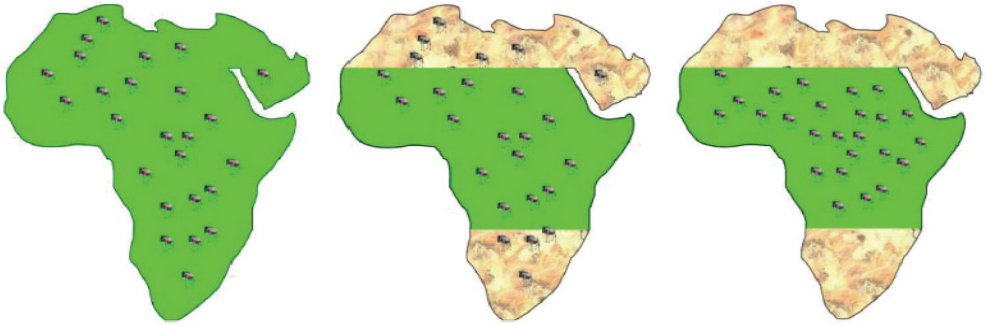
Leader animal which is an individual having best accuracy of segmentation, establishes a living area for new migration process. For simulating animal migration process, animals migrate into new living area from current locations. As indicated in figure 4 (a), in living area, there are NP animals, which are drinking, eating, reproducing, moving and so on. There will be random movement of some individuals and updated their patched images count. Using segmentation accuracy, animals best patched images are computed and stored.

As time goes on, there will be gradual diminishing in amount of water and food which is shown in figure 4 (b). Few animal move to place with abundant food and water from current area with no water and food as shown in figure 4 (c). Living areas with abundant water and food are represented as a green parts in figure 3 and in these areas, animals can live. Areas with water and food lacking is represented as yellow parts and in these areas, animal cannot live. So, they move to new place like green parts of figure 4 (c). Ceaselessly, animals move to a new area for living after some time duration as indicated in figure 4 (a) and (c).

Current best patched image is always near to globally optimum patched image as stated by thumb rule and after every iteration, living area of animals are reduced much more and individuals are getting close to globally optimum patched images. Living area boundary is represented as,

$$low = X_{best} - R, up = X_{best} + R$$
$$R = \rho . R$$

Where, current best patched images are represented as X_{best}, living area's lower bound is represented as low and upper bound is represented as up, living area radius is represented as R, shrinkage coefficient is represented as ρ with $\rho \in (0,1)$, and R, up, low are $1 \times D$ row vectors. Search space size defines R's original value in general. Algorithm's exploration ability is enhanced using huge R value as iteration progresses and exploitation ability is enhanced with small R value.



(a) The Gth iteration living area (b) Animals begin to migrate (c) The G+ 1th iteration living area

FIGURE 4: ANIMALS MIGRATION PROCESS

3. Auto-Encoder based Convolutional Neural Network (AECNN)

For flower identification, every portrait $p^{(i)}$is given with fine-tuned network in parallel. Function f corresponds to AECNN as,

$$f: p^{(i)} \rightarrow \{m_F^{(i)}, m_B^{(i)}\}$$

Every input $p^{(i)}$ is mapped into two pixel-dense scoremaps using above function. Pixel-wise likelihood of pixels in $p^{(i)}$ belonging to foreground is represented as $m_F^{(i)} \in R^{r \times r}$, pixel-wise background likelihood is represented as $m_B^{(i)} \in R^{r \times r}$.

Auto-Encoder

There are two-parts in auto-encoder. Figure 5 shows encoder and decoder. Deterministic mapping function is used for converting input x to a hidden representation y which is a feature code in encoder. It is an affine mapping function, which follows non-linearity [25].

$$y = f(Wx + b)$$

Where, weight between hidden representation y and input x is represented as W and bias is represented as b. Construction process of output z using y is implemented by decoder and it is expressed as,

$$z = f'(W'y + b')$$

Where, weight between output z and hidden representation y is represented as W' and bias is represented as b'. Construction of x is considered as z as like input x.

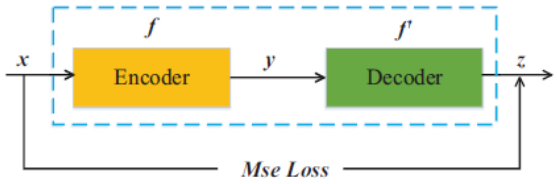


FIGURE 5: THE ARCHITECTURE OF AN AUTO-ENCODER

Reconstruction error can be minimized by training an auto-encoder, cost function J_{AE} minimization is used for realizing the same.

$$J_{AE} = \frac{1}{p}\sum_{i=1}^{p} L[x_i, z_i]$$

Where, input images count is represented as p, input image is represented as x_i and reconstructed image corresponds to $x_i$is represented as z_i, input image x_i's reconstruction error is represented as $L[x_i, z_i]$. Cross entropy or Mean Square Error (MSE) are used for measuring this value. MSE between reconstructed patch of image $z_i(i = 1,2, \dots, p)$ and input image $x_i, (i = 1,2, \dots p)$ is utilized in this work. The reconstructed image can be represented as,

$$L_{AE}[x_i, z_i] = \|x_i - z_i\|^2$$

Inputs are added with convolution operation by combining auto-encoder with local convolution connection using Convolutional auto-encoder. There exist a convolutional decoder and convolutional encoder in convolutional auto-encoder. Convolutional conversion process to feature maps from input is realized using convolutional encoder and convolutional conversion process to output from feature map is carried out using a convolutional decoder.

4. Fusion and Refinement

Predictions obtained for every $p^{(i)} \in P$ are combined for generating two global scoremaps $M_B$and M_F, after the evaluation of every portrait. After padding pixel elimination in I, $p^{(i)}$'s pixel coordinates are represented as $c^{(i)}$. Same resolution as like I is contained in scoremaps $M_B$and $M_F$as defined by fusion procedure.

$$\forall p^{(i)} \in P, M_{F,B}\left(c^{(i)}\right) = m_{F,B,}^{(i)}$$

Exactly one portrait is used for obtaining single prediction score for each image pixel. This is done to avoid artificial boundaries produced artifacts. Softmax function is used after fusion for normalizing scoremaps M_B and $M_F$into scoremaps$\widetilde{M}_B$ and $\widetilde{M}_F$.

$$\widetilde{M}_{F,B}(q_j) = \frac{\exp\left(M_{F,B}(q_j)\right)}{\exp\left(M_B(q_j)\right) + \exp\left(M_F(q_j)\right)}$$

Where, in input image I, j-th pixel is represented as q_j. For every pixel q_j, scores $\widetilde{M}_B\left(q_j\right)$and $\sim\widetilde{M}_F\left(q_j\right)$ are added using this formulation. This indicates probability of q_j belonging to that class. With respect to actual flower boundaries, obtained CNN predictions are coarse. So, image I and score map are given to RGR refinement module, instead of directly thresholding $\widetilde{M}_F$. Two high-confidence classification regions R_F and R_B defines this refinement algorithm and is given by,

$$R_{F,B} = \{q_j | \widetilde{M}_{F,B}(q_j) > \tau_{F,B}\}$$

Where, high-confidence background threshold is represented as τ_B and high-confidence foreground threshold is represented as τ_F. Multiple Monte Carlo region growing steps Figure 6 are performed using Region Growing Refinement (RGR) algorithm with high-confidence regions as starting points for forming clusters using similar pixels. Every generated cluster is classified into background or foreground by RGR via conduction of majority voting procedure.

Figure 6: Coarse Segmentation Refined Segmentation

4. RESULTS AND DISCUSSION

Three public datasets namely, Peach, AppleB and AppleA are used for evaluating the propose method [28]. In diverse controlled environment with various capturing angle, collected the images of various fruit flower species, which are presented in table 1. In USAD orchard, on a sunny day, collected apple tree images for forming AppleB and AppleA dataset. Trellises are used for supporting trees and in rows they are planted in both datasets. A hand-held camera is used for acquiring 147 image collection to form AppleA dataset. Training set is constructed by selecting 100 images from this collection in a random manner and which is used for training AECNN. Test is formed by selecting 30 images randomly from remaining 47 images. With respect to branches, leaves, occlusion, cluttering and size, there will be variation in flowers in dataset. Average area of flower image is 10,730 pixels with 17,150 pixels standard deviation. For imaging, used a utility vehicle enabled with a background unit. This is done for avoiding other row trees in images. Resolution of images in AppleA dataset is $.3\times$ greater than other dataset images. Rather than splitting images into 321×321 pixels portraits, AppleA dataset images are split as 155×155 pixels portraits.

Metrics like Intersection-over-Union (IoU), F-Score(F_1), Recall(R) and Precision (P) defined segmentation accuracy's quantitative analysis [27]. These metrics are computed in pixel level, where in existing methods, it is computed in super-pixel levels.

Expected positive cases ratio that are right defined Precision (P) and it is expressed as,

$$P= T_P/ (T_P+F_P) \quad (1)$$

Positive cases ratio that are predicted correctly defined Recall (R) and it is expressed as,

$$R= T_P/ (T_P+F_N) \quad (2)$$

Recall and Precisions harmonic mean defines balanced F-score (F_1 score) or F-measure and it is expressed as,

$$F\text{-score}= 2T_P/(2T_P+F_P+F_N) \quad (3)$$

TABLE 2: PERFORMANCE COMPARISON METRICS VS. FRUIT FLOWER DETECTION METHODS

Dataset Name	Methods/Metrics	Precision(P)(%)	Recall(R) (%)	F-Score(F_1) (%)	Intersection-over-Union (IoU) (%)
AppleA	**R-CNN**	37.7	58.5	45.7	35.5
	CNN	65.1	75.2	69.8	53.3
	AECNN	82.4	89.7	85.3	73.4
AppleB	**R-CNN**	68.1	60.9	68.5	51.3
	CNN	70.1	69.4	70.2	55.6
	AECNN	77.8	93.2	78.3	65.0
Peach	**R-CNN**	20.5	21.5	22.1	18
	CNN	62.2	65.8	68.2	55.1
	AECNN	88.8	73.3	75.2	61.5

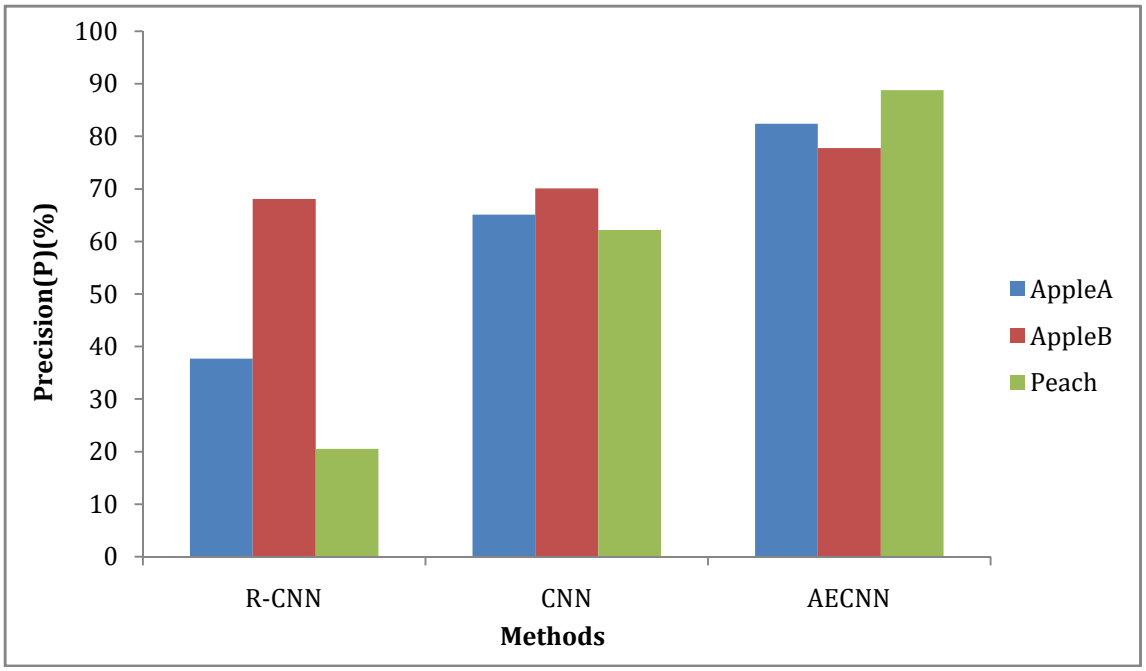


FIGURE 7: PRECISION(P)COMPARISON OF FRUIT FLOWER DETECTION METHODS

Figure 7 demonstrates the precision metric results of three deep learning techniques namely, CNN, R-CNN and proposed AECNN. For three various datasets, better precision value is produced by proposed AECNN, when compared with CNN and R-CNN. They are producing less precision results as demonstrated by results.

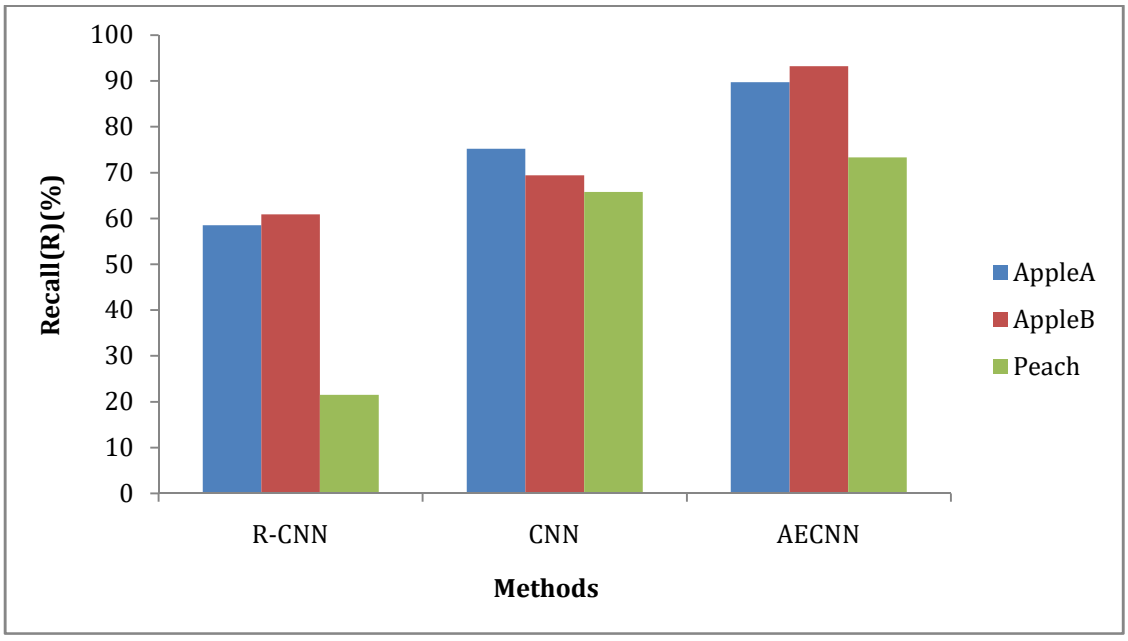


FIGURE 8: RECALL(R)COMPARISON OF FRUIT FLOWER DETECTION METHODS

Figure 8 demonstrates the recall metric results of three deep learning techniques namely, CNN, R-CNN and proposed AECNN. For three various datasets, better recall value is produced by proposed AECNN, when compared with CNN and R-CNN. They are producing less recall results as demonstrated by results.

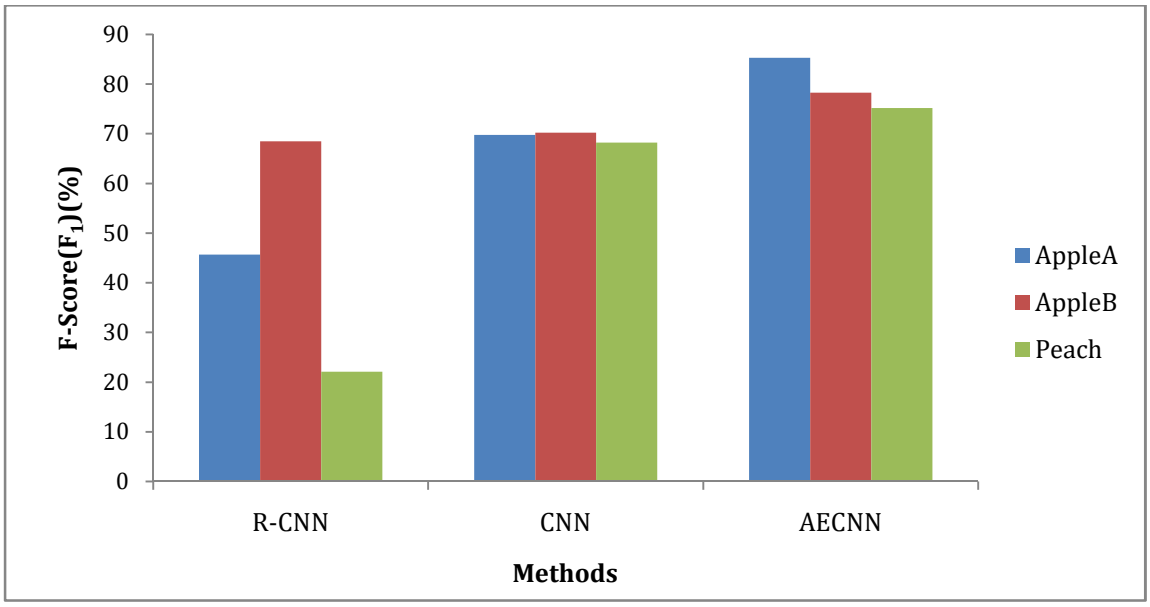


FIGURE 9: F-SCORE(F_1)COMPARISON OF FRUIT FLOWER DETECTION METHODS

Figure 9 demonstrates the F-score metric results of three deep learning techniques namely, CNN, R-CNN and proposed AECNN. For three various datasets, better F-score value is produced by proposed AECNN, when compared with CNN and R-CNN. They are producing less F-score results as demonstrated by results.

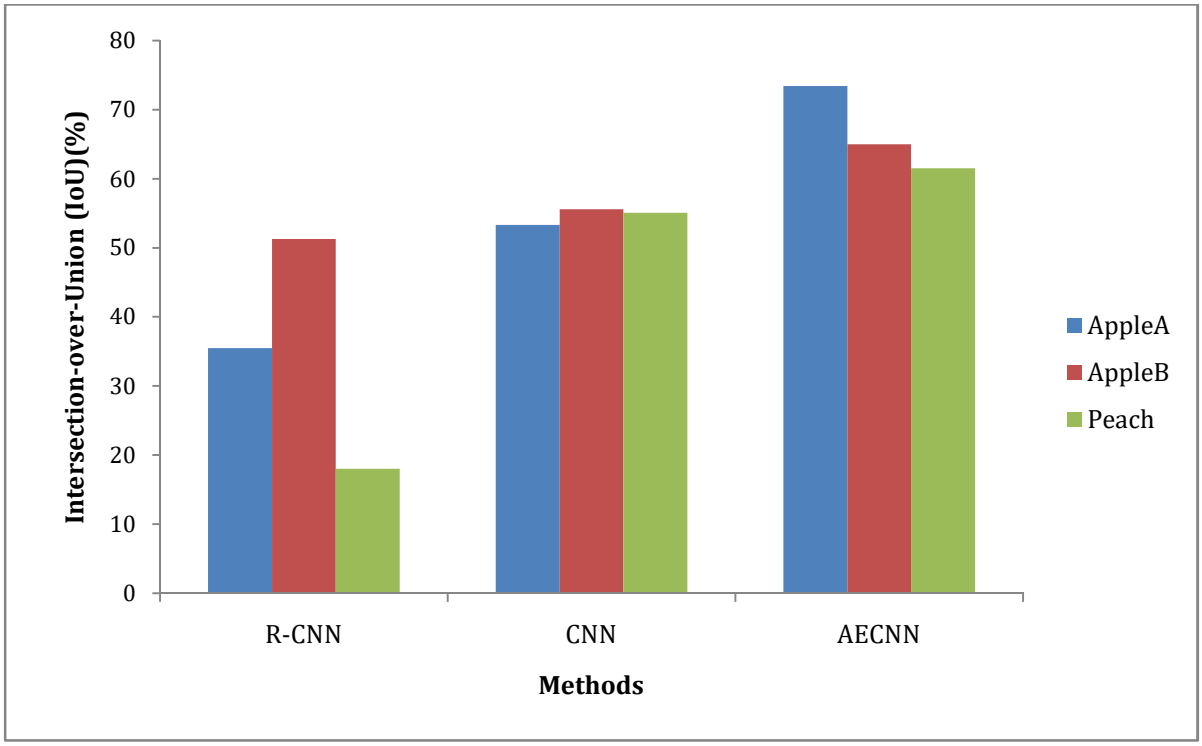


FIGURE 10: INTERSECTION-OVER-UNION (IoU)COMPARISON OF FRUIT FLOWER DETECTION METHODS

Figure 10 demonstrates the Intersection-over-Union (IoU) metric results of three deep learning techniques namely, CNN, R-CNN and proposed AECNN. For three various datasets, better Intersection-over-Union (IoU).

value is produced by proposed AECNN, when compared with CNN and R-CNN. They are producing less Intersection-over-Union (IoU). results as demonstrated by results.

5. CONCLUSION AND FUTURE WORK

In agriculture sector, fruit yield's advance prediction is more important. It is used for fruit yield prediction per orchard as well as for fixing and computing cost of the product. Intensity of flower is used for this detection. For semantic segmentation of images, state-of-the-art deep learning techniques are exploited in novel flower detection methods. In dataset where high variation with respect to flower species, flower density, image resolution, background composition and illumination conditions, better results are produced by proposed method.

This work proposes an Auto-Encoder based Convolutional Neural Network (AECNN) algorithm to detect fruit flower. There are three main stages in this work. They are, training of network for deep Fully Convolutional Network (FCN), pre-processing, reduction of dimension and segmentation. For segmentation of binary flower, this work used DeepLab. In this fine-tuning and network surgery procedures are performed. In trees, undesired branch pruning corresponds to surgery procedure.

On commercial GPU, high resolution image evaluation procedure are defined with deep FCN. GPU memory space is needed for fully convolutional computations and based on image resolution, there will be an exponential increase. From image, noises are reduced or removed using an Additive white Gaussian Noise (AWGN) technique which is introduced next. At last, small patches are formed by splitting high resolution image, AMO is used for reducing dimensionality, fine-tuned AECNN is used for evaluating every patch and final segmentation is performed by applying refinement algorithm. Three datasets of peach and apple flowers images are used for experimentation. For further enhancing this model's generalization ability, multi-species flower datasets can be used in future for training and evaluation. Complete autonomous online bloom intensity estimation system can be developed. For precise agriculture applications like thinning spray treatments and timing prediction, this model can be utilized as this not involving block level flower map creation.

REFERENCES

1. J.P. Underwood, C. Hung, B. Whelan and S. Sukkarieh, Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors, Computers and electronics in agriculture, Vol.130, Pp.83-96, 2016.
2. M. Paustian and L. Theuvsen, Adoption of precision agriculture technologies by German crop farmers, Precision agriculture, Vol.18, No.5, Pp.701-716, 2017.
3. A. Gongal, A. Silwal, S. Amatya, M. Karkee, Q. Zhang and K. Lewis, Apple crop-load estimation with over-the-row machine vision system, Computers and Electronics in Agriculture, Vol.120, Pp. 26–35, 2016.
4. K. Kapach, E. Barnea, R. Mairon, Y. Edan and O. Ben-Shahar, Computer vision for fruit harvesting robots–state of the art and challenges ahead, International Journal of Computational Vision and Robotics, Vol.3, No.1-2, Pp.4–34, 2012.
5. A. Gongal, S. Amatya, M. Karkee, Q. Zhang and K. Lewis, Sensors and systems for fruit detection and localization: A review, Computers and Electronics in Agriculture, Vol.116, Pp.8-19, 2015.
6. P.A. Dias and H. Medeiros, Semantic segmentation refinement by Monte Carlo region growing of high confidence detections, Asian Conference on Computer Vision, Pp.131-146, 2018.
7. M. Dyrmann, A.K. Mortensen, H.S. Midtiby and R.N. Jørgensen, Pixel-wise classification of weeds and crops in images by using a fully convolutional neural network, Proceedings of the International Conference on Agricultural Engineering, Aarhus, Denmark, Pp.26–29, 2016.
8. G.L. Grinblat, L.C. Uzal, M. G. Larese and P.M. Granitto, Deep learning for plant identification using vein morphological patterns, Computers and Electronics in Agriculture, Vol.127, Pp.418–424, 2016.
9. F. Mola, J. Antoch, L. Frigau and C. Conversano, Classification of Images Background Subtraction in Image Segmentation, Acta Universitatis Palackianae Olomucensis, Facultas Rerum Naturalium. Mathematica, Vol.55, No.1, Pp.73-86, 2016.

10. P.A. Dias, A. Tabb and H. Medeiros, Apple flower detection using deep convolutional networks, Computers in Industry, Vol.99, Pp.17-28, 2018.
11. P. Lin and Y. Chen, Detection of Strawberry Flowers in Outdoor Field by Deep Neural Network, IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Pp.482-486, 2018.
12. A. Koirala, K.B. Walsh, Z. Wang and N. Anderson, Deep Learning for Mango (Mangifera indica) Panicle Stage Classification, Agronomy, Vol.10, No.1, Pp.1-22, 2020.
13. R. Horton, E. Cano, D. Bulanon and E. Fallahi, Peach flower monitoring using aerial multispectral imaging, Journal of Imaging, Vol.3, No.1, Pp.1-10, 2017.
14. M. Hočevar, B. Širok, T. Godeša and M. Stopar, Flowering estimation in apple orchards by image analysis, Precision agriculture, Vol.15, No.4, Pp.466-478, 2014.
15. P.A. Dias, A. Tabb and H. Medeiros, Multispecies fruit flower detection using a refined semantic segmentation network, IEEE Robotics and Automation Letters, Vol.3, No.4, Pp.3003-3010, 2018.
16. I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez and C. McCool, Deepfruits: A fruit detection system using deep neural networks, Sensors, Vol.16, No.8, Pp.1-23, 2016.
17. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, Caffe: Convolutional architecture for fast feature embedding, Proceedings of the 22nd ACM International Conference on Multimedia, Pp.675–678, 2014.
18. H. Al-Ghaib and R. Adhami, On the digital image additive white Gaussian noise estimation, International Conference on Industrial Automation, Information and Communications Technology, Pp.90-96, 2014.
19. P. Carbone, J. Schoukens, I. Kollár and A. Moschitta, Measuring the noise cumulative distribution function using quantized data, IEEE Transactions on Instrumentation and Measurement, Vol.65, No.7, Pp.1540-1546, 2016.
20. J. Li, H. Liu, J. He and P. Yang, Application of Non-local Mean Filtering in Real Head MR Image, Australian & New Zealand Control Conference (ANZCC), Pp.330-333, 2018.
21. A. Peng, S. Luo, H. Zeng and Y. Wu, Median filtering forensics using multiple models in residual domain, IEEE Access, Vol.7, Pp.28525-28538, 2019.
22. J. Tang, Y. Wang, W. Cao and J. Yang, Improved Adaptive Median Filtering for Structured Light Image Denoising, IEEE 7th International Conference on Information, Communication and Networks (ICICN), Pp.146-149, 2019.
23. S.H. Chan, T. Zickler and Y.M. Lu, Understanding symmetric smoothing filters: A Gaussian mixture model perspective, IEEE Transactions on Image Processing, Vol.26, No.11, Pp.5107-5121, 2017.
24. M. Ma, Q. Luo, Y. Zhou, X. Chen and L. Li, An improved animal migration optimization algorithm for clustering analysis, Discrete Dynamics in Nature and Society, 2015.
25. B. Hou and R. Yan, Convolutional Auto-Encoder Based Deep Feature Learning for Finger-Vein Verification, IEEE International Symposium on Medical Measurements and Applications (MeMeA), Pp.1-5, 2018.
26. Y. Chen, W.S. Lee, H. Gan, N. Peres, C. Fraisse, Y. Zhang and Y. He, Strawberry Yield Prediction Based on a Deep Neural Network Using High-Resolution Aerial Orthoimages, Remote Sensing, Vol.11, No.13, Pp.1-21, 2019.
27. M. Everingham, S.A. Eslami, L. Van Gool, C.K. Williams, J. Winn and A. Zisserman, The pascal visual object classes challenge: A retrospective, International journal of computer vision, Vol.111, No.1, Pp.98-136, 2015.
28. https://data.nal.usda.gov/dataset/data-multi-species-fruit-flower-detection-using-refined-semantic-segmentation-network