

# Cannabis Seed Variant Detection Using Faster R-CNN

Toqi Tahamid Sarker, Taminul Islam, Khaled R Ahmed

*School of Computing*

*Southern Illinois University*

Carbondale, Illinois, USA

{toqitahamid.sarker, taminul.islam, khaled.ahmed}@siu.edu

**Abstract**—Analyzing and detecting cannabis seed variants is crucial for the agriculture industry. It enables precision breeding, allowing cultivators to selectively enhance desirable traits. Accurate identification of seed variants also ensures regulatory compliance, facilitating the cultivation of specific cannabis strains with defined characteristics, ultimately improving agricultural productivity and meeting diverse market demands. This paper presents a study on cannabis seed variant detection by utilizing Faster R-CNN, a two-stage object detection model, to accurately detect and classify 17 distinct classes of cannabis seeds sourced locally in Thailand. We evaluate six Faster R-CNN models by comparing performance across key metrics and achieving a mAP score of 94.08% and an F1 score of 95.66%. To our knowledge, this paper presents the first known application of a deep neural network-based object detection model to the novel task of visually identifying variants of cannabis seeds.

**Index Terms**—Object Detection, Faster R-CNN, Cannabis, Precision Agriculture

## I. INTRODUCTION

Seed detection is vital in agriculture and horticulture to ensure quality, purity, and genetic diversity. In cannabis cultivation, it is particularly important due to the unique characteristics and regulatory restrictions imposed by different countries with this plant. The classification of cannabis into distinct categories, such as marijuana or industrial hemp, is primarily dependent on the threshold level of THC, the psychoactive compound found in the plant. Although 1% THC is typically considered enough to cause intoxication, many jurisdictions establish the legal distinction between marijuana and hemp at the 0.3% THC threshold [1]. The cultivation of hemp is typically permitted only if it contains less THC than this defined threshold. Therefore, accurately identifying cannabis seeds with high or low THC content is essential for regulatory compliance and distinguishing between marijuana and industrial hemp [2].

Moreover, in recent years, Cannabis has seen legalization in many countries in the world. The 2014 Farm Bill [3] legalized hemp pilot programs in the United States for research purposes, enabling the development of a domestic hemp industry after decades of prohibition. In the years since this policy change, it has proven economically beneficial for the United States. As seen in the trade data, US imports of hemp seeds, fibers, oils, and other ingredients have steadily risen, totaling \$67.3 million in 2017 [4]. Such legalization efforts

have suddenly increased the demand for Cannabis for various purposes, such as recreational, medicinal, and industrial uses. In the hemp industry, beyond its medicinal value, hemp plays a pivotal role as a crucial resource for the manufacturing of ropes, textiles, and paper [2]. The presence of undesired seeds can lead to cross-pollination, resulting in undesirable traits in the crop. Thus, identifying and detecting cannabis seeds with different variants ensures uniform product quality and a profitable yield.

Previous studies have applied a range of artificial intelligence methods for different applications in cannabis agriculture. For instance, Sieracka et al. [5] used artificial neural networks to forecast industrial hemp seed yield based on cultivation data. Bicakli et al. [6] demonstrated that random forest models can distinguish illegal cannabis crops from other vegetation in satellite imagery. In another study, Ferentinos et al. [7] introduced a deep learning system using transfer learning to identify diseases, pests, and deficiencies in cannabis plant images. Most recently, Boonsri et al. [8] applied deep learning-based object detection models to detect male and female cannabis seeds from augmented seed image datasets. Despite the progress of deep learning in many agricultural applications, its use for cannabis seed variety detection and classification has garnered less attention. Therefore, the main objective of this paper is to classify and detect seeds of 17 kinds of cannabis.

The rest of the paper is organized as follows. Section 2 explores the related work. Section 3 describes the dataset, dataset preprocessing, training parameters, Faster R-CNN architecture, loss functions, and evaluation metrics. Section 4 presents the experimental results, discusses the findings, and compares the performance of the object detector. Finally, Section 5 summarizes the conclusions from our work.

## II. RELATED WORK

Seed identification has been tackled using traditional methods [9], [10], image processing-based techniques [11], [12], and model-based approaches [13], [14]. Traditional methods include manual inspection and biochemical and molecular identification. The manual inspection methods rely on external characteristics such as color, shape, and size of seeds [15]. However, seeds with similar external features can be challenging to classify accurately [9], [15]. The biochemical seed iden-

tification method can recognize seeds with different genetic characteristics; however, identifying closely related varieties is challenging. The molecular seed identification methods used DNA markers, and they show stability and independence of environmental conditions in identifying seeds. However, their identification process risks damaging the sample and is expensive [10]. Image processing-based techniques include morphological operations, thresholding, segmentation, feature extraction, and texture analysis. Model-based approaches are divided into two categories: traditional machine learning models such as logistic regression, random forest, and decision tree. The second category encompasses deep learning models, including the commonly used convolutional neural networks.

#### A. Image Processing in Seed Classification

Traditional image processing techniques have been widely used for seed detection and analysis in previous research. For example, Ahmad et al. [11] developed a model that distinguishes between asymptomatic and symptomatic soybean seeds based on color features. The methods used include RGB color analysis, thresholding for segmentation, discriminant analysis for feature selection, and linear discriminant classification with unequal priors to identify different seed damage types. Another study [12] proposed high-throughput methods using image analysis to efficiently measure the morphological and color traits of soybean seeds. The performance of traditional image processing pipelines depends heavily on extensive tuning and feature engineering specific to each seed type. They generalize poorly to unseen data and lack the learning capacity to improve with more training data.

#### B. Machine Learning in Seed Classification

In recent years, there has been a lot of research carried out by researchers on the application of machine learning to the identification and classification of seeds and their other traits [16]. Thu Hong et al. [13] introduced an automated classification system for distinguishing rice seed varieties, crucial for ensuring seed purity in rice production. The research employs various feature extraction techniques, including morphological, color, texture, GIST, and SIFT features, and evaluates the performance of machine learning classifiers such as KNN, SVM, and Random Forest. The results demonstrate that the Random Forest method with basic feature extraction achieves the highest 90.54% accuracy in classifying six rice varieties. In another research [14], the authors trained Random Forest, BayesNet, LogitBoost, and Multilayer Perceptron to classify six corn seed varieties. Of all the classifiers, Multilayer Perceptron achieved the highest accuracy of 98.93% on seed images. The effectiveness of supervised machine learning models relies significantly on feature extraction techniques. Inadequate or low-quality features can result in suboptimal performance.

#### C. Deep Learning in Seed Classification

Recent studies have begun leveraging deep learning for automated seed detection and classification. In one study [17],

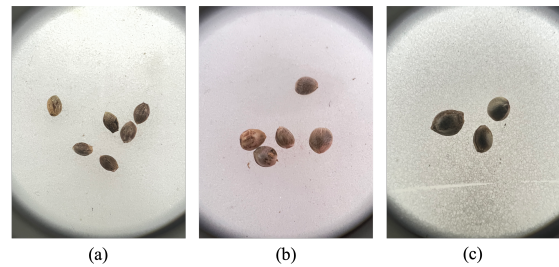


Fig. 1. Three different cannabis seed types. (a) AK47, (b) Gelato, and (c) Gorilla Purple.

researchers used Faster R-CNN with various ResNet backbones to accurately detect, count, and discriminate germinated and non-germinated parasitic plant seeds from images. In a different study [18], researchers introduced a real-time, high-throughput seed sorting system utilizing ResNet-18 image classifier and a batch inference strategy to achieve high speed (500 fps) and high accuracy. This approach separated the object-detection task into localization and classification, demonstrating superior sorting accuracy (99.58%) and purity (99.994%) compared to commercial optical seed sorting systems. While deep learning has shown promising results for seed analysis, there remain considerable challenges. Collecting and annotating thousands of high-quality seed images needed to train a model is labor-intensive, time-consuming, and expensive.

Existing cannabis agriculture research has utilized deep learning to identify grow sites [19] and detect crop issues and pests [20]. However, the ability of the deep learning models to discriminate between cannabis seed varieties remains unexplored. Although deep learning approaches have shown promise for screening the gender of cannabis seeds [8], the capability of these models to discriminate between seed varieties has yet to be explored. Our work seeks to bridge this research gap. In this paper, we use Faster R-CNN to accurately detect and delimit the bounding box region of the cannabis seed varieties. We evaluate the precision, recall, and inference speed of the detector on the cannabis seed dataset to assess their suitability and performance for this agricultural application.

### III. MATERIALS AND METHODS

#### A. Cannabis Seed Dataset

For this research, we select a cannabis seed variety dataset [21] with 17 distinct classes. The seeds are locally available in Thailand. To the best of our knowledge, this is the first open-access dataset of cannabis seeds. The dataset contains 3335 high-resolution images of size 3023 x 4032 pixels, captured with an Apple iPhone 13 Pro. We remove 16 images due to their blurriness, leaving us with 3,319 images for this experiment. The images are taken with a consistent white background but with different lighting conditions and angles. Figure 1 shows the samples of different cannabis seed variants.

### B. Dataset Preprocessing

Labeling each image manually is time-consuming and requires human labor. Instead of manual data annotation, we use Grounding DINO [22], an open-set object detector, to extract bounding boxes from the dataset given an input text query. Open-set object detectors can detect arbitrary object categories rather than being limited to categories it was trained on. Grounding DINO can generate multiple 2D bounding boxes corresponding to an input image and noun phrases associated with a given image-text pair. This allows us to extract object boundaries from the dataset without needing to manually draw boxes around each object instance.

After annotating the dataset, we split the dataset into training, validation, and test sets. We use 1771 images (~53%) to train our object detection models and 723 images (~22%) to validate the model during training time. The rest of the 825 images (~25%) are held out to test the trained model's performance.

Data augmentation is a technique to increase the dataset size when the data is limited. Applying various image transformations helps the model generalize unseen data during testing. We use Albumentations [23] image augmentation library to apply geometric transformations, color transformations, and blur operations on training images. The augmentation parameters are as follows: random flips with 50% probability, maximum image shift of 0.0625, maximum scale change of 0.1, and maximum rotation angle of 45 degrees. The parameters also include randomly changing brightness and contrast within the range of 0.1 to 0.3 with a 20% probability, random shifts to RGB channels by up to 10 intensity levels each, as well as maximum hue, saturation, and value shifts of 20, 30 and 20, respectively, for 10% of the images. Additional augmentation parameters include a 10% probability of applying channel shuffling and a 10% probability of applying either random blur or median blur with a maximum kernel size of 3 during image augmentation.

### C. Training

We use PyTorch [24] and MMDetection [25] to train Faster R-CNN models on an NVIDIA RTX 3090 GPU. We initialize Faster R-CNN with weights from detection models previously trained on the COCO dataset. The models are then fine-tuned with the training dataset for 100 epochs with stochastic gradient descent optimization. A learning rate of 0.02 and 0.01 is used for Faster R-CNN along with a weight decay of 0.0001 and momentum of 0.9. We resize all images to a width of 360 and a height of 640 dimensions to serve as input to the models. The batch sizes for training, validation, and testing are configured as 2, 1, and 1, respectively.

### D. Faster R-CNN Network Architecture

ResNet-50 [26] model pre-trained on ImageNet is used as the backbone of the Faster R-CNN network. The backbone is the initial processing layer in which features are extracted from the input image at several intermediate levels and act as the bottom-up pathway for the feature pyramid network.

Feature pyramid network [27] selects features generated from the last residual blocks from conv2, conv3, conv4, and conv4 of the ResNet 50 backbone to define the pyramid level. Our FPN has four lateral 1x1 convolutions, which merge with the upsampled top-down pathway and reduce the four output feature maps [256, 512, 1024, 2048] from the bottom-up pyramid levels to fixed 256-channel outputs. We apply four 3x3 convolutions, each taking the merged feature map as input and outputting a 4-level feature pyramid, with each level having a 256-channel feature map. Feature maps from the FPN are then forwarded to the Region Proposal Network (RPN) [28]. RPN operates by sliding a small network on the feature maps extracted from the feature pyramid. Our RPN head utilizes a 3x3 convolution with 256 channels followed by ReLU activation to process the 256-channel feature map from the convolutional layer as input. For each sliding window, our anchor generator uses one scale (8) and three aspect ratios (1:2, 1:1, 2:1) to generate regional proposals. Each region proposal is then passed to a 1x1 256-channel convolution layer for classification and another parallel 1x1 12-channel convolution layer to regress the bounding box coordinates. We apply non-maximum suppression (NMS) based on their classification scores to reduce redundancy among region proposals. We set the IoU threshold for NMS to 0.7, retaining a maximum of 1000 region proposals per image and removing boxes with high overlap.

The region proposals are then fed into the Fast R-CNN branch that extracts features using 4 ROI Align [29] layers to output a fixed size 7x7 features for each proposal. The feature vector is fed into two fully connected layers: a softmax layer to predict the class score with cross-entropy loss and regression layers to predict the bounding box location. If the IoU is greater than the IoU threshold value of 0.5, the proposal is identified as a positive result. In the bounding box regressor layer, we use five different loss functions, including L1 loss, IoU loss, Generalized IoU (GIoU) loss [30], Distance IoU (DIoU) loss [31], and Complete IoU (CIoU) loss [32] to minimize the loss for the regression layer.

### E. Loss Functions

The five loss functions we use in the bounding box regression layer of Faster R-CNN are defined as follows:

1) *L1 loss*: L1 loss calculates the absolute difference between the coordinates of the ground truth bounding box and the predicted bounding box. It is defined as,

$$L_{L1} = \frac{\sum_{i=1}^n |B^G - B^P|}{n} \quad (1)$$

From (1), L1 loss is the ratio of the sum of absolute differences between the ground truth bounding box  $B^G$  and the predicted bounding box  $B^P$  divided by the total number of elements  $n$ . The range of L1 loss is  $0 \leq L_{L1} \leq \infty$ .

2) *IoU loss*: Intersection over Union (IoU) is a metric that compares the similarity between two arbitrary shapes. The IoU computation for the ground truth bounding box  $B^G$  and predicted bounding box  $B^P$  is defined as

TABLE I  
MEAN AVERAGE PRECISION, AVERAGE RECALL, F1 AND REAL-TIME INFERENCE RESULTS FOR FASTER R-CNN

Model	mAP @ IoU:0.50:0.95	mAP @ IoU:0.50	Average Recall	F1	Inference Speed (ms)	FPS
mBaseline	0.9168	0.9235	0.951	0.9336	55.6	18
mL1	<b>0.9408</b>	0.9428	<b>0.973</b>	<b>0.9566</b>	59.5	16.8
mIoU	0.9388	<b>0.9431</b>	0.969	0.9537	57.5	17.4
mGIoU	0.9359	0.9396	0.964	0.9497	66.7	15
mDIoU	0.938	0.9428	0.969	0.9532	61	16.4
mCIoU	0.9365	0.9412	0.963	0.9496	57.8	17.3

$$IoU = \frac{|B^G \cap B^P|}{|B^G \cup B^P|} \quad (2)$$

It is the area of the intersection of  $B^G$  and  $B^P$  divided by the union area of  $B^G$  and  $B^P$  defined in (2). Therefore, (3) defines the IoU loss as

$$L_{IoU} = 1 - IoU \quad (3)$$

The range of  $IoU$  is  $0 \leq IoU \leq 1$ , and  $L_{IoU}$  is  $0 \leq L_{IoU} \leq 1$ .

3) *GIoU loss*: IoU loss can not tell if the two bounding boxes are close or far from each other if there is no intersection between the ground truth and predicted bounding boxes. In non-overlapping cases, the gradient of IoU loss becomes zero. However, the GIoU loss [30] solves the vanishing gradient issue for non-overlapping cases by penalizing based on the smallest enclosing box covering both predicted and target boxes, moving the prediction toward the target. The equation for GIoU loss is

$$L_{GIoU} = 1 - GIoU = 1 - IoU + \frac{A^C \setminus (B^G \cup B^P)}{A^C} \quad (4)$$

$A^C$  is the area of the smallest enclosing rectangles of the two bounding boxes in (4), and it acts as a penalty term to move the predicted bounding box closer to the ground truth bounding box. The range of GIoU is  $-1 \leq GIoU(B^G, B^P) \leq 1$ , and the range of  $L_{GIoU}$  is  $0 \leq L_{GIoU} \leq 2$ . Despite resolving the vanishing gradient issue for non-overlapping boxes, GIoU loss converges slowly and struggles with precise bounding box regression.

4) *DIoU loss*: DIoU loss [31] minimizes the distance between the center points of the ground truth bounding box and the predicted bounding box. It is defined as

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(b^p, b^g)}{c^2} \quad (5)$$

From (5),  $b^p$  and  $b^g$  denote the central coordinates of the ground truth and predicted bounding box, with  $c$  measuring the diagonal distance of the smallest box covering both boxes.  $\rho^2(b^p, b^g)$  calculates the Euclidean distance between the central points of  $b^p$  and  $b^g$ . DIoU loss minimizes the distance between the ground truth and predicted bounding box centers, converging faster than GIoU loss.

5) *CIoU loss*: CIoU loss [32] considers the overlap area, central point distance, and aspect ratio for regressing the bounding box coordinates. CIoU loss is based on DIoU loss and adds an aspect ratio parameter defined as

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(b^p, b^g)}{c^2} + \alpha V \quad (6)$$

In (6),  $V$  measures the consistency of the aspect ratio, and  $\alpha$  is the trade-off parameter. When IoU between predicted and ground truth boxes is less than 0.5, CIoU loss reverts to DIoU loss. This behavior reflects that the consistency of the aspect ratio is not essential when there is minimal overlap between two boxes. Conversely, when IoU is greater or equal to 0.5, the consistency of the aspect ratio becomes critical, and CIoU loss refines the localization accuracy through the aspect ratio parameters  $\alpha V$ .

#### F. Evaluation Metrics

We use mean average precision (mAP), recall, and F1 [33] evaluation metrics to evaluate the performance change in our object detection models. mAP is the most commonly used metric to assess the performance of our object detection models. It has three parameters: IoU, precision, and recall. We evaluate our models using a range of IoU thresholds, from 0.50 to 0.95 in 0.05 increments. As the IoU threshold increases, the criterion for considering a predicted bounding box as a true positive becomes stricter and requires a high overlap for a detection to be considered positive. For each IoU threshold, we calculate precision and recall. Precision is the ratio of correctly predicted true positives to the total predicted positives. True Positives are the seeds with an IoU value greater than a given threshold, with correctly identified class labels. Consequently, false positives are seeds with an IoU less than the IoU threshold or the class label of the predicted bounding box is incorrect. Recall is the ratio between the true positive and the total positive samples. Here, false negatives are the seeds that are incorrectly predicted as negatives by the model. Specifically, mAP is calculated by computing the average precision for individual seed classes and averaging the average precision across all classes. Additionally, we report F1 scores. F1 is the harmonic mean of precision and recall. The harmonic mean gives more weight to small values, which means if the recall or precision score is low, it will lower the F1 score.

#### IV. EXPERIMENTAL EVALUATION

This section will present a detailed description of the Faster R-CNN model's performance on the evaluation metrics. We evaluate our experiments with the six different Faster R-CNN models we trained, 1) a baseline model trained with no image augmentation and L1 loss (mBaseline), 2) with image augmentation and L1 loss (mL1), 3) with image augmentation and IoU

TABLE II  
CLASSWISE MAP AT IOU THRESHOLD  $.50 \leq IoU \leq .95$  AND AT IOU THRESHOLD  $.50$

Class	mAP @ IoU:0.50:0.95						mAP @ IoU:0.50					
	mBaseline	mL1	mIoU	mGIoU	mDIoU	mCIoU	mBaseline	mL1	mIoU	mGIoU	mDIoU	mCIoU
AK47	0.984	0.984	0.988	<b>0.989</b>	0.987	0.986	0.985	0.984	0.988	<b>0.989</b>	0.988	0.987
BBA	0.992	0.99	<b>0.993</b>	0.991	0.992	0.99	<b>0.998</b>	0.995	<b>0.998</b>	<b>0.998</b>	<b>0.998</b>	<b>0.998</b>
CP	0.291	<b>0.399</b>	0.366	0.337	0.377	0.352	0.296	<b>0.403</b>	0.369	0.343	0.387	0.355
GELP	0.951	0.956	<b>0.966</b>	0.957	0.963	0.958	0.957	0.956	<b>0.966</b>	0.957	0.964	0.959
GP	0.926	0.972	0.975	0.971	<b>0.976</b>	0.971	0.946	0.978	0.982	0.978	<b>0.984</b>	0.981
HKRKU	0.998	<b>1</b>	<b>1</b>	0.998	0.999	<b>1</b>	<b>1</b>	<b>1</b>	0.999	<b>1</b>	<b>1</b>	<b>1</b>
HKRPPST1	0.934	0.957	0.954	0.954	<b>0.959</b>	0.952	0.949	0.959	0.959	0.956	<b>0.965</b>	0.959
HSSNTT1	0.947	0.949	0.967	<b>0.976</b>	0.968	0.967	0.949	0.949	0.968	<b>0.977</b>	0.969	0.967
KDKT	0.952	<b>0.975</b>	0.97	0.964	0.966	0.971	0.953	<b>0.975</b>	0.972	0.964	0.966	0.971
KD	0.925	<b>1</b>	0.987	<b>1</b>	0.987	0.985	0.933	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
KKV	0.953	<b>0.995</b>	0.989	0.986	0.988	0.987	0.954	<b>0.999</b>	0.994	0.994	0.994	0.993
PD	0.995	<b>0.999</b>	0.995	0.997	0.996	0.996	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
SDA	0.999	<b>1</b>	<b>1</b>	0.998	0.999	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
SKA	0.973	0.976	0.973	0.975	0.974	<b>0.984</b>	0.979	0.977	0.976	0.976	0.977	<b>0.987</b>
TFT	0.987	<b>0.996</b>	0.995	<b>0.996</b>	<b>0.996</b>	0.995	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>1</b>
TKDRD1	0.841	0.869	<b>0.888</b>	0.867	0.866	0.863	0.847	0.873	<b>0.895</b>	0.877	0.871	0.871
TKKWA1	0.938	<b>0.976</b>	0.953	0.955	0.953	0.964	0.954	<b>0.98</b>	0.965	0.966	0.965	0.972

loss (mIoU), 4) with image augmentation and Generalized IoU loss (mGIoU), 5) with image augmentation and Distance IoU loss (mDIoU), and 6) with image augmentation and Complete IoU loss (mCIoU).

Table I gives a comparison of the six Faster R-CNN models, evaluating their performance across various key metrics. The metrics include mAP at different IoU thresholds, specifically from 0.50 to 0.95 in increments of 0.05, mAP at IoU:0.50, Average recall, and F1 score. Our mBaseline model serves as a reference point for evaluating the performance of the other models. The mL1 model demonstrates the highest mAP at IoU:0.50:0.95, with values of 0.9408. Additionally, the mL1 model achieves the highest average recall of 0.973. The mIoU, mGIoU, mDIoU, and mCIoU models exhibit relatively similar performance levels at IoU:0.50. However, the mIoU model demonstrates the highest mAP at IoU threshold 0.50 with a value of 0.9431. The mL1 model achieves the top F1 score of 0.9566, exhibiting strong precision-recall balance. The mIoU model follows closely with a 0.9537 score, while the mBaseline model, without any image augmentation applied during the training, lags significantly at 0.9336 F1 measure. Overall, the mL1 model stands out as the most effective among the listed models, offering the highest mAP, recall, and F1 values, which suggests that mL1 excels in capturing objects with varying levels of spatial overlap.

The classwise mAP@0.50:0.95 and mAP@0.50 of cannabis seed variants are shown in Table II. In terms of the broader IoU range (0.50:0.95), all loss functions perform quite similarly, with mL1 and mIoU slightly outperforming the mGIoU, mDIoU, and mCIoU on average across classes. However, mL1 overall achieves the best performance, with the highest mAP scores for 9 out of the 17 classes. When looking at mAP specifically for an IoU threshold of 0.50, no single loss function dominates the performance across classes. The analysis of the provided tables reveals an interesting pattern regarding the L1 loss function, which consistently yields the highest mAP scores across nine out of the seventeen classes in both IoU threshold scenarios. Moreover, ‘CP’ exhibits the

biggest mAP boost from using L1 loss compared to the mBaseline, increasing mAP by 37.11% at high IoU and by 36.15% at IoU 0.50. This suggests that L1 regression loss provides advantages for improving the localization of this more challenging class.

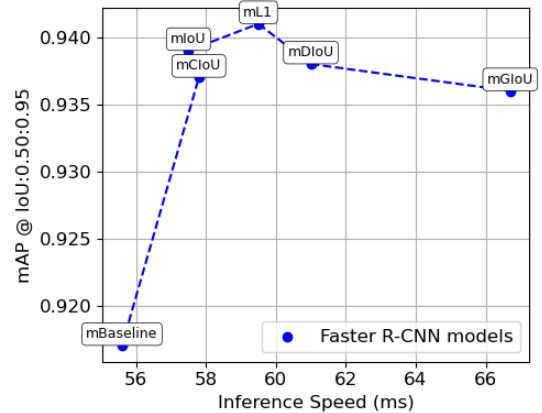


Fig. 2. mAP vs Inference Speed of Faster R-CNN Models

We see a tradeoff between accuracy and speed while comparing the mAP and inference speeds of the Faster R-CNN models in Table I. The best-performing mL1 lags in inference speed by 6.67% compared to the fastest mBaseline model. However, Figure 2 depicts that mIoU strikes the best balance by achieving the second-highest mAP (0.9388) after mL1 and an inference speed of 57.5 ms, which is faster than the top-performing mL1 model. The small compromise in speed in mIoU compared to the fastest mBaseline is justified by sizable gains in detection precision. Therefore, mIoU provides the best tradeoff when both accuracy and real-time performance need to be optimized.

## V. CONCLUSION

This experimental evaluation of Faster R-CNN models for cannabis seed variant detection unveils nuanced insights into their performance across key metrics. The mL1 model emerges

as the most effective, showcasing superior mAP at various IoU thresholds, with a peak value of 0.9408. Notably, mL1 also excels in average recall and achieves the highest F1 score of 0.9566, indicating a commendable balance between precision and recall. While advanced loss functions like Generalized IoU, Distance IoU, and Complete IoU offer marginal improvements, the L1 loss consistently yields the highest mAP scores across a majority of seed classes, revealing its robust performance across evaluation scenarios. Real-time inference performance reveals a trade-off between accuracy and speed, with the mBaseline model being the fastest but less accurate, while mGIoU sacrifices speed for enhanced precision. This analysis provides valuable insights for practitioners, emphasizing the importance of selecting an appropriate loss function based on specific priorities such as accuracy requirements and real-time processing constraints in cannabis seed variant detection applications. In the future, we aim to expand this research and implement a vision transformer model for detecting cannabis seed variants on a more robust dataset.

## REFERENCES

- [1] E. Small and A. Cronquist, "A practical and natural taxonomy for cannabis," *Taxon*, vol. 25, no. 4, pp. 405–435, Aug. 1976.
- [2] J. Cherney and E. Small, "Industrial hemp in north America: Production, politics and potential," *Agronomy (Basel)*, vol. 6, no. 4, p. 58, Nov. 2016.
- [3] "2014 Farm Bill," <https://www.agriculture.senate.gov/issues/farm-bill/>, accessed: 2023-11-24.
- [4] Congressional Service, *Hemp As an Agricultural Commodity*. Createspace Independent Publishing Platform, Jul. 2018.
- [5] D. Sieracka, M. Zaborowicz, and J. Frankowski, "Identification of characteristic parameters in seed yielding of selected varieties of industrial hemp (cannabis sativa L.) using artificial intelligence methods," *Collect. FAO Agric.*, vol. 13, no. 5, p. 1097, May 2023.
- [6] F. Bicakli, G. Kaplan, and A. S. Alqasemi, "Cannabis sativa L. spectral discrimination and classification using satellite imagery and machine learning," *Collect. FAO Agric.*, vol. 12, no. 6, p. 842, Jun. 2022.
- [7] K. P. Ferentinos, M. Barda, and D. Damer, "An image-based deep learning model for cannabis diseases, nutrient deficiencies and pests identification," in *Progress in Artificial Intelligence*, ser. Lecture notes in computer science. Cham: Springer International Publishing, 2019, pp. 134–145.
- [8] P. Boonsri and Y. Limpiyakorn, "Object detection model for gender screening of cannabis seeds," in *Proceedings of the 2023 9th International Conference on Computer Technology Applications*. New York, NY, USA: ACM, May 2023.
- [9] A. A. Aznan, R. Ruslan, I. H. Rukunudin, F. A. Azizan, and A. Y. Hashim, "Rice seed varieties identification based on extracted colour features using image processing and artificial neural network (ANN)," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 7, no. 6, p. 2220, Dec. 2017.
- [10] N. Sharopova, M. D. McMullen, L. Schultz, S. Schroeder, H. Sanchez-Villeda, J. Gardiner, D. Bergstrom, K. Houchins, S. Melia-Hancock, T. Musket, N. Duru, M. Polacco, K. Edwards, T. Ruff, J. C. Register, C. Brouwer, R. Thompson, R. Velasco, E. Chin, M. Lee, W. Woodman-Clikeman, M. J. Long, E. Liscum, K. Cone, G. Davis, and E. H. Coe, Jr, "Development and mapping of SSR markers for maize," *Plant Mol. Biol.*, vol. 48, no. 5-6, pp. 463–481, 2002.
- [11] I. S. Ahmad, J. F. Reid, M. R. Paulsen, and J. B. Sinclair, "Color Classifier for Symptomatic Soybean Seeds Using Image Processing," *Plant Dis.*, vol. 83, no. 4, pp. 320–327, Apr. 1999.
- [12] J. Baek, E. Lee, N. Kim, S. L. Kim, I. Choi, H. Ji, Y. S. Chung, M.-S. Choi, J.-K. Moon, and K.-H. Kim, "High Throughput Phenotyping for Various Traits on Soybean Seeds Using Image Analysis," *Sensors*, vol. 20, no. 1, Jan. 2020.
- [13] P. T. Thu Hong, T. T. Thanh Hai, L. T. Lan, V. T. Hoang, V. Hai, and T. T. Nguyen, "Comparative study on vision based rice seed varieties identification," in *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, Oct. 2015.
- [14] A. Ali, S. Qadri, W. K. Mashwani, S. Brahim Belhaouari, S. Naeem, S. Rafique, F. Jamal, C. Chesneau, and S. Anam, "Machine learning approach for the classification of corn seed using hybrid features," *Int. J. Food Prop.*, vol. 23, no. 1, pp. 1110–1124, Jan. 2020.
- [15] A. C. Martin and W. D. Barkley, *Seed Identification Manual*. Scientific Publishers, 2010.
- [16] L. Zhao, S. M. R. Haque, and R. Wang, "Automated seed identification with computer vision: challenges and opportunities," *Seed Sci. Technol.*, vol. 50, no. 2, pp. 75–102, Oct. 2022.
- [17] J. Braguy, M. Ramazanova, S. Giancola, M. Jamil, B. A. Kountche, R. Zarban, A. Felemban, J. Y. Wang, P.-Y. Lin, I. Haider, M. Zurbriggen, B. Ghanem, and S. Al-Babili, "SeedQuant: a deep learning-based tool for assessing stimulant and inhibitor activity on root parasitic seeds," *Plant Physiol.*, vol. 186, no. 3, pp. 1632–1644, Jul. 2021.
- [18] Y. J. Heo, S. J. Kim, D. Kim, K. Lee, and W. K. Chung, "Super-high-purity seed sorter using low-latency image-recognition based on deep learning," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3035–3042, Oct. 2018.
- [19] F. Bicakli, G. Kaplan, and A. S. Alqasemi, "Cannabis sativa L. Spectral Discrimination and Classification Using Satellite Imagery and Machine Learning," *Agriculture*, vol. 12, no. 6, p. 842, Jun. 2022.
- [20] K. P. Ferentinos, M. Barda, and D. Damer, "An Image-Based Deep Learning Model for Cannabis Diseases, Nutrient Deficiencies and Pests Identification," in *Progress in Artificial Intelligence*, ser. Lecture Notes in Computer Science, P. Moura Oliveira, P. Novais, and L. P. Reis, Eds. Cham: Springer International Publishing, 2019, pp. 134–145.
- [21] P. Chumchu and K. Patil, "Dataset of cannabis seeds for machine learning applications," *Data Brief*, vol. 47, p. 108954, Apr. 2023.
- [22] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.
- [23] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.
- [24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
- [25] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv preprint arXiv:1512.03385*, 2015.
- [27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *arXiv preprint arXiv:1612.03144*, 2016.
- [28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [29] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv preprint arXiv:1703.06870*, 2017.
- [30] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," *arXiv preprint arXiv:1902.09630*, 2019.
- [31] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," *arXiv preprint arXiv:1911.08287*, 2019.
- [32] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation," *arXiv preprint arXiv:2005.03572*, 2020.
- [33] K. R. Ahmed, "DSTEELNet: A Real-Time Parallel Dilated CNN with Atrous Spatial Pyramid Pooling for Detecting and Classifying Defects in Surface Steel Strips," *Sensors*, vol. 23, no. 1, p. 544, Jan. 2023.