An Optimized Deep Learning Model For Flower Classification Using NAS-FPN And Faster R-CNN

Isha Patel, Sanskruti Patel

Abstract— In computer vision, object detection is widely used in many applications such as face detection, video surveillance, vehicle detection, plant leaf detection etc. Deep neural networks have greater capabilities for image pattern recognition and are widely used in Computer Vision algorithms. In this paper, a deep convolutional neural network based on NAS-FPN and Faster R-CNN is proposed for flower object detection, localization and classification. Using the method of transfer learning, different pre-trained models including ResNet 50, ResNet 101, Inception ResNet V2, Inception V2, NAS, and MobileNet V2 are trained and evaluated on flower 30 dataset and flower 102 dataset that contains 19679 flower images. Based on the experiment carried out, the result demonstrates that the performance of the proposed NAS-FPN with Faster R-CNN model using transfer learning approach gives optimum mAP score of 87.6% on 102 flower class and 96.2% on 30 flower class datasets. Also, the proposed model is able to detect, locate and classify flowers with other significant details that includes flower name, division, class, subclass, order, family, and herb flower using multi-class classification and multi-labeling techniques.

Index Terms— Deep Convolutional Neural Network, Object detection, NAS-FPN, Faster R-CNN, Multi-labeling.

1 Introduction

Deep learning is an emerging field of machine learning that has been grown rapidly and may apply to many domains with high success frequency including image processing, speech recognition and text understanding. Deep learning algorithms are mainly inheritor of artificial neural network architecture with higher number of hidden layers, therefore known as deep neural networks [1]. Computer Vision aims to make computers process, analyze images and videos and extract details in the same way a human mind does. In last few years, there has been significant progress observed of applying deep neural networks for image pattern recognition and hence they are widely used in Computer Vision algorithms. For analyzing visual images, a class of deep neural networks, Convolutional Neural Network (CNN), is most commonly applied [2]. Large availabilities of datasets and support of powerful Graphics Processing Units (GPU) makes the integration of deep learning methods with computer vision popular. As deep learning requires large datasets and powerful resources to perform training and tuning of the model, both the requirements have been satisfied in the field of agriculture. Motivated by the results available for image classification and object detection using deep learning CNN models, we aimed to evaluate and apply CNN based deep learning techniques for flower object detection, localization and classification. It is a challenge in the field of agriculture to perform multi-class flower detection, localization and classification on flower image dataset efficiently. The identification of flower species in real-time environment has great significance and theoretical value [3, 4]. According to the available research of applying deep learning technologies, less significant work has been done to detect the diversity in flower images, when there are more than one flower objects presented in the image or there is an overlap between flowers objects presented in one image. Moreover, flower images have complex background scene that makes the detection and classification process time-consuming and less accurate, particularly with a large number of species. In such images, it is required to detect the multiple flower class and its location respect to each flower object from an input image [5, 6]. In this paper, we have proposed an optimized architecture for detection and recognition of flower species using widely used deep learning techniques, NAS-FPN (Neural Architecture Search-Feature Pyramid Network) and Faster R-CNN (Faster Region based Convolutional Neural Network). Based on the experiment carried out and the results obtained, the proposed architecture is able to detect, locate and classify flowers with an optimum accuracy with other significant details that includes flower division, class, subclass, order, family and herb flower or not. The rest of the paper is organized as follows. The related work is reviewed in section 2. In section 3, an optimized object detection model using NAS-FPN and Faster R-CNN for flower detection has been discussed. In section 4, it contains the details regarding experiment setup and datasets used. The experimental results and performance evaluation metrics of flower detection are discussed in section 5. The conclusion is presented in section 6.

- ***Isha Patel**, Faculty of Computer Science and Applications, Charotar University of Science and Technology (CHARUSAT), Changa, Gujarat 388421, India.*
- ***Sanskruti Patel**, Faculty of Computer Science and Applications, Charotar University of Science and Technology (CHARUSAT), Changa, Gujarat 388421, India.*

2 Literature Review

Mengxiao Tian et al. [6] described the SSD model for flower detection and identification. The flower data set published by Oxford University was used for the experiment. The experimental results of the average accuracy are 83.64% based on the evaluation standard of Pascal VOC2007, and 87.4% based on the evaluation standard of Pascal VOC2012. Swati Kosankar et al. [7] presented experimental implementation in their paper using MobileNet model on the TensorFlow platform. They have retrained the 102 flower category datasets, which achieved 70.6% accuracy. Hazem Hiary et al. [8] presented their work regarding placing the automatically segmentation of flower images by applying

bounding boxes around the flower object. They have also applied CNN classifier to distinguish the different flower types and also evaluated their method on three frequently used flower datasets. Their classification results were 81%, 78.7%, and 77.3% respectively. Isha et al. [9] and Tao Kong et al. [10] applied multi-scale representations of FPN and then reformulate the feature reconfiguration process. Also, combine low-level and high-level features were effectively combined to increase the efficiency of the model. Weifeng Ge et al. [26] performed a transfer learning approach, for improving the performance of deep learning tasks with insufficient training data. Shaoqing Ren et al. [36] trained the full-image convolutional features with end-to-end RPN detection networks. They applied Fast R-CNN to generate high-quality region proposals to detect an object and merged into a single network by sharing their convolutional features. Yun Ren et al. [39] implemented small object detection in optical remote sensing images. Longsheng Fu et al. [42] performed a detection on kiwifruit using deep convolutional neural network. A Faster R-CNN was trained end-to-end by using SGD techniques with ZFNet and back-propagation. The AP of the kiwifruit detector was 89.3%. Faming Shao et al. [44] presented the Faster R-CNN for traffic sign detection in real traffic situations. RPN based on SGWs and MSERs is proposed. Shaoming Zhang et al. [48] adopted an improved Faster-RCNN, and CNN with an integration of multiresolution convolutional features and performed ROI pooling on a larger feature map in an RPN using the VGG16. Marco Seeland et al. [55] compared various methods of combinations, evaluate, and their parameters towards classification accuracy. Also, various features extracted from flower images like shape and color using three types of flower datasets; Oxford Flower 17, 102 and Jena Flower 30. The accuracy obtained are 49.9%, 79.9%, and 67.7% respectively.

3 An Integrated architecture using NAS-FPN and Faster R-CNN for Flower object detection, localization and classification

Computer Vision is an interdisciplinary field that comprise with methods used for acquiring, processing, analysing and understanding digital images or videos. Deep learning is a subfield of machine learning that comprises with deep neural network architecture and algorithms that learns to perform the task automatically [11]. Computer vision uses techniques from machine learning and, in turn, some machine learning techniques are developed specially for performing computer vision tasks. The association between two fields helps to create a predictive model that has power to predict over unknown future data. The following figure 1 shows the proposed integrated architecture that contains the three-step process for flower object detection, localization and classification.

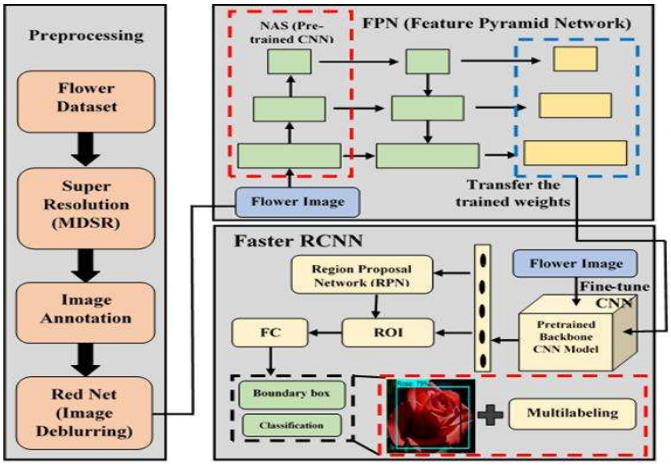

Fig. 1. An Integrated deep learning model for Flower classification.

In the first step, two pre-processing techniques, super-resolution (SR) and RED Net are applied due to the existence of low resolution and noisy images in the dataset to enhance image quality. An image super-resolution reconstruction algorithm reconstructs a high-resolution (HR) image from one or multiple low-resolution (LR) images [12]. A type of super-resolution method, MDSR (multi-scale deep super-resolution system), handles the SR at multi-scales; first, pre-processing modules are used to reduce the variance from input images of different scales, and at the end of multi-scale model, scale-specific upscaling modules are located in parallel to handle multi-scale reconstruction [15-16]. The generated HR images are rich in content, they are considered for image annotation. For a machine to understand image, the training data needs to be labeled and presented in a language that the machine would eventually learn and implement by itself. For that, image annotation technique, 2D Bounding Boxes, is applied that draws rectangles or cuboids around objects in an image and labels them to respective flower class. For image restoration, a very deep CNN-based framework, RED-Net (Residual Encoder-Decoder Network), is applied [21]. We observed that it is beneficiary to train a very deep model for low-level tasks like denoising, and deblurring. It will generate the clean flower image from the corrupted image. In the second step, a pre-trained network with transfer learning is applied that is faster and easier approach than training a network from scratch [24]. In the deep learning implementation, it is required to use a pre-trained image classification network. Several pre-trained networks which have already learned to extract powerful and informative features from natural images are available as a starting point. The pre-trained models experimented and used in this research work are trained on a COCO (Common Objects in Context) dataset [51]. In deep learning, Feature Pyramid Network (FPN) offers pyramid representation for object detection tasks [29]. NAS-FPN (Neural Architecture Search-Feature Pyramid Network)) is an automatic neural architecture search algorithm integrated with FPN that focuses on finding optimal connections between different layers for pyramidal representations [30]. In the third step, the trained weights are transferred to Faster R-CNN, a well-known deep convolutional neural network for object detection that is used as feature extractors to obtain high-level features from input images [35]. Faster R-CNN uses a fully convolutional network paired with a classification deep convolutional network, to

locate regional proposals, which improves training and testing speed while also increasing detection accuracy [36 and 37]. It consists of fine-tune convolutional layers, region proposal networks (RPN), an ROI pooling layer, and a classifier which is shown in above figure 1. The model was trained on COCO dataset and fine-tuned by initializing a new classification layer and updating all layers for both the region proposal network and classification networks [39 and 51]. At the end classifier layer is the final layer of the whole Faster R-CNN model and just behind the ROI Pooling layer. It takes as an input the region proposed by the RPN and predict object class (classification) and bounding box (regression) [38, 40 and 41]. In this paper, the outcome of classifier layer is prediction of the flower class (classification) with localization (bounding box) of flower image.

3.1 Preprocessing of the flower images

Preprocessing normally supress noise and distortion from the image and enhance the image quality. By enhancing important features, it facilitates subsequent Deep learning and computer vision tasks to produce more efficient results. The preprocessing techniques applied in this research work are described as below.

3.1.1 Super Resolution

With the advent of deep convolutional neural networks, a progress has been observed in the techniques for super-resolution (SR) of images. It is required as multiple datasets have been chosen to train the model where some of the datasets contain Low-resolution (LR) images which need to convert images into High-resolution (HR) images for better image annotation and better accuracy for even small object detection tasks [13]. Methods available for super-resolution often classified into three categories. They are interpolation-based methods, reconstruction-based methods, and learning-based methods. The learning-based method has widely used as it produces promising results compared to other algorithms. Learning-based methods uses a machine learning algorithm to generate the mapping relationship between high resolution and low resolution images [14]. In this research, a multi-scale deep super resolution (MDSR) method is applied that reconstructs HR flower images of different upscaling factors using a single model. The following figure 2 describes MDSR model architecture [16].

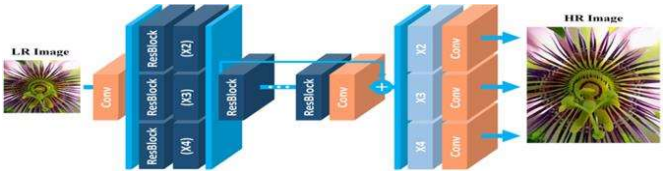

Fig. 2. Architecture of MDSR to generate HR images.

Multi-scale learning refers to passing features through multiple paths of the model, which performs different operations, for providing better modeling capabilities [17]. Using multi-image schemes (MDSR), images are resized at multiple scales. Therefore, super-resolution is required for fast and accurate flower detection and it achieved significant improvements, both quantitatively and qualitatively.

3.1.2 Image Annotation

Image annotation is one of the most important tasks in computer vision for detecting objects. Image annotation is the human-powered task of annotating an image with labels. These labels are predetermined by the engineer and are chosen to give the computer vision model information about what is represented in the image. The number of labels on each image can vary largely. Some datasets will require only one label to represent the content of an entire image i.e. image classification. Other datasets could require multiple objects to be tagged within a single image, each with a different label. There are different types of annotation accomplished on images like bounding box annotation, Polygon annotation, Semantic annotation, Key point annotation, and Redaction. Also, different tools are used for Image Annotation like LabelImg, Annotorious, Comma Colouring and VGG Image Annotator (VIA) [18]. To label flower datasets used in this research work, image labeling software LabelImg is used. The following figure 3 described the usage of LabelImg tool for flower image annotation.

Fig. 3. Image annotation using 2D bounding boxes method.

Here, we have applied 2D bounding boxes annotation for flower detection. LabelImg saves a .xml file containing the label data for each flower image. When each flower image is labeled and saved, there will be one .xml file for each flower image is generated [19]. The labeled datasets are divided into three categories which are; train, test, and validation. After the flower images labeled, we need to create TFRecords, which convert the .xml files to .csv files and that can be served as input data for training of the object detector. The TFRecord file format is a simple record-oriented binary format for machine learning training data [20]. That also generates three files; train.record, test.record and validation.record. A TFRecord combines all the labels (bounding boxes) and images for the entire dataset into one file. These files can be used to train and fine-tune proposed flower object detection model. This annotated image data would train the model for unique visual characteristics to each type of flower species.

3.1.3 Image Restoration

For image restoration such as applying denoising and deblurring operations, we have used Residual Encoder-Decoder Network, known as RED-Net. It is a very deep fully convolutional encoding-decoding framework that composed of several layers of convolution and de-convolution operators [21]. They are used for learning end-to-end mappings from corrupted images to the original ones. The framework offers a set of convolutional layers, work as feature extractor, that capture the abstraction of image contents. It preserves the primary components of objects, while eliminating noises and corruptions [23]. It may possible that the refined details of the image contents lost during this process. To

recover the image details, de-convolutional layers are applied. The output of the de-convolutional layers is the "clean" version of the input image. Skip-layer connections with symmetrically link convolutional and de-convolutional layers have applied [22]. Due to this, the training converged much faster and better performance is achieved. Significantly, with the large capacity, we can handle different levels of noise images using a single model. The following figure 4 describes the architecture of RED-Net for flower image restoring.

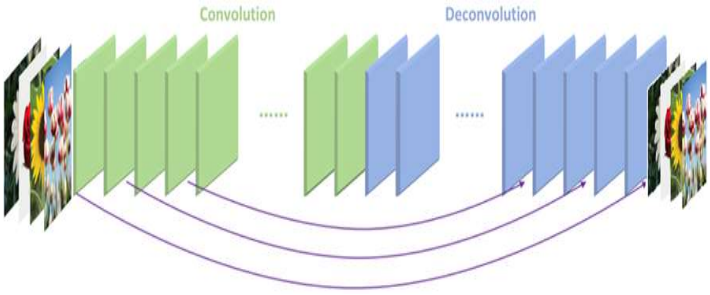

Fig. 4. RED-Net architecture for image restoration.

The kernel size for convolution and de-convolution is set to 3X3, which is shown excellent image recognition performance. We have used RED-Net architecture in our experiment with 30 layers for denoising and deblurring of flower image.

3.2 Transfer Learning with NAS-FPN

Transfer learning is an adaption of pre-trained models to perform similar or moderately different tasks, by fine-tuning parameters of the pre-trained models [25, 26, and 27]. Transfer learning toolkit makes it easy to prune and retrain models. It normally used when a small training dataset is available and found that a problem is similar to the task for which there is an availability of pre-trained models. In the case of object detection, parameter transfer is possible in fewer layers of the neural network architecture [28]. To improve the performance of many computer vision tasks, multi-scale features from different layers are combined. FPN is a pyramid representation that combines low-resolution with strong semantic features and week semantic features with high-resolution via top-down and lateral connections [29]. Applying Neural Architecture Search (NAS) on FPN generates NAS-FPN network that provides a flexible and efficient way for building accurate object detection model [30]. The integration of NAS-FPN produces a significant improvement upon many backbone architectures. Therefore, we have used pre-trained NAS-FPN as a feature encoder that transfer the weights for fine tuning the Faster R-CNN backbone CNN model. The development of this pyramid involves a bottom-up pathway, a top-down pathway, lateral connections, and predict [31] the result as described in the following figure 5.

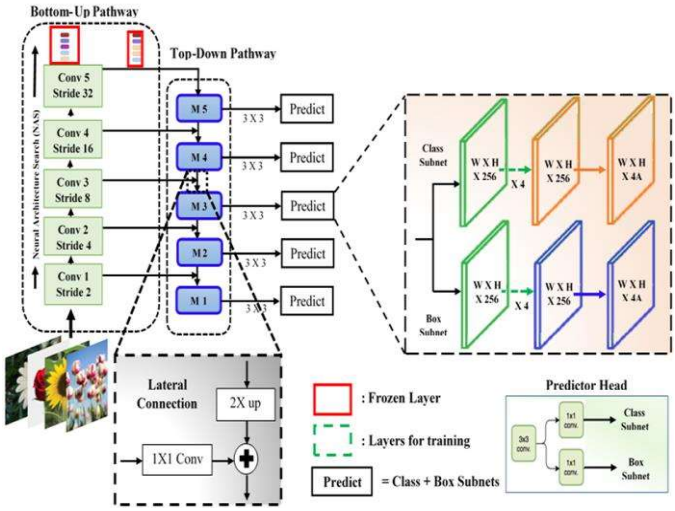

Fig. 5. Architecture of NAS-FPN (Neural Network Search-Feature Pyramid Network).

3.2.1 Bottom-up pathway

To construct a bottom-up pathway, the feed-forward propagation is performed on the backbone convolutional neural network. Here, we applied the NAS pre-trained model to compute a feature hierarchy. This hierarchy consists of feature maps. During transfer learning, the last convolutional layers (i.e. fully connected layer and softmax layer) are to be freeze. We have used five feature map convolution layers described as conv1, conv2, conv3, conv4, and conv5, where each layer has combined with the corresponding stride of (2, 4, 8, 16, 32) pixels concerning to the input image. We have chosen the output of the last layer in each stage as feature maps, which passed to the next pathway (top-down) to create the desired pyramid [32-33].

3.2.2 Lateral Connections and Top-down pathway

The bottom-up pathway passes the feature map to top-down pathway to upsamle the feature map. It constructs the same size pathway which is semantically coarser but having spatially stronger feature maps. To merge feature maps of the same spatial size from the bottom-up pathway and the top-down pathway, a series of lateral connections are applied [32-33]. A factor of 2x is considered for upsampling of the spatial resolution along with a coarser-resolution feature map. The generated upsampled featured map is then merged with the corresponding bottom-up feature map through element wise addition. For this, it applied a 1×1 convolutional layer that is used to reduce the channel dimensions. Till the finest resolution map is generated, the process is iterated. To start the process, a 1×1 convolutional layer is integrated on conv 5 layer. This will produce the coarsest resolution map. At the end of the process, a 3×3 convolution is applied on each merged map in order to generate the absolute feature map [34].

3.2.3 Predict

During the process of upsampling, the aliasing effect is generated. In order to eliminate it, on each merged feature map, a 3 x 3 convolution is executed. Finally, the finished feature maps are generated; namely (P1, P2, P3, P4, P5) corresponding to (M1, M2, M3, M4, M5) that are respective to the same spatial sizes [34]. It is required to fix the dimension of feature as all levels of the pyramid used shared classifiers and/or regressors. The

dimension of features is annotated as numbers of channels where channels are denoted as d. Here, d = 256 is considered and thus all extra convolutional layers have 256 channel outputs. The NAS-FPN architecture is used as it considers both inputs and outputs of a pyramid network as feature layers in the deep identical feature extraction scales, and thus produces better detection and accuracy [30, 32, and 33].

3.3 Modified Faster R-CNN for Flower Detection

Belonging to the wider family of architecture based on convolutional neural network, Faster R-CNN is proposed as an improvisation of its predecessors [35] that inputs the source image to a CNN called a Region Prediction Network (RPN) [43]. RPN generates proposals that are the regions which have the highest probability to contain the objects of interest. The pooling layer, named as Region of Interest (RoI), classifies the images within each region proposal to predict the offset values for the bounding boxes. We have considered an improved Faster Region Convolutional Neural Network (Faster R-CNN) detection network for flower object detection as it calculates the loss function of classification and regression and for that it uses a set of anchor boxes on each pixel. The Faster R-CNN is a two-stage detection model as described in figure 6. This process is divided into five steps; in the first step, using the transfer learning approach a pre-trained model NAS-FPN shared feature map weights to CNN architecture (i.e. ResNet 50 V1) [46]. In the second step, the process of feature extraction is performed by CNN using fine-tuned convolution neural network backbone architecture. Convolution feature maps are generated at the end of the last layer. In the third step, anchor boxes are generated based on the feature map using a sliding window approach. The presence of objects is indicated by refining these anchor boxes in the next step. In the fourth step, generated anchors are refined using a smaller network which calculates the loss function to select top anchors containing objects. Finally, in the last step, it classifies the image with prediction of the class (classification) and bounding box (localization) [47 and 48].

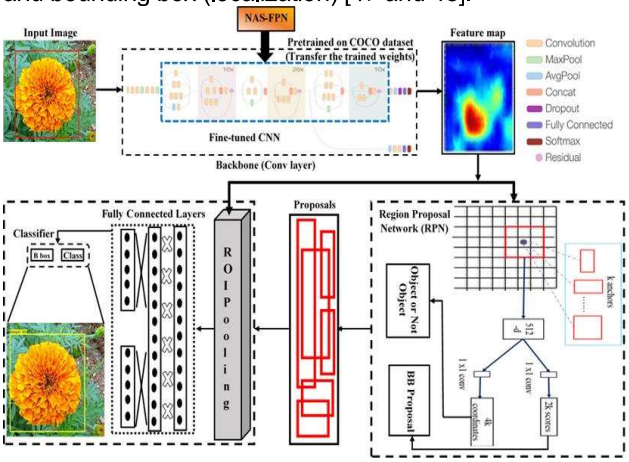

Fig. 6. Modified Faster R-CNN with Integration of NAS-FPN.

Based on the pre-training NAS-FPN architecture, feature extraction generates strong semantic features and week semantic features with high-resolution as training data to share the weights with the Faster R-CNN backbone fine-tuned CNN [45 and 46]. And then, the combination of pre-trained network and fine-tuned network generates strong feature maps for the RPN with the sliding window. For the region proposal network, the prerequisite step is the extraction of convolution features that are extracted using the backbone network.

3.3.1 The backbone network

The use of the transfer learning approach is to solve the complexity of classifying the multiclass images and to obtain a high precision training of a large number of the dataset. Therefore, the backbone model of CNN uses the COCO dataset, implemented on ResNet-50 V1, to get the pre-trained model to build the RPN and Faster-RCNN network [42 and 44]. ResNet-50 is a special type of convolution neural networks that is used to learn features between NAS-FPN fine-tuned models. The beginning of the convolutional neural network in the ResNet-50 V1 network retrains the weights from the NAS-FPN architecture. Since it is used as initialization parameter of the Faster R-CNN, it shares convolutional layer to extract features of the image [46].

3.3.2 Region proposal network (RPN)

Region proposal network generates different proposal of regions and outputs them to the detection network to realize the identification for the proposed region. RPN consists of three parts: (i) anchor window, (ii) loss function, and (iii) set of region proposals. In the first step, a set of various sized rectangle boxes generated around the object, called as anchor, where the number of maximum possible proposals for each location is denoted as 'k'. Therefore, the outcomes of the regression layer are 4k, and the classification layer are 2k; for an object or not object for each proposal. In the second step, two loss functions have been applied to calculate the prediction error in RPN. Binary cross-entropy loss is used for classified correct ground truth box, whereas smooth L1 loss is used to take the absolute value and penalises different location errors. At the last step, the combination of the backbone network feature map and RPN proposed the set of region proposals, which is used as input for the next layer (i.e. RoI Pooling (Region of Interest) layer) [42 and 44].

3.3.3 ROI Pooling

ROI pooling layer is used to convert the features obtained from the fine-tuned CNN layer to a fixed-sized of feature maps. The selected boxes are further fine-tuned using the loss function at the end of the RPN. The fully connected layer is used to realize classification for the target. The feature vector is divided into two layers; one is the softmax layer and another is the regression layer. The performance of the classifier is a multi-class classification which classifies bounding box in more than one class. The regressor defines the location of the predicted bounding box [42 and 44]. The formula of loss function which calculates the loss for both regression and classification is given in the following Eq.1:

$$L\left(\{p_i\},\{t_i\}\right) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where i is the index of an anchor in a training and pi is the predicted probability of anchor i. ti = {tx; ty; tw; th} is a vector representing the four parameterized coordinates of the predicted bounding box. p_i^* is the ground-truth label. If the given label is one then it is positive, and the label is zero then it is negative.

3.3.4 Classification and bounding box regression

The purpose of the classification and regression stage is to use similar size of feature maps that are generated from ROI pooling [44]. The optimization of the loss function is calculated based on bounding box classification and regression loss. The classification result generates the class for each category, and the bounding box regression creates the rectangular box around the particular object to localize it.

3.3.5 Multi-class and Multi-Label Classification

Multi-class and multi-label image classification is one of the pivotal and long-lasting problem in computer vision. Classifying an image into more than one class known as a multi-class classification problem [49], where as to categorize an object from an image into more than one label is known as multi-label classification problem [50]. Now, there can be two scenarios of multi-class classification: First, each flower image contains only a single flower object and hence, it can only be classified in one of the categories. Second, the flower image might contain more than one flower object and but the image is required to classify in only one of the categories. The following figure. 7 describes the multi-class and multi-label classification.

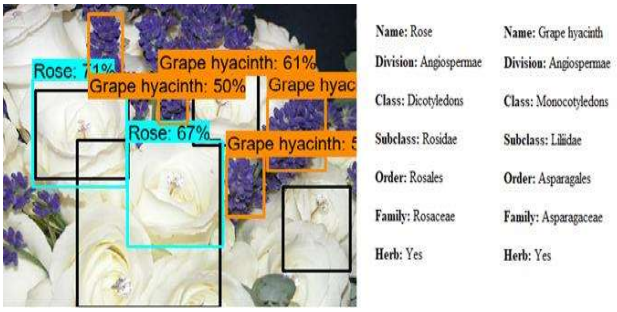

Fig. 7. Multi-clas and Multi-label classification of Flower Image.

For example, in the above figure 7, an image is classified in two image class i.e. Rose and Grape hyacinth. Further, both of the flowers are labeled with its flower division, class, subclass, order, family and herb flower or not.

4 Flower Datasets and Experiment Setup

There are 4 distinct types of datasets used during experiment. Dataset 1 has a total 102 flower class and a total of 8189 images. Dataset 2 has a total of 17 flower classes and a total of 1360 images. Dataset 3 has a total of 5 flower classes and a total of 3500 images. Dataset 4 has a total of 20 flower classes and a total of 5157 images. The final dataset 5 is a combination of all the above datasets (Dataset 1, 2, 3, and 4) which has a 102 flower class and a total of 18200 images. One more dataset 6 has a total of 30 flower class and a total of 1479 images.

4.1 Description of Flower Dataset

Dataset 1: Dataset 1 consists of 102 categories of flowers with 18200 total images. It is a combination of multiple datasets which is described as follows;

Table. 1. Description of Flower Datasets

Dataset Name	Categories	Images	Description
Flower102 dataset [52]	102	8189	Each category contains between 40 and 258 images. Moreover, this dataset is more challenging than the Flower17 dataset since it has more images and some of the having similar categories.
Kaggle's Flower dataset [54]	05	3500	This dataset is more challenging than Flower102 and Flower17 dataset since it has too many low-resolution images with too many are similar and presence of different flowers in the same image.
Flower17 dataset [53]	13	1360	The diversity between classes and small differences between categories makes it challenging.
Dataset Accumulated	20	5157	This dataset, was created by our self, where each category contains 52 to 503 different images.

Dataset 2: This dataset consists of 30 classes based on common wild-flowering species. Each class is representing 11 to 70 images with a total of 1,479 images [55]. This dataset is different in flower classes to compare the other flowering dataset. Also, the images are very noisy. It is also known as the Jenna30 dataset.

4.2 API and Experiment Setup

The experiment carried out on a machine using the operating platform Windows 10, which uses the TensorFlow Object Detection API framework. The setup and prerequisite software are installed on a machine equipped with 32 GB RAM, Intel® Core™ i7 8th generation processor, NVIDIA Titan Xp GPU. The software tools used are Anaconda virtual environment included CUDA 10, CUDNN 7.6, Python 3.8 and Microsoft Visual Studio for editing purposes. Tensorflow Object Detection API also used and it depends on the various libraries like; Protobuf 3.0.0, Python-tk, Pillow 1.0, lxml, Jupyter notebook, Matplotlib, Tensorflow 2.0, Cython, contextlib2, coco API.

5 Experiment Results and Performance Analysis

To evaluate the effectiveness of the proposed integrated approach, a series of comparisons of different object detection models have been done. In this research, three object detection models are implemented over two types of datasets. They are namely Faster R-CNN, SSD and the proposed integrated NAS-FPN with Faster R-CNN. During the experiment we have; (1) trained the Faster R-CNN, SSD and proposed NAS-FPN with Faster R-CNN model on the two flower datasets, one is having 102 flower class and another is having 30 flower class images and analyzed its performance; (2) Trained the object detection models using transfer learning approach on different backbones that includes Inception ResNet V2, Inception V2, ResNet 50, ResNet 101, NAS, MobileNet V2, ResNet 50 V1, MobileNet V1.

5.1 Quantitative Analysis of Flower Detection Performance

Both quantitative and qualitative measurements were taken to evaluate the performance of above mentioned three approaches. The average precision (AP), average recall (AR), and mean average precision (mAP) were used as the evaluation metric for flower detection for quantitative analysis [56 and 57]. Precision and Recall are two most commonly used parameters and they are calculated according to the following equations:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Where TP is the true positive; FP indicates false positives and FN indicates false negatives. Moreover, the correctness of a detected object is also evaluated by the intersection-over-union (IoU) overlap with the corresponding ground truth bounding box [58]. It is calculated according to the following equation:

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

If the IoU is greater than the threshold value, it was considered as a true positive (TP). In case of non matching of a detected object with the ground truth bounding box, it was considered to be a false positive (FP). Moreover, in the occurrence of missed ground truth bounding box, a false negative (FN) is determined. In this research, we have chosen 0.5 and 0.75 for the threshold value. The evolution of AP across scales are AP for small, medium, and large object. Same as for AR across scales are AR for small, medium, and large object [59]. The overall detection performance was measured with the mean average precision (mAP) score, which is the average AP value over all classes [60]. The higher the mAP was, the better the overall detection performance of the flower dataset. The detection performance comparison results of the different object detection models are shown in Tables 2, 3, 4 and 5.

Table. 2. The Performance of Average Precision (AP) for different object detection models for J30 flower class dataset.

Object Detection Model	Backbone Pre-trained model	AP, IoU			AP, Area		
		0.5:0.95	0.50	0.75	Small	Medium	Large
Faster RCNN	ResNet 50	0.589	0.912	0.672	-	0.314	0.592
	ResNet 101	0.709	0.932	0.843	-	0.411	0.711
	Inception ResNet V2	0.678	0.913	0.803	-	0.449	0.680
	Inception V2	0.665	0.924	0.815	-	0.236	0.667
	NAS	0.628	0.865	0.728	-	0.034	0.630
SSD	MobileNet V2	0.770	0.957	0.876	-	0.080	0.776
Proposed Model (NAS-FPN with Faster R-CNN)	ResNet 50 V1	**0.824**	**0.974**	**0.915**	**-**	**0.181**	**0.828**

Table. 3. The Performance of Average Recall (AR) for different object detection models for J30 flower class dataset.

Object Detection Model	Backbone Pre-trained model	AR, Detections			AR, Area		
		1	10	100	Small	Medium	Large
Faster RCNN	ResNet 50	0.610	0.668	0.671	-	0.433	0.672
	ResNet 101	0.723	0.757	0.766	-	0.433	0.768
	Inception ResNet V2	0.696	0.737	0.737	-	0.467	0.738
	Inception V2	0.690	0.729	0.733	-	0.233	0.736
	NAS	0.677	0.708	0.709	-	0.067	0.712
SSD	MobileNet V2	0.764	0.806	0.811	-	0.300	0.815
Proposed Model (NAS-FPN with Faster R-CNN)	ResNet 50 V1	**0.811**	**0.858**	**0.859**	**-**	**0.300**	**0.862**

The above tables 2 and 3 described the performance of the average precision and average recall of different object detection models based on the Jenna 30 wildflower class. The combination of the proposed NAS-FPN and Faster R-CNN model gives the highest performance with the values of different AP IoU (0.5:0.95, 0.50, and 0.75) of the 30 classes are 0.824, 0.974, and 0.915. Moreover, the values of different AR detections (1, 10, and 100) of the 30 classes are 0.811, 0.858, and 0.859. In the J30 flower dataset, there are no small size flower objects available. Therefore, there is no performance value in the small area column. The values of AP large area are 0.828 and AR large area is 0.862 respectively.

Table. 4. The Performance of Average Precision (AP) for different object detection models for F102 flower class dataset.

Object Detection Model	Backbone Pre-trained model	AP, IoU			AP, Area		
		0.5:0.95	0.50	0.75	Small	Medium	Large
Faster RCNN	ResNet 50	0.232	0.519	0.151	0.055	0.113	0.263
	ResNet 101	0.372	0.723	0.340	0.102	0.144	0.389
	Inception ResNet V2	0.272	0.560	0.231	0.050	0.120	0.307
	Inception V2	0.223	0.504	0.152	0.062	0.096	0.256
	NAS	0.245	0.510	0.190	0.034	0.097	0.284
SSD	MobileNet V2	0.242	0.489	0.206	0.028	0.052	0.259
Proposed Model (NAS-FPN with Faster R-CNN)	ResNet 50 V1	**0.412**	**0.762**	**0.396**	**0.073**	**0.155**	**0.435**

Table. 5. The Performance of Average Recall (AR) for different object detection models for F102 flower class dataset.

Object Detection Model	Backbone Pre-trained model	AR, Detections			AR, Area		
		1	10	100	Small	Medium	Large
Faster RCNN	ResNet 50	0.225	0.414	0.456	0.129	0.302	0.489
	ResNet 101	0.325	0.511	0.545	0.203	0.310	0.563
	Inception ResNet V2	0.243	0.423	0.448	0.095	0.297	0.486
	Inception V2	0.220	0.398	0.434	0.125	0.309	0.473
	NAS	0.244	0.405	0.420	0.076	0.222	0.473
SSD	MobileNet V2	0.280	0.381	0.392	0.087	0.108	0.418
Proposed Model (NAS-FPN with Faster R-CNN)	ResNet 50 V1	**0.345**	**0.523**	**0.558**	**0.169**	**0.314**	**0.580**

The above tables 4 and 5 described the performance of the average precision and average recall of different object detection models based on the 102 flower class wildflower class. Here also, the proposed model i.e. NAS-FPN with Faster R-CNN model gives the highest performance with the values of different AP IoU (0.5:0.95, 0.50, and 0.75) of the 102 classes are 0.412, 0.762, and 0.396. Moreover, the values of different AR detections (1, 10, and 100) of the 102 classes are 0.345, 0.523, and 0.558. The values of AP small, medium, and a large area are 0.073, 0.155,

and 0.435 and for AR small, medium, and a large area are 0.169, 0.314, and 0.580 respectively. The following graphs described in figure 8, 9 and figure 10, 11 described the performance of different models on 102 flower class dataset and 30 flower class dataset respectively. Figure 8 and Figure 10 represented the result of Average Precision (AP), while figure 9 and figure 11 described the result of Average Recall (AR). Here, high precision specifies high correctness of the detection results. Accordingly, high recall indicates that fewer targets are missed during the detection. It is shown that, on both of the datasets, the proposed model has high precision and high recall values among all other object detection models.

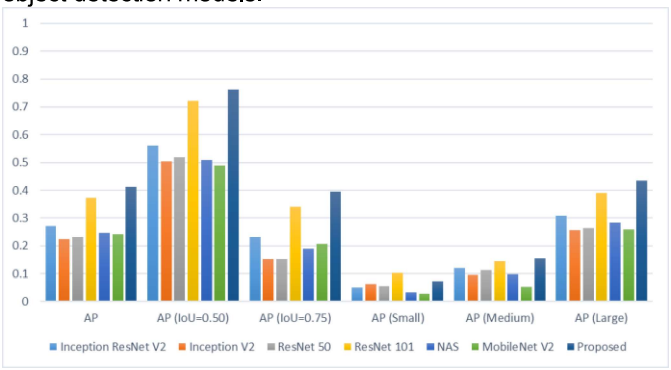

Fig. 8. Average Precision of Object Detection Models on 102 flower class dataset.

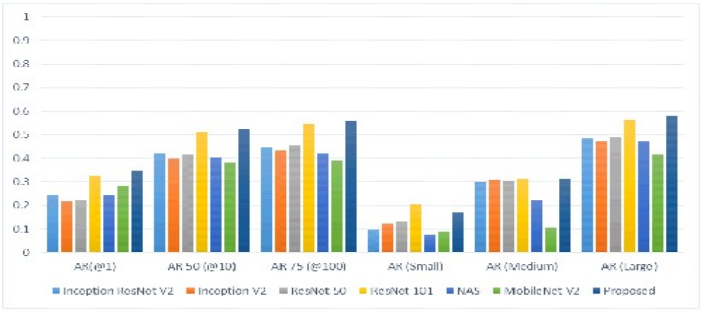

Fig. 9. Average Recall of Object Detection Models on 102 flower class dataset.

The following graphs described in figure 10 and 11 respectively, described the performance of different models on 30 flower class dataset for AP and AR. High precision indicates high correctness of the detection results while high recall reveals fewer targets are missed during the detection.

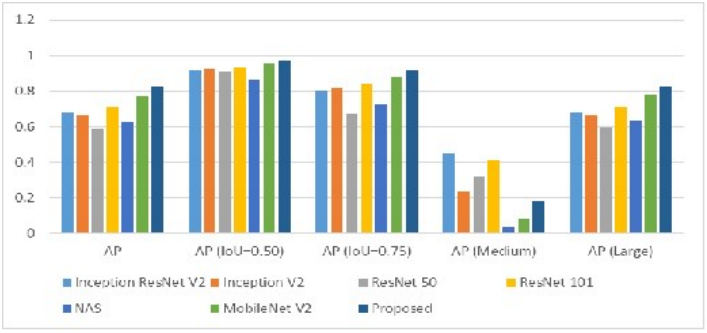

Fig. 10. Average Precision of Object Detection Models on 30 flower class dataset.

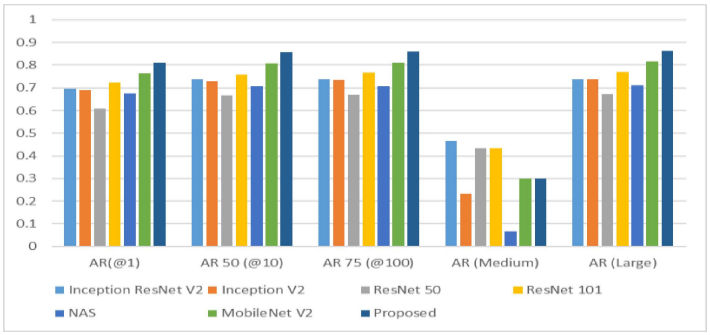

Fig. 11. Average Recall of Object Detection Models on 30 flower class dataset.

The overall detection performance was measured with the mean average precision (mAP) score, which calculated as the average of AP value over all classes. The higher the mAP is, the better the overall detection performance of the model. The values of mAP are plotted in a graph that is include in figure 12. The proposed model is able to achieve the highest accuracy, 87.6% for 102 flower class dataset and 96.2% for 30 flower class dataset, among all other object detection models.

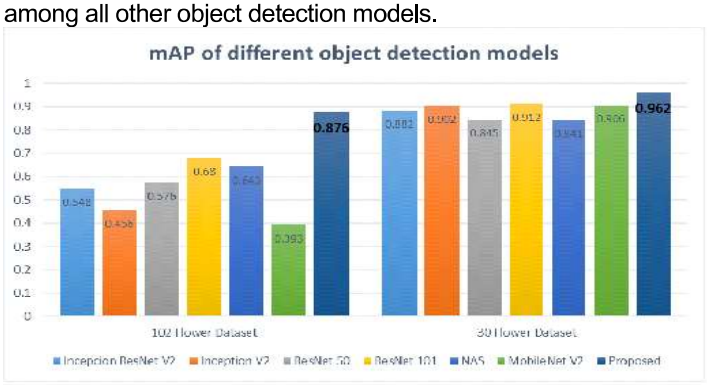

Fig. 12. Accuracy of object detection models on flower datasets.

5.2 Qualitative Analysis for Different Flower Detection Experiment Results

The correctness of a detected object is qualitatively evaluated by the intersection-over-union (IoU) overlap with the corresponding ground truth bounding box. The IoU overlap is defined as following figure 13. The ground truth bounding boxes indicates the hand labeled bounding boxes form the training set that specify where in the image a flower is. And prediction bounding box indicates as output that can be evaluated using IoU (i.e. prediction of object detection model).

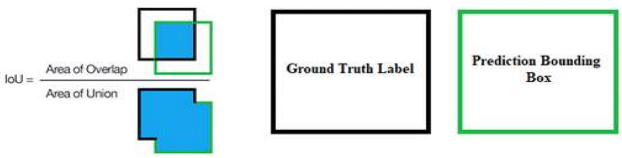

Fig. 13. IoU for flower detection.

The following outcome shows that all the object detection model has performed well on flower dataset, finding all of the flowers dataset 1 and dataset 2 IoU is predict in the following figures. The black box indicates the ground truth label and the colorful box indicates the prediction bounding box. To check the effect of different pre-trained CNN architecture on flower detection performance, we compared the object detection

model results between 30 flower class dataset and 102 flower class dataset. Figures 14 and 15 show some quantitative results for 30 flower class and 102 flower class.

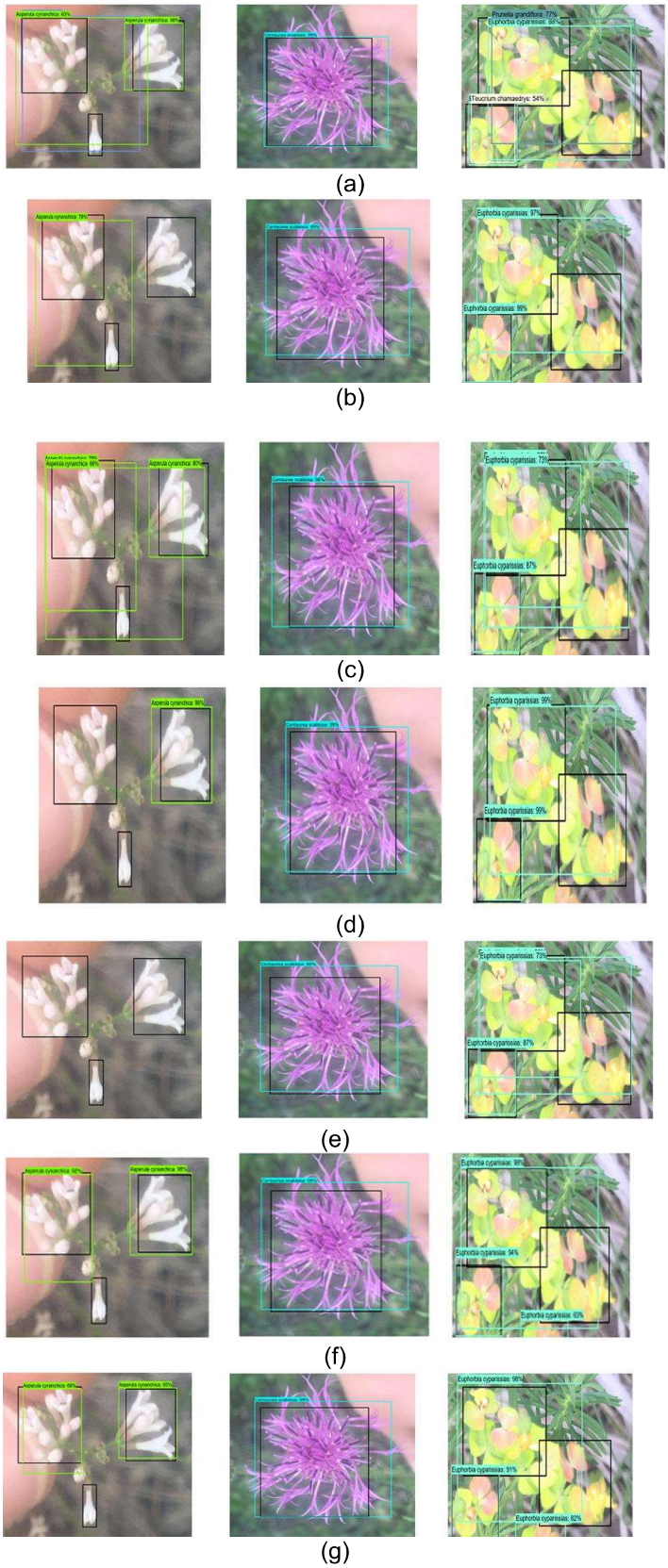

Fig. 14. Performance of Object Detection models for flower30 dataset using Faster RCNN using (a) Inception ResNet V2, (b) Inception V2, (c) ResNet 50, (d) ResNet 101, (e) NAS, (f) SSD using MobileNet V2 and (g) proposed approach using ResNet 50 V1.

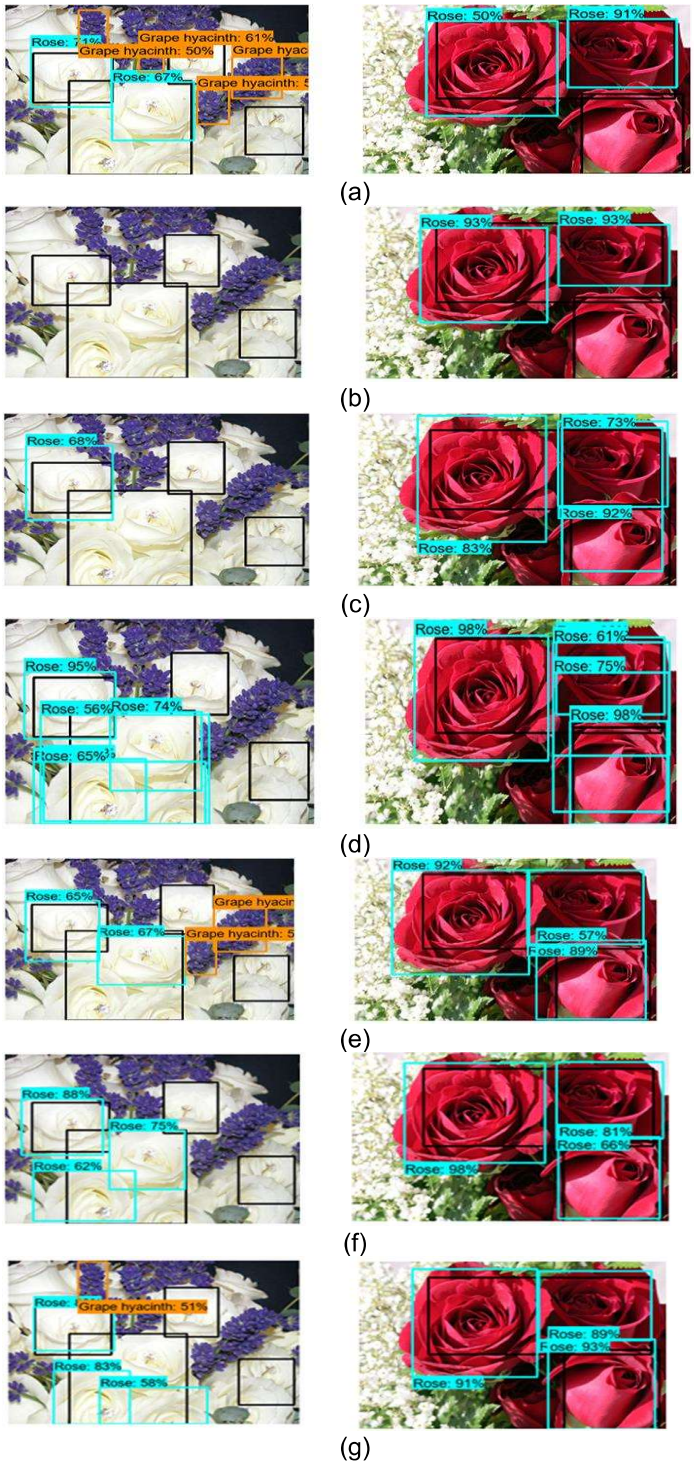

Fig. 15. Performance of Object Detection models for flower102 dataset for Faster RCNN using (a) Inception ResNet V2, (b) Inception V2, (c) ResNet 50, (d) ResNet 101, (e) NAS, (f) SSD using MobileNet V2 and (g) proposed approach using ResNet 50 V1.

The result of qualitative analysis of the object detection models is represented in figure 14 and figure 15 for the flower 30 dataset and flower102 dataset respectively. The black

colored box is represented as Ground Truth label and the blue colored box represented as Prediction bounding box. The value of IoU is also represented along with the Prediction bounding box. Among all the models, the proposed NAS-FPN with Faster R-CNN provides highest accuracy and hence it is an optimized deep learning model for detecting, localizing and classifying the flower objects.

6 Conclusion

In this paper, we proposed an optimized deep convolutional neural network (DCNN) based model for flower object detection, localization and classification. For that, NAS-FPN is integrated with Faster R-CNN and transfer learning based on COCO dataset is used. Flower images are pre-processed using MDSR, Image Annotation and RED-Net architecture. After that, the flower images split into two datasets; one is having 30 flower class and contains 1479 images and another is having 102 flower class and contains 18200 images. NAS-FPN and Faster R-CNN along with other object detection models are evaluated using pre-rained models of COCO dataset. The proposed model provides an optimum result with a mAP of 87.6% on 102 flower class dataset and 96.2% on 30 flower class dataset. Apart from this, a multi-label classification model is built to provide further botanical information of a flower that can help farmers or non-botanical person to recognize the flower class.

Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

[1] Dat Thanh Tran, Toke Thomas Høyey, Moncef Gabbouj, Alexandros Iosifidis, “Automatic Flower and Visitor Detection System”, 26th European Signal Processing Conference (EUSIPCO), 2018.

[2] Wei Qin1, Xue Cui, Chang-An Yuan, Xiao Qin, Li Shang, Zhi-Kai Huang, and Si-Zhe Wan, “Flower Species Recognition System Combining Object Detection and Attention Mechanism”, Springer Nature Switzerland AG 2019.

[3] I.Gogul, V.Sathiesh Kumar, “Flower Species Recognition System using Convolution Neural Networks and Transfer learning”, 4th International Conference on Signal Processing, Communications and Networking (ICSCN -2017), March 16 – 18, 2017.

[4] Yang Yu, Kailiang Zhang, Li Yang, Dongxing Zhang, “Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN”, Science Direct, Computers and Electronics in Agriculture, 163, 2019, pp. 104-846.

[5] Musa Cıbuk, Umit Budak, Yanhui Guo, M. Cevdet Ince, Abdulkadir Sengur, “Efficient deep features selections and classification for flower species recognition”, Science Direct, Measurement 137 (2019) 7–13, pp. 1-8.

[6] Mengxiao Tian, Hong Chen, Qing Wang, "Detection and Recognition of Flower Image Based on SSD network in Video Stream", IOP Conf. Series: Journal of Physics, ICSP 2019.

[7] Swati Kosankar, Dr. Vasima Khan, "Flower Classification using MobileNet: An Optimized Deep Learning Model", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 8, Issue 4, April 2019, pp. 54–60.

[8] Hazem Hiary, Heba Saadeh, Maha Saadeh, Mohammad Yaqub, "Flower Classification using Deep Convolutional Neural Networks", IET (The Institution of Engineering and Technology), 2015, pp. 1–8.

[9] Isha Patel, Sanskruti Patel, "Flower Identification and Classification using Computer Vision and Machine Learning Techniques", International Journal of Engineering and Advanced Technology (IJEAT), Volume-8 Issue-6, August 2019.

[10] Tao Kong, Fuchun Sun, Wenbing Huang, and Huaping Liu, “Deep Feature Pyramid Reconfiguration for Object Detection”, springer, ECCV, 2018.

[11] Sanskruti Patel, Atul Patel, “Deep Leaning Architectures and its Applications: A Survey”, International Journal of Computer Sciences and Engineering Vol.6, Jun 2018.

[12] Zhihao Wang, Jian Chen, Steven C.H. Hoi, “Deep Learning for Image Super-resolution: A Survey”, arXiv: 1902.06068v1, 16 Feb 2019, pp. 1-23.

[13] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, Qingmin Liao, “Deep Learning for Single Image Super-Resolution: A Brief Review”, arXiv: 1808.03344v2, 16 Mar 2019, pp. 1-17.

[14] Khizar Hayat, “Multimedia super-resolution via deep learning: A survey”, Digital Signal Processing, Science Direct, 81 (2018) 198–217.

[15] Jian Lu, Weidong Hu, Yi Sun, “A deep learning method for image super-resolution based on geometric similarity”, Signal Processing: Image Communication, Science Direct, 70, 2019, pp. 210–219.

[16] Bee Lim Sanghyun Son Heewon Kim Seungjun Nah Kyoung Mu Lee, “Enhanced Deep Residual Networks for Single Image Super-Resolution”, arXiv: 1707.02921v1, 10 Jul 2017.

[17] Bokyoon Na, Geoffrey C Fox, “Object Detection by a Super-Resolution Method and a Convolutional Neural Networks”, IEEE International Conference on Big Data (Big Data), 2018, pp. 2252 – 2258.

[18] https://github.com/tzutalin/labelImg

[19] https://towardsdatascience.com/creating-your-own-object-detector-ad69dda69c85

[20] https://towardsdatascience.com/how-to-train-your-own-object-detector-with-tensorflows-object-detector-api-bec72ecfe1d9

[21] Xiao-Jiao Mao, Chunhua Shen, Yu-Bin Yang, “Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections”, arXiv: 1603.09056v2, 1 Sep 2016.

[22] Xiao-Jiao Mao, Chunhua Shen, Yu-Bin Yang, “Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections”, arXiv: 1606.08921v3, 30 Aug 2016.

[23] Yali Peng, Lu Zhang, Shigang Liu, Xiaojun Wu, Yu Zhang, Xili Wang, “Dilated Residual Networks with Symmetric Skip Connection for image denoising”, Science Direct, Neurocomputing, 345, 2019, pp.67–76.

[24] Keiller Nogueira, Otavio A. B. Penatti, Jefersson A. dos Santos, “Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification”, arXiv: 1602.01517v1, 4 Feb 2016, pp. 1-27.

[25] Yin Cui, Yang Song, Chen Sun, Andrew Howard, Serge Belongie, “Large Scale Fine-Grained Categorization and Domain-Specific Transfer Learning”, arXiv:1806.06193v1, 16 Jun 2018, pp. 1-10.

[26] Weifeng Ge Yizhou Yu, "Borrowing Treasures from theWealthy: Deep Transfer Learning through Selective Joint Fine-Tuning", arXiv: 1702.08690v2, 6 Jun 2017.
[27] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, Herv´e J´egou, "Fixing the train-test resolution discrepancy", arXiv: 1906.06423v2 [cs.CV] 19 Jul 2019.
[28] Ioannis Athanasiadis, Panagiotis Mousouliotis, Loukas Petrou, "A Framework of Transfer Learning in Object Detection for Embedded Systems", arXiv:1811.04863v2 [cs.CV] 24 Nov 2018.
[29] Tsung-Yi Lin, Piotr Doll´ar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature Pyramid Networks for Object Detection", IEEE, pp. 2117-2125.
[30] Golnaz Ghaisi Tsung-Yi Lin Ruoming Pang Quoc V. Le, "NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection", arXiv: 1904.07392v1, 16 Apr 2019.
[31] ZHAO Baojun, ZHAO Boya, TANG Linbo, WANG Wenzheng, and WU Chen, "Multi-scale object detection by top-down and bottom-up feature pyramid network", Journal of Systems Engineering and Electronics, Vol. 30, No. 1, February 2019, pp.1 – 12.
[32] Selim Seferbekov, Vladimir Iglovikov, "Feature Pyramid Network for Multi-Class Land Segmentation", IEEE, pp. 272 – 275.
[33] Yinjuan Gu, Bin Wang, Bin Xu, "A FPN-Based Framework for Vehicle Detection in Aerial Images"
[34] https://towardsdatascience.com/review-fpn-feature-pyramid-network-object-detection-262fc7482610
[35] Ross Girshick, "Fast R-CNN", arXiv:1504.08083v1 [cs.CV] 30 Apr 2015.
[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", arXiv: 1506.01497v3, 6 Jan 2016, pp. 1-14.
[37] Zhenwei He, Lei Zhang, "Multi-adversarial Faster-RCNN for Unrestricted Object Detection", arXiv: 1907.10343v1 [cs.CV] 24 Jul 2019, pp. 4321-4330.
[38] Linu Shine, Anitha Edison, Jiji C. V., "A Comparative Study of Faster R-CNN Models for Anomaly Detection in 2019 AI City Challenge", IEEE, pp. 306 – 314.
[39] Yun Ren, Changren Zhu and Shunping Xiao, "Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN", Appl. Sci., www.mdpi.com/journal/applsci, 2018, 8, 813; pp. 1-11.
[40] Arthur Daniel Costea, Andra Petrovai, Sergiu Nedevschi, "Fusion Scheme for Semantic and Instance-level Segmentation", Conference Paper, November 2018.
[41] Xusheng Lei, Zhehao Sui, "Intelligent fault detection of high voltage line based on the Faster R-CNN", Elsevier, 2019, pp. 379-385.
[42] Longsheng Fu, Yali Feng, Yaqoob Majeed, Xin Zhang, Jing Zhang, Manoj Karkee, Qin Zhang, "Kiwifruit detection in field images using Faster R-CNN with ZFNet", Elsevier, 2018, pp. 45-50.
[43] Matthew C. Chan and John P. Stott, "Deep-CEE I: Fishing for Galaxy Clusters with Deep Neural Nets", arXiv: 1906.08784v2 [astro-ph.GA] 25 Jun 2019.
[44] Faming Shao, Xinqing Wang, Fanjie Meng, Jingwei Zhu, Dong Wang and Juying Dai, "Improved Faster R-CNN Traffic Sign Detection Based on a Second Region of Interest and Highly Possible Regions Proposal Network", Sensors 2019, 19, 2288, pp. 1-28.
[45] Gulraiz Khan, Zeeshan Tariq and Muhammad Usman Ghani Khan, "Multi-Person Tracking Based on Faster R-CNN and Deep Appearance Features", Intech open, Visual Object tracking in the Deep Neural Networks Era, pp. 1-24.
[46] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, Kevin Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors", arXiv: 1611.10012v3 [cs.CV] 25 Apr 2017, pp. 1-21.
[47] Yun Ren, Changren Zhu and Shunping Xiao, "Object Detection Based on Fast/Faster RCNN Employing Fully Convolutional Architectures", Hindawi, Mathematical Problems in Engineering, Volume 2018, pp. 1-8.
[48] Shaoming Zhang, RuizeWu, Kunyuan Xu, JianmeiWang and Weiwei Sun, "R-CNN-Based Ship Detection from High Resolution Remote Sensing Imagery", Remote Sens. 2019, 11, 631, pp. 1-15.
[49] Yongcheng Liu, Lu Sheng, Jing Shao, Junjie Yan, Shiming Xiang, Chunhong Pan, "Multi-Label Image Classification via Knowledge Distillation fromWeakly-Supervised Detection", arXiv:1809.05884v2 [cs.CV] 21 Feb 2019.
[50] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, Wei Xu, "CNN-RNN: A Unified Framework for Multi-label Image Classification"
[51] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Doll ´ ar, "Microsoft COCO: Common Objects in Context", arXiv:1405.0312v3, 21 Feb 2015.
[52] http://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html
[53] http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html
[54] https://www.kaggle.com/alxmamaev/flowers-recognition
[55] Marco Seeland, Michael Rzanny, Nedal Alaqraa, Jana WaÈldchen, Patrick MaÈder, "Plant species classification using flower images - A comparative study of local feature representations", PLOS ONE, February 24, 2017, pp. 1-29.
[56] Bernardo Augusto Godinho De Oliveira, Flávia Magalhães Freitas Ferreira, Carlos Augusto Paiva Da Silva Martins, "Fast and Lightweight Object Detection Network: Detection and Recognition on Resource Constrained Devices", IEEE, VOLUME 6, 2018.
[57] Jun Sun, Xiaofei He, Xiao Ge, Xiaohong Wu, Jifeng Shen and Yingying Song, "Detection of Key Organs in Tomato Based on Deep Migration Learning in a Complex Background", www.mdpi.com/journal/agriculture, Agriculture 2018, 8, 196, pp. 1-15.
[58] Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, Wanli Ouyang, "FishNet: A Versatile Backbone for Image, Region, and Pixel Level Prediction", arXiv: 1901.03495v1 [cs.CV] 11 Jan 2019.
[59] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, Qi Tian, "CenterNet: Keypoint Triplets for Object Detection", arXiv:1904.08189v3 [cs.CV] 19 Apr 2019.
[60] Yu Peng Chen, Ying Li, Gang Wang, "An Enhanced Region Proposal Network for object detection using deep learning method", PLOS ONE, September 20, 2018.