




Article

Crop Disease Diagnosis with Deep Learning-Based Image Captioning and Object Detection

Dong In Lee ^{1,†} , Ji Hwan Lee ^{2,†} , Seung Ho Jang ^{3,†} , Se Jong Oh ^{4,*}  and Ill Chul Doo ^{4,*} 

¹ Computer and Electronic Systems Engineering, Hankuk University of Foreign Studies, Yongin 17035, Republic of Korea

² Artificial Intelligence Convergence, Hankuk University of Foreign Studies, Yongin 17035, Republic of Korea

³ Statistics, Hankuk University of Foreign Studies, Yongin 17035, Republic of Korea

⁴ Artificial Intelligence Education, Hankuk University of Foreign Studies, Yongin 17035, Republic of Korea

* Correspondence: tbells@hufs.ac.kr (S.J.O.); dic@hufs.ac.kr (I.C.D.)

† These authors contributed equally to this work.

Abstract: The number of people participating in urban farming and its market size have been increasing recently. However, the technologies that assist the novice farmers are still limited. There are several previously researched deep learning-based crop disease diagnosis solutions. However, these techniques only focus on CNN-based disease detection and do not explain the characteristics of disease symptoms based on severity. In order to prevent the spread of diseases in crops, it is important to identify the characteristics of these disease symptoms in advance and cope with them as soon as possible. Therefore, we propose an improved crop disease diagnosis solution which can give practical help to novice farmers. The proposed solution consists of two representative deep learning-based methods: Image Captioning and Object Detection. The Image Captioning model describes prominent symptoms of the disease, according to severity in detail, by generating diagnostic sentences which are grammatically correct and semantically comprehensible, along with presenting the accurate name of it. Meanwhile, the Object Detection model detects the infected area to help farmers recognize which part is damaged and assure them of the accuracy of the diagnosis sentence generated by the Image Captioning model. The Image Captioning model in the proposed solution employs the InceptionV3 model as an encoder and the Transformer model as a decoder, while the Object Detection model of the proposed solution employs the YOLOv5 model. The average BLEU score of the Image Captioning model is 64.96%, which can be considered to have high performance of sentence generation and, meanwhile, the mAP50 for the Object Detection model is 0.382, which requires further improvement. Those results indicate that the proposed solution allows the precise and elaborate information of the crop diseases, thereby increasing the overall reliability of the diagnosis.

Keywords: crop diseases diagnosis; farm-tech; deep learning; Inceptionv3; transformer; image captioning; YOLOv5; object detection



Citation: Lee, D.I.; Lee, J.H.; Jang, S.H.; Oh, S.J.; Doo, I.C. Crop Disease Diagnosis with Deep Learning-Based Image Captioning and Object Detection. *Appl. Sci.* **2023**, *13*, 3148. <https://doi.org/10.3390/app13053148>

Academic Editor: José Salvador Sánchez Garreta

Received: 27 January 2023

Revised: 20 February 2023

Accepted: 22 February 2023

Published: 28 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The unique form of farming that has taken place inside or near the city rather than rural areas, or urban farming, is growing at a rapid rate. Its market share has also shown a steady increase. The lack of knowledge and inexperience, however, makes the novice farmers face challenges since the crops are vulnerable to climate changes that only the professional farmers can properly manage. There have been a number of attempts to detect crop diseases using cutting-edge techniques such as CNN [1]. However, those types of models show whether the plants were infected or not rather than giving the specific details such as the cause, the area, or the gravity of the diseases.

Therefore, our research aims to develop a solution that can help the novice urban farmers who are struggling with maintaining their crops. Our proposed solution was designed to show the infected areas and diagnose the diseases with detailed explanations.

Therefore, we employed the Image Captioning model that requires image and sentence inputs to generate the captions which play a key role in diagnosing diseases. Moreover, we also utilize the YOLOv5-based Object Detection model that shows bounding boxes to indicate the damaged area.

Preventing the spread of diseases in crops is indispensable for farmers, and therefore, it is crucial to identify the characteristics of diseases and handle them in advance. However, most urban farmers are struggling with them since they are not familiar with farming. We expect our improved solution suggested in this paper can give helpful assistance to those who are not familiar with agriculture.

We expected that our research model could contribute to boost the reliability and accuracy of detecting the crop diseases by presenting sentences of its diagnosis based on our Image Captioning model and boxing the damaged areas without the Object Detection model. We also anticipated that our model could help existing urban farmers with a lack of farming knowledge.

Our research was conducted as follows. In Section 2, we presented background and the related work in Image Captioning and Object Detection. We also introduced data collection and preprocessing work along with the explanation of our two models in Section 3. We described the output of our model test by illustrating the tables and the plots of the qualitative and quantitative results in Section 4. We concluded our paper with an overall summary of our research in Section 5.

2. Related Work

2.1. History of Deep Learning-Based Image Captioning and Object Detection

Research using deep learning has been conducted in various fields. Among them, Image Captioning and Object Detection model research using deep learning are also being actively conducted.

Image Captioning is a technique that captures various features of images and generates sentences describing them in detail. It should accurately grasp the various features contained in the image, such as people, objects, and actions, and generate high-quality sentences that explain these features well. A representative Image Captioning technique using deep learning is the 'End-to-End' method. It is a method built on the Encoder-Decoder structure, which originated with the field of machine translation [2]. The End-to-End method typically consists of a CNN-based encoder that processes features of the image and a RNN-based decoder that generates captions which describe features of the image. For example, there is a model which was designed to extract the features of the image from the CNN model of the encoder, compress the extracted features into a 'global visual feature vector', and pass it on to the RNN model of the decoder to generate sentences [3]. In addition, a model with "Attention", a technique for finding parts to focus on among various features, was devised to enhance the accuracy of image feature description [4]. Recently, models developed based on Attention, such as Transformer, have also been applied to the decoder of the End-to-End method [5]. The Image Captioning model used in this study was also applied with the End-to-End technique that applied the Transformer model to the decoder.

Object Detection is a technique that detects a specific object, such as a person, from an image to classify the class of the object and generates a bounding box that distinguishes the object from others [6]. With the advent of the deep CNN algorithm, the performance of feature extraction of images has increased dramatically. Research on deep learning-based Object Detection models has become active as well [7]. Deep learning-based Object Detection algorithms are largely divided into two types. First, there is a type that separates the process of detecting and classifying an object. For example, The RCNN model proposed by Girschick et al. (2014) finds multiple candidate regions that are expected to be objects, converts the sizes of each candidate region to the same, inputs candidate regions into the CNN model to extract features, and passes each feature to the SVM model to classify the object category [8]. Second, there is a type that processes detecting and classifying an ob-

ject at once. This type of model has the advantage of relatively fast image processing since it detects and classifies an object simultaneously. For Instance, The YOLO model proposed by Joseph et al. (2015) is one of the representatives of this type. As shown in Figure 1, it selects candidate regions separately from images, inputs the entire image into the neural network instead of inputting each candidate region, divides the image into multiple grid cells, and performs detecting and classifying objects simultaneously using each cell [9].

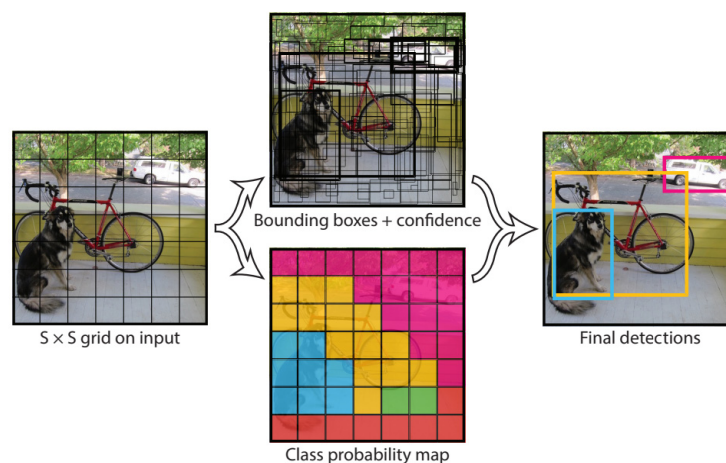


Figure 1. Structure of the YOLO.

Even after the YOLO was released, other YOLO models with improved performance were steadily released. YOLOv5, the fifth-updated YOLO model, was applied to the Object Detection model used in this study.

2.2. Model Application Case

There were only a few number of studies related to Korean context on Image Captioning. Those studies translate the English captions of the MS COCO dataset into Korean using RNNs and LSTMs [10,11]. Moreover, most of them employ old-fashioned models with poor accuracy in the field of Natural Language Processing. Although several models have started to conduct research using Attention, we have rarely found research that handles Korean context using Transformer.

In the non-agricultural field, several studies using Object Detection models have been conducted. There are examples of implementing dirt and garbage detection models using SSDs and VGG16, and there are also studies that have constructed a total of three CNN models—InceptionV3, VGG16, MobileNet—to determine whether people are wearing a mask at the time when COVID-19 was spread worldwide [12].

There were several cases in which an Object Detection model was constructed using a deep learning model in the agricultural field. Leveraging Plant village data, AlexNet and GoogleNet-based crop disease detection cases exist [13]. In addition, there are Fast R-CNN, R-FCN, SDD-based tomato disease, and pest detection models [14]. Meanwhile, using the In Yolo County California, 2014 dataset, the crop classification model combining the enhanced LSTM model and the 1D CNN model yielded higher accuracy of 85.54% and f1_score results of 0.73 compared to existing machine learning models such as XGBoost, RF, and SVM [15]. In addition, there is also a case of improving the apple leaf disease detection model through deep residual learning [16]. There was also a study of recognizing plant leaf diseases by segmenting disease symptoms through a hue, saturation and intensity-based, and LAB-based hybrid segmentation algorithm, as well as by classifying the name of diseases through CNN network [17]. Moreover, we find transfer-learning-based plant disease detection studies that detect rice plant diseases based on the DenseNet pre-trained on ImageNet and the Inception module [18]. There is also a case of identifying plant diseases based on the VGGNet pre-trained on ImageNet and the Inception module [19]. Meanwhile, we also find a case of identifying plant diseases based on ensemble

learning, aggregating three lightweight CNNs, including SE-MobileNet, Mobile-DANet, and MobileNet V2 [20].

There were also cases in which Object Detection research was conducted using the YOLO model. There existed a YOLOv3-based apple growth phase prediction model utilizing the Author Collection Dataset, which led to performance above the Fast R-CNN model and the VGG16 model. There was also a study in which the YOLOv5 model used in this work is applied in conjunction with the optical flow algorithm to develop a tram-to-pedestrian collision prediction prevention model [21].

Therefore, through this study, we used a Transformer model for Image Captioning technology to apply it to Korean, which has not yet been studied, to help agricultural fields and diagnose crop diseases. We also wanted to study it because there were few cases of crop disease diagnostic studies using Image Captioning models.

In addition, we used YOLOv5 as our Object Detection model to visualize the affected area of the crop's disease with bounding boxes so that the damaged part can be seen at a glance.

3. Materials and Methods

3.1. Data Collection

We collected a crop image dataset for model training from AI-hub, a Korean platform that discloses open-source data needed for developing AI-based technology, products, and services [22]. As for crop image data, the "Facility Crop Disease Diagnostic Image" dataset [23], which is the main field crop (10 types) disease image data, and the "Outdoor Crop Disease Diagnostic Image" dataset [24], which is the main facility horticultural crop disease image data (12 types), were used for the disease diagnosis of crops. Both datasets contain images of normal and disease-infected crops. We selected five crops; "pepper", "pumpkin", "tomato", "bean", and "spring onion", which are common in Korean households.

Using the annotation of metadata contained in AI-hub, the crop image data was used to detect normal crops and nine diseases with "Pepper Anthracnose", "Pepper White Powder", "Pumpkin Old Disease", "Pumpkin White Powder Disease", "Tomato Leaf Mold Disease", "Soybean Spot Disease", and "Black Bottle". As shown in Figure 2a, crop images appropriate for model training could be well collected.

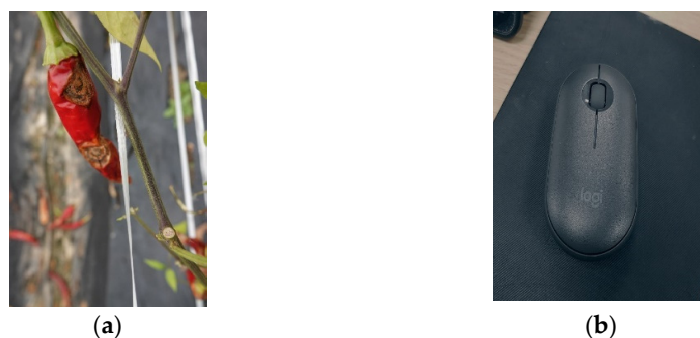


Figure 2. (a) Red Pepper with Anthracnose; (b) Non-Crop Image.

Meanwhile, to prevent disease from being detected in non-crop objects, we had to secure non-crop images for model training so that the model could learn there are no diseases in the non-crop image. We collected non-crop image datasets by randomly crawling non-crop images on websites such as Coupang, Danawa, Lotte Home Shopping, Monthly Airlines, JoongAng Ilbo, Dong-A Ilbo, and Chosun Ilbo. We also took pictures of objects, such as cups and pencils, which can be commonly encountered in everyday life. As shown in Figure 2b, non-crop images could be well secured.

3.2. Data Preprocessing

3.2.1. Data Preprocessing for Image Captioning

In order for the Image Captioning model to generate detailed and accurate diagnostic sentences in images of disease-infected crops, appropriate diagnostic sentences which comprehensively take into account crop types, disease types, disease damage levels, and disease symptoms must be produced. To this end, websites of both AI-Hub and the National Crop Pest Management System were referred to in order to identify types of crops, diseases, disease damage, and disease symptoms, and a crop disease diagnosis sentence necessary for model training could be produced. The process of producing diagnostic sentences for each image, used in model training, is as follows.

Referring to Metadata in AI-Hub

AI-Hub also provides metadata about additional information of the crop images, as shown in Table 1.

Table 1. Examples of Metadata Information Provided by AI-Hub.

Item	Type	Code	Example of Code
Disease	Integer	0, 1, . . . , 20	14
Crop	Integer	0, 1, . . . , 10	9
Area	Integer	0, 1, . . . , 7	3
Grow	Integer	11, 12, 13	12
Risk	Integer	0, 1, 2, 3	2
Bounding Box Points	Dictionary	{x1l, y1l, x1r, y1r}	{"x1l": 100, "y1l": 200, "x1r": 1100, "y1r": 1200}

The metadata of the image is provided as a file in JSON form, as shown in Figure 3. Among the various categories that make up the metadata, the type of disease was identified in the 'disease' part, the type of disease-infected crops in the 'crop' part, and the degree of disease damage, such as early, mid, late stage, in the 'risk' part.

```

"annotations": {
  "disease": 1,
  "crop": 1,
  "area": 1,
  "grow": 13,
  "risk": 1,
  "points": [
    {
      "x1l": 2322,
      "y1l": 480,
      "x1r": 3382,
      "y1r": 3627
    }
  ]
}

```

Figure 3. Metadata in JSON form provided to each image.

Referring to Disease Keywords Provided by National Crop Pest Management System

National Crop Pest Management System provides various information to prevent diseases of crops, such as the type of disease of crops and the timing of disease outbreaks [25]. Particularly, some information about a specific disease can be found there, which contains the types of crops that can be infected with the disease and various symptoms which can be identified from the infected crops. In the information describing the symptoms, the features that appear prominently as symptoms of the disease were identified, and these features were regarded as keywords for symptoms of the disease. For Example, if 'Anthracnose', one of the names of diseases which can infect crops, is searched, several crops that

can be infected with the disease, such as peppers, are introduced. Among them, “round spots”, “dark or yellowish-brown spores”, and “dry twisting” can be identified as keywords for the symptoms of the disease.

Using All the Information Gathered to Produce Crop Disease Diagnostic Sentences

First of all, disease symptom keywords were allocated according to the type of disease and the degree of damage previously identified. As a result of comparing the images of various crops and the symptom keywords of the disease in which the crop was infected according to the degree of disease damage, it was confirmed that the number of keywords appearing in the crop image increased as the disease damage intensified. Reflecting this point, more symptom keywords were assigned to disease diagnosis sentences as the degree of disease damage in the crop image increased. For example, keywords according to the degree of disease damage of pepper anthracnose are as follows.

- Early Stage: “round spots”
- Middle Stage: “round spots” + “dark or yellowish-brown spores”
- Late Stage: “round spots” + “dark or yellowish brown spores” + “dry twisting”

Next, a disease diagnosis sentence including both symptoms and types of disease was produced using keywords allocated according to the degree of disease damage. For example, the disease diagnosis sentence for pepper anthracnose is as follows.

- Early Stage: A round spot appears on the pepper, which is suspected to be a red pepper anthracnose.
- Middle Stage: The pepper has yellowish brown spores and round spots, so it is suspected to be a pepper anthracnose.
- Late Stage: It is suspected that it is a pepper anthracnose, as it appears to have circular spots, yellow-brown spores, and dry twist on the pepper.

Sentence Augmentation for Model Training

Finally, two text augmentation techniques, ‘word order change’ and ‘back translation’, were used to produce additional synonymous sentences to enable the model to learn sentences in various contexts. Word order change is a technique that changes the structure of a sentence by changing the order of sentence components. In Korean, the overall meaning of the sentence is generally maintained even if the order of the sentence components is changed. Using the characteristics of Korean, which has relatively flexible word order [26], various forms of sentences were produced by changing the positions of various sentence components that make up the sentence. Next, back translation is a technique that uses a machine translator to translate a sentence into another language and, then, translates it back into the language before it was translated to change the form of the sentence [27]. Many IT companies, such as Google and Naver, are providing machine translation services. As a result of directly comparing the quality of various machine translation services, the machine translation service called “Papago” provided by Naver tended to be more natural, in terms of foreign to Korean translation, than other services. Therefore, using Naver’s ‘Papago’, an original diagnostic sentence was translated from Korean to English, translated from English to Japanese, and then translated from Japanese to Korean again to produce a new sentence.

For model training, basically 5 disease diagnosis sentences were produced per image using disease symptom keywords, and then, word order change and back translation techniques were applied to the 5 basically produced sentences, respectively, to produce a total of 10 new sentences. In addition, 5 additional sentences were produced by applying back translation techniques to the existing 5 sentences produced by applying word order change. In this way, a total of 20 candidate sentences were produced according to the disease and symptoms of one crop, and 5 non-overlapping sentences were assigned out of 20 candidate sentences per image. In the same way, sentences for normal crops and non-crops were generated and assigned to each image. The number of images used for model training was 123,913. As a result, a total of 619,565 sentences were produced for 123,913 images. On the other hand, the number of images used for model testing was 303,

collected from sample images provided by AI-hub's "Facility Crop Disease Diagnostic Image" dataset and "Outdoor Crop Disease Diagnostic Image" dataset, along with non-crop images collected by crawling websites and taking them in person. Thus, a total of 1515 sentences were produced for 303 images.

The overall preprocessing step of Image Captioning is shown in the Figure 4.

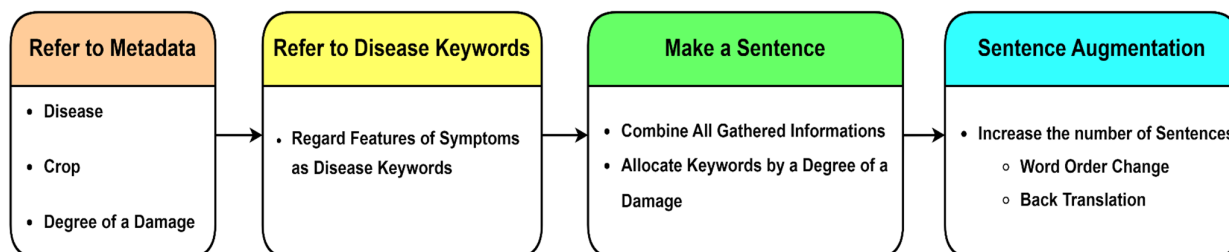


Figure 4. The Overall Preprocessing Step of Image Captioning.

3.2.2. Preprocessing for Object Detection Model

We also collected crop images from AI-hub's "Outdoor Crop Disease Diagnostic Image" and "Facility Crop Disease Diagnostic Image" datasets to train the Object Detection model. We classified seven predicted classes, which was different from the Image Captioning model that defines nine predicted classes since Anthracnose and Leaf Blight were shown in two types of plants.

We labelled images after the data collecting and class defining process. We used 'LabelImg', a python-based program that can create bounding boxes of the images, to box the damaged area. Bounding boxes were based on metadata in AI-hub, which describes the name and types of diseases. We generated boxes where crops were affected with diseases. There were also a number of cases that box the entire part when the infected areas were widely spread on the crop.

The total number of the original dataset from AI-hub is 5970, and among them, we split the entire dataset into a training set and a validation set given the ratio 8:2. The specific figure is as follows: the red pepper data with 1507 images for training and 377 images for validation, the tomato data with 850 images for training and 213 images for validation, the spring onion data with 928 images for training and 112 images for validation, the bean data with 928 images for training and 201 images for testing, and zucchini data with 784 images for training and 196 images for validation. We collected the total 4871 images for training and 1099 images for validation.

We decided to augment image data since the original dataset is not enough to train the proper model with each label not uniformly distributed. What our team had to do before the augmentation process is make a module to resize the bounding boxes based on the augmentation, since the existing tools offer augmentation only for images. The process includes transferring the coordinates from PascalVOC type to Yolo type for model training.

The augmenting options are as follows: Vertical Flip, Horizontal Flip, Linear Contrast, Grayscale, Superpixel, Affine transformation, Embossing, and Sharpening. The Grayscale option was triggered with a 20% chance while other options were with a 50% chance as a way to guarantee the randomness and maximize the diversity of augmented images. We also resized the image to 640×640 to reduce the training time of the model.

The total number of image datasets after augmentation were 31,394 with 25,458 images for training and 5936 images for validation. The specific figures were as follows: the red pepper data with 4521 images for training and 1131 images for validation, the tomato data with 3196 images for training and 952 images for validation, the spring onion data with 5950 images for training and 1278 images for validation, the bean data with 3208 images for training and 928 images for validation, and the zucchini data with 3712 images for training and 804 images for validation.

3.3. Structure of the Crop Disease Diagnosis Solution

3.3.1. Image Captioning Model

The Image Captioning model used in this study is based on the End-to-End structure, which is one of the machine translation models in natural language processing. Image features are extracted from the encoder, and caption sentences are generated in the decoder based on the extracted features. In this study, InceptionV3, a CNN-based model pretrained with ‘ImageNet’ data, was used as an encoder, and Transformer was utilized as a decoder.

(a) InceptionV3

The InceptionV3 is a CNN-based image recognition model that showed a greater performance than VGGNET, commonly known as ILSVRC14 (ImageNet Large-Scale Visual Recognition Challenge 2014) [28]. GoogleNet, developed by researchers at Google, is InceptionV1, an earlier version of InceptionV3 [29].

Thanks to the Inception Module, InceptionV3 was able to be trained with 48 deep neural networks for high accuracy, as shown in Figure 5. As illustrated in Figure 6, the Inception Module performs four types of convolution and pooling operations on the input values, and it combines the results in the channel direction. The Inception V3 model has nine Inception Modules.

We perform various dimensions of convolution operations in parallel to extract the feature map from the Inception Module in a more effective way. In the Inception Module, the 1×1 convolution layer plays a key role in reducing dimensions and reducing operations.

The InceptionV3 model was improved by changing the optimizer to RMSProp from the InceptionV2 by applying the Label Smoothing technique, which prevents overfitting and Batch Normalization (BN) in the fully connected layers.

Those techniques allow InceptionV3 to reduce the complexity while operating with fewer number of parameters, thereby improving accuracy in a shorter period of time.

In this study, we removed the SoftMax layer, the last layer of the InceptionV3 model, in order to extract certain feature points of the input images and use them as other inputs for the Transformer.

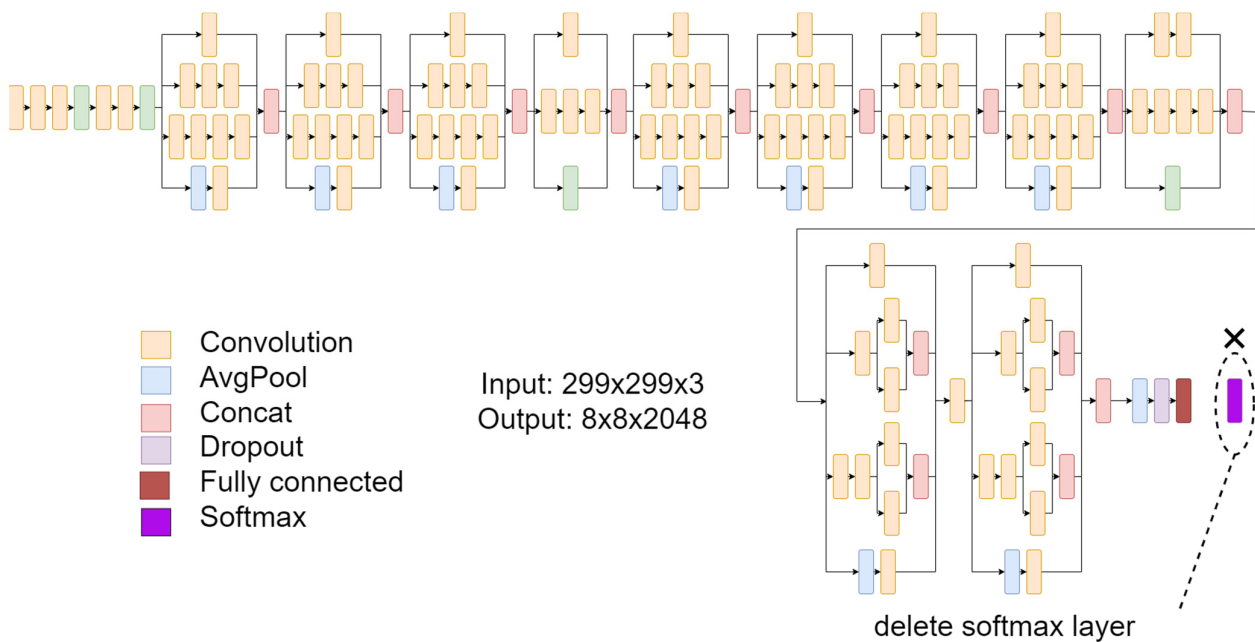


Figure 5. Structure of InceptionV3.

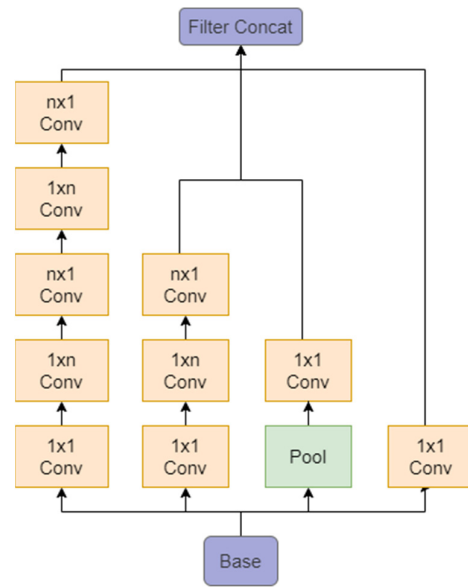


Figure 6. Structure of Inception Modules.

(b) Transformer

Transformer is a model that adopts the Attention mechanism from seq2seq’s Encoder-Decoder structure. This model shows better performance than RNN without using RNN. RNN-based seq2seq models squeeze the input sequence into a single context vector, and the decoder generates an output sequence through this context vector. However, this structure loses some information of the input sequence while the encoder compresses the input sequence into a context vector. Therefore, Attention was used to correct it [30]. Encoder-Decoder structure with Attention allows Transformer to achieve better parallel processing and thus make the model faster than RNN [31].

The Transformer model obtains the positional information of the input sequence through positional encoding. Existing RNNs allow input values to be entered in sequential order, but Transformers use a positive encoding technique and, therefore, cannot be entered in sequential order. As shown in Figure 7, Transformer also utilizes three attentions, which are Self-Attention in encoders, Masked Self-Attention in decoders, and a vanilla Attention layer.

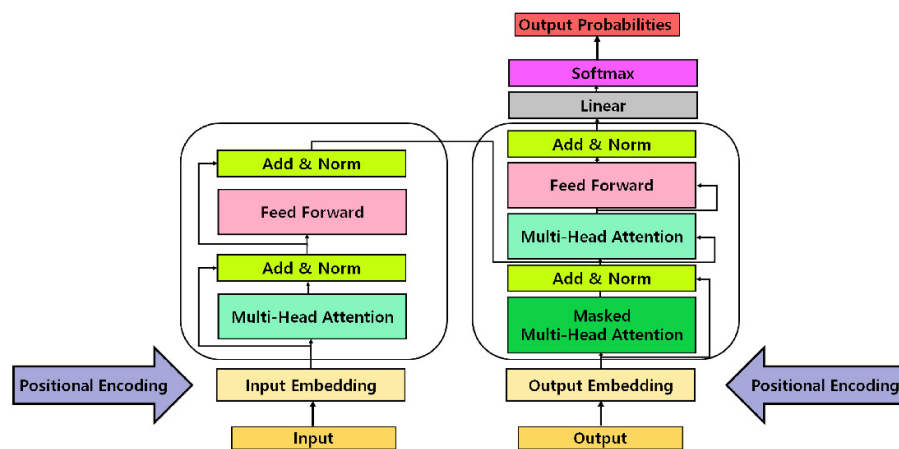


Figure 7. Structure of Transformer Model.

In this study, the feature points of the image extracted through the InceptionV3 model can also be used as an input value of the transformer.

(c) InceptionV3-Transformer

The process of creating captions for images is shown in Figure 8. First, when the image goes as an input into InceptionV3, various features are analyzed in the image. For example, if an image of a red pepper infected with Anthracnose is loaded as an input, the model analyzes the shape, the number, and the color of the affected area. Features extracted from InceptionV3 become an input into the Decoder of Transformer to generate captions. Meanwhile, the Transformer model itself has an Encoder-Decoder structure. First, the features of the image were input into Transformer’s encoder, along with the location information of each feature, and then analyzed through the Self-Attention process. The analyzed information is then entered into Transformer’s decoder, which also includes a pre-made disease diagnostic sentence for model training. Finally, features of the image and disease diagnostic sentences are comprehensively analyzed by the Self-Attention process in the decoder of Transformer, resulting in generated captions predicted by the model.

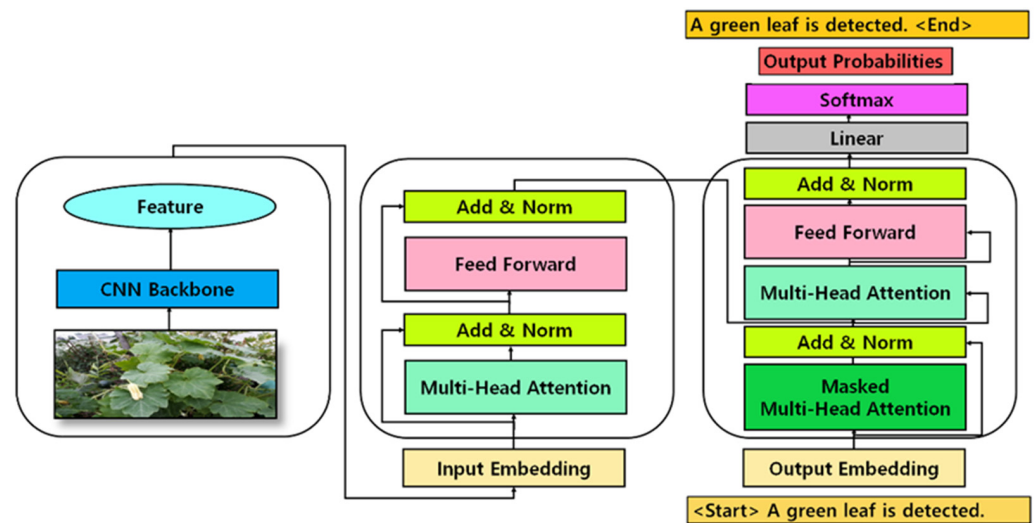


Figure 8. Structure of Image Captioning Model.

3.3.2. Object Detection Model

(a) YOLOv5

There have been a number of models dedicated to the Object Detection, and we decided to use a pretrained model to reduce the training time. Among them, our model employed the YOLO algorithm released in 2016. YOLO algorithm outperforms its competitors, such as Fast R-CNN, in accuracy and speed. We use the YOLOv5 model among other YOLO family models since YOLOv5 showed improved processing time when in deeper networks. Our Object Detection model employed YOLOv5m, one of the subvariants of YOLOv5, considering our situation.

YOLOv5 made features from input images, and the features passed into a prediction system to create a rectangle to form a boundary around the detecting object for prediction using class labels.

As illustrated in Figure 9, Backbone, Neck, and Head are three main parts of YOLOv5. The Backbone collects and creates image features, while the Neck passes those features to the Head for the prediction. Then, bounding boxes and class predication were made in the Head part.

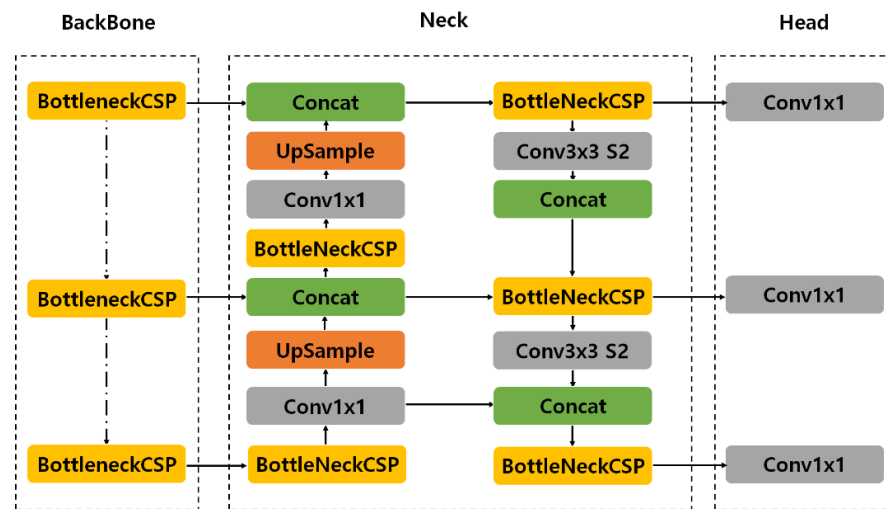


Figure 9. Structure of YOLOv5.

3.3.3. Flow of Crop Disease Diagnosis Solution

The flow of crop disease diagnosis solution consisting of the Image Captioning model and Object Detection model introduced above is shown in Figure 10. First, images of disease-infected crops are simultaneously input into the Image Captioning model and the Object Detection model. In the Image Captioning model, the type, symptom, and degree of damage of the disease are recognized from the features of the crop image. Then, appropriate disease diagnostic sentences in which these features are comprehensively considered are generated. In the Object Detection model, the affected area of the crop is detected in the input image and a bounding box is displayed on the affected area to ensure high reliability in disease diagnosis. Finally, the crop disease diagnostic sentence and the bounding box shown on the affected area are displayed together. On the other hand, when an image of a normal crop that is not infected with disease is input into the crop disease diagnosis solution, an image without a bounding box is presented with a diagnostic sentence that the crop is normal. When an image of a non-crop object is input, an image without a bounding box is presented with a sentence that the crop is unrecognizable.

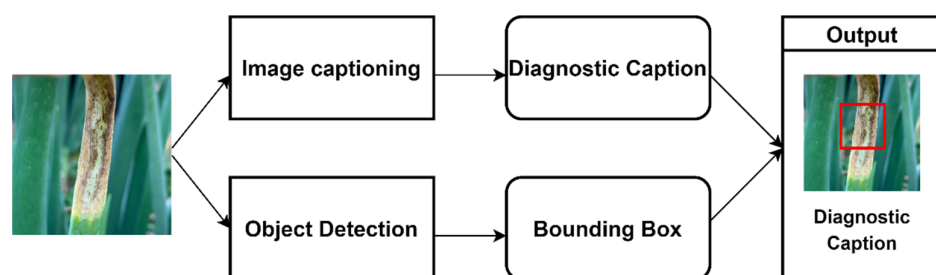


Figure 10. Structural Diagram of Crop Disease Diagnosis Solution.

4. Result

4.1. Quantitative Result

4.1.1. Quantitative Result of Image Captioning Model

The overall training process of the Image Captioning model was done by extracting the features of train images from InceptionV3 and caching them to disk with the '.npy' extension. Meanwhile, we used five captions for every feature as an input of the Transformer.

The training was conducted with setting the epoch to 30 with the Adam optimizer. The loss function was Sparse Category Cross-Entropy with the drop rate at 0.2 to prevent overfitting.

Hyperparameters are the default values published in the ‘Attention all you need’ paper, which are set with four layers of Encoder-Decoder, 2048 hidden layers of Encoder-Decoder Position-Wise-Feed Forward neural networks, 512 input/output dimensions of Encoder-Decoder, and eight heads for parallel use in Multi-Head-Attention.

Caption generation performance of the Image Captioning model was evaluated using BLEU score [32]. The BLEU score is a representative sentence generation performance indicator frequently used in the field of machine translation, yielding a mathematically calculated score of similarity between human-made sentences and model-generated sentences. The BLEU score has a value between 0 and 1, which means that the closer it is to 1, the higher the sentence generation performance. The BLEU score is calculated by tokenizing each sentence in a word unit and comparing the number of tokens shared between the tokens of the human-made sentence and the tokens of the sentence generated by the model. Meanwhile, the N-gram technique can be applied when calculating the BLEU score. Similarities in sentence structure can be analyzed by grouping tokens by the number of words in N-gram units to compare pairs of words with each other. In this study, the performance of the Image Captioning model was analyzed using BLEU_1, BLEU_2, BLEU_3, and BLEU_4, which are BLEU scores applied with 1-g, 2-g, 3-g, and 4-g, respectively, as well as BLEU_AVG, which is the average of these four BLEU scores. The performance of the model was analyzed with 306 images evenly containing disease-infected and normal crops, and each score was multiplied by 100 so that the final score could be expressed as a percentage. Each calculated BLEU score is shown in Table 2.

Table 2. BLEU scores (%).

BLEU_1	BLEU_2	BLEU_3	BLEU_4	BLEU_AVG
78.64	69.77	61.08	50.58	64.96

All scores were calculated by discarding the third decimal place.

As shown in Table 2, all five BLEU scores were more than 50 points, and in particular, the average score of each N-gram-based BLEU score was calculated to be about 64.96 points. According to the ‘AutoML Translation Guide’ on the Google Cloud website, the quality of the sentence is considered very high if the BLEU score of the generated sentence is over 50 points [33]. Evaluating the model’s ability to generate captions based on this criterion, it can be interpreted that the Image Captioning model in this study produces very high quality captions since all measured BLEU scores are over 50 points.

4.1.2. Quantitative Result of Object Detection

The Object Detection model was designed to display bounding boxes to highlight the affected area of the plant, and each label had different colors of boxes. The actual classes for prediction are seven rather than nine, since Powdery Mildew was detected in red peppers and zucchinis, while leaf blight was shown in tomatoes and beans. Meanwhile, among five pretrained models of YOLOv5, which are YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, our team chose YOLOv5m.

The confusion matrix on Figure 11 shows the performance of our Object Detection model. Figure 12 displays the precision-recall curve of our Object Detection model.

According to Figures 12 and 13, Anthracnose, Alternaria Leaf Spot, and Leaf Mold were detected quite well, while detecting Powdery Mildew, Leaf Blight, and Frogeye Leaf Spot shows low level of precision and F1-confidence. Figure 12 suggests mAP50 of each class, which has a similar output compared to the confusion matrix. The average mAP50 was 0.382, indicating that the performance of the model is low. Anthracnose, Alternaria Leaf Spot, and Leaf Mold had a mAP50 score above 0.5, while other classes showed the mAP50 below 0.3. Meanwhile, Figure 13 suggests the highest average F1-score is 0.44 at 0.204 confidence, which shows the significant difference among the classes.

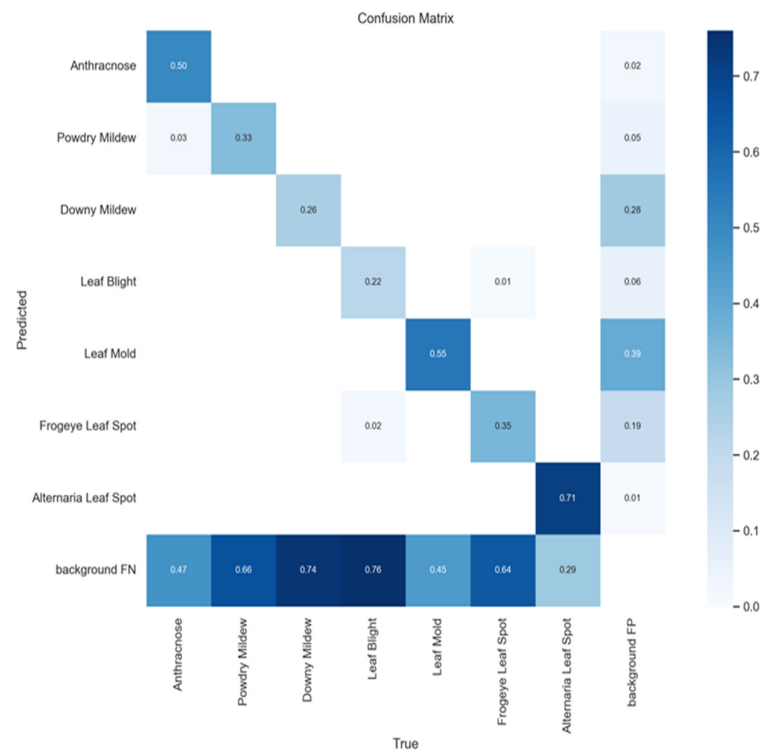


Figure 11. Confusion Matrix of Object Detection Model.

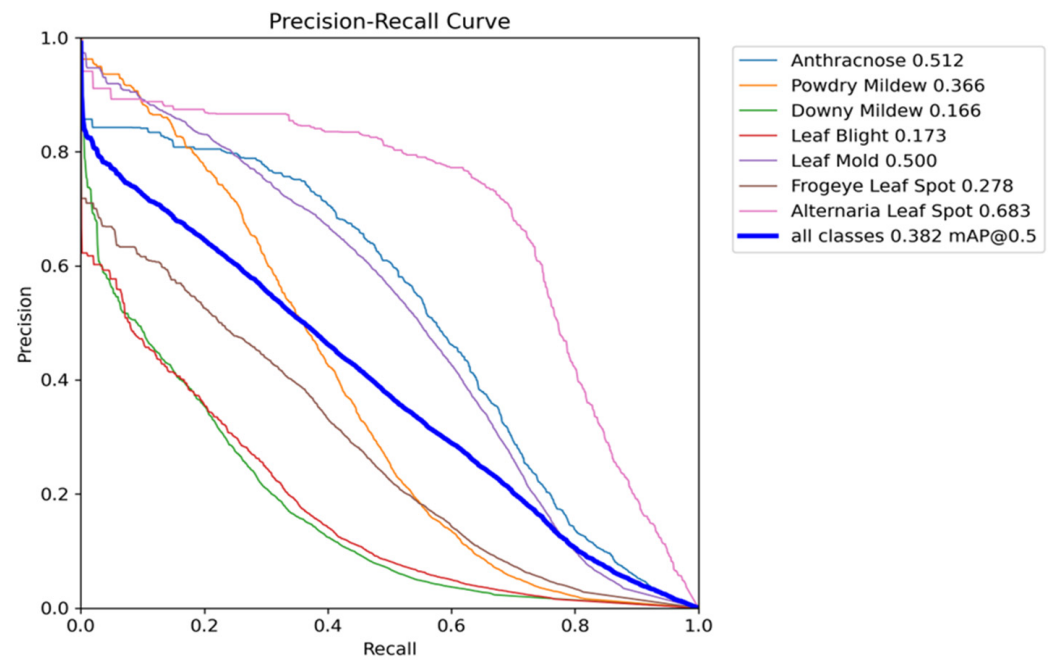


Figure 12. Precision-recall curve of the Object Detection model.

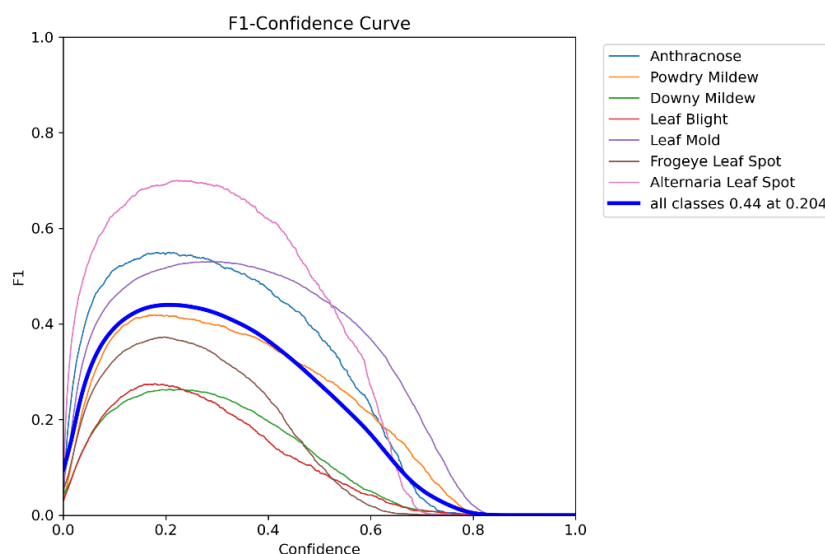


Figure 13. F1-Confidence of Object Detection Model.

4.2. Qualitative Result

4.2.1. Qualitative Result of Image Captioning Model

The following are scenes in which a sentence predicted by the model and a correct answer sentence produced for model training are displayed together. The predicted sentence was generated by inputting a randomly selected image of a disease-infected crop or a normal crop.

First of all, comparing a generated sentence and a correct answer sentence for the image of the disease-infected crop, as shown in Figure 14, they are both structurally and semantically similar, except that the sentence generated by the model replaces the ‘생긴 것으로 (be formed)’ of the answer sentence with ‘나타나는 것 (was appeared)’. In addition, the sentence generated by the model also describes the disease symptom keywords “yellowish-brown spores” and “circular spots” in the correct answer sentence, and, at the same time, accurately diagnoses “pepper anthracnose” as a disease for a pepper, which is a disease diagnosis target crop input in the model. In conclusion, it can be confirmed that the model accurately recognizes the symptoms and degree of damage of the crop and accurately diagnoses the type of disease.

Real Caption: 고추에 황갈색의 포자와 원형반점이 생긴 것으로 보아 고추 탄저병으로 의심됩니다
 Predicted Caption: 고추에 황갈색의 포자와 원형반점이 나타나는 것으로 보아 고추탄저병으로 의심됩니다



Figure 14. Generated sentence (predicted caption) and correct answer sentence (real caption) for the image of an Anthracnose-infected pepper.

Next, comparing the generated sentence and the correct answer sentence for the image of a normal crop, as shown in Figure 15, the positions of ‘토마토 열매에서 (in tomato fruit)’ and ‘특별한 질병이 (special disease)’ in the correct answer sentence are switched with each other. Due to the characteristics of Korean, the meaning of the sentence does not change much even if the word order is changed, and the meaning of the generated sentence is still the same even if the position of those two expressions is changed. Additionally, ‘발견되지 않은 (not found)’ and ‘사료됩니다 (to be thought)’ are replaced by

‘검출되지 않았기 (not detected)’ and ‘간주됩니다 (to be considered)’, respectively. Since the meaning of those replaced expressions is similar to the expression of the correct answer sentence, the meaning of the generated sentence is still the same as the correct answer sentence.

Real Caption: 토마토 열매에서 특별한 질병이 발견되지 않은 것으로 보아 정상으로 사료됩니다
 Predicted Caption: 특별한 질병이 토마토 열매에서 검출되지 않았기 때문에 정상으로 간주됩니다



Figure 15. Generated sentence (predicted caption) and correct answer sentence (real caption) for the image of a normal tomato.

In conclusion, since the Image Captioning model generated sentences with similar meanings to correct answer sentences, it can be evaluated that it has high-performance sentence generation ability.

4.2.2. The Qualitative Result of Object Detection Model

Figure 16 shows the red pepper with Anthracnose detected. The Object Detection model precisely detected the infected area rather than a different region, given that another crop that has a similar color with the damaged area was not detected.



Figure 16. The red pepper with Anthracnose.

Meanwhile, Figure 17 indicates that the crop without symptoms was not detected. Figure 17 is the photo of spring onion, and any related disease, such as leaf spot, was not detected.



Figure 17. Spring onion without anything detected.

4.3. Result Analysis

We designed the solution where an input crop image generates a diagnosis in a detailed sentence and shows the infected area with bounding boxes. This output can help users to know which part is damaged and find a way to manage it.

Figure 18 illustrates the final output of our solution. When the user takes a photo of the infected crops, such as the red pepper with Anthracnose in Figure 18a, the solution generates the diagnosis and bounding boxes on the screen. The diagnosis generated the sentence ‘고추에 원형 반점과 황갈색 포자덩어리가 나타나고 말라 비틀어진 것을 보아 고추탄저병으로 의심됩니다 (The pepper is dried and twisted with yellow-brown spore and round spot on the surface, which is suspected to be infected with Anthracnose)’.

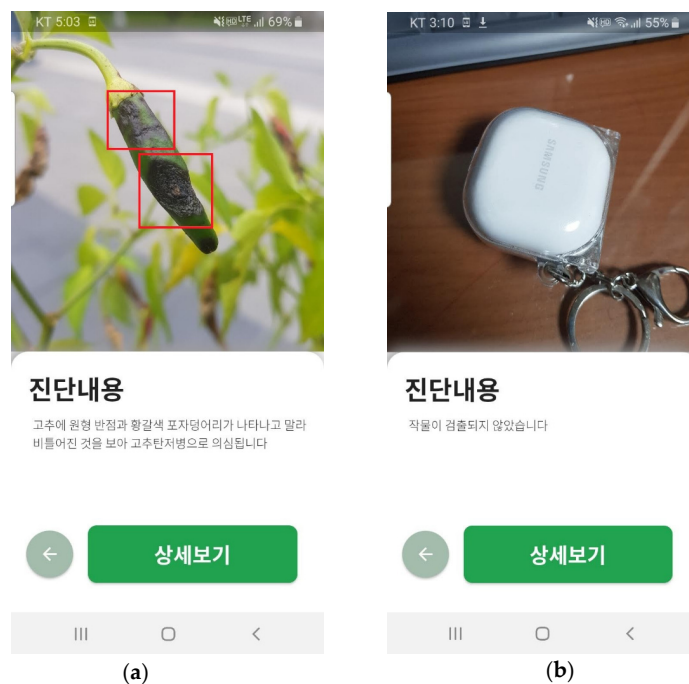


Figure 18. (a) The output of the red pepper with Anthracnose below 진단내용 (Diagnosis); (b) The output of non-crop image below 진단내용 (Diagnosis).

Figure 18b is a wireless earphone with no bounding box. The diagnosis also generated the sentence ‘작물이 검출되지 않았습니다 (Crops were not detected)’.

Total running time was 8.3 s in the Amazon AWS cloud server with vCPU 1, a main memory with a volume of 1GB and 100G environment of network outbound. The running time could have been faster if our research had been conducted in an environment based on better equipment. Consequently, we developed the solution that diagnoses the diseases with precise explanations and clearly illustrates the area of the damaged part rather than merely showing the figures.

5. Conclusions

The research aims to develop technology that can help inexperienced urban farmers manage their crops in a more efficient way. To do so, we utilized the Image Captioning model that presents evidence sentences of its diagnosis and Object Detection that creates bounding boxes on the damaged area. Those two models allowed us to design an accurate crop disease diagnosis solution with detailed explanations. We anticipated a few positive social effects if our solution combines with application services, IOT, or robots. The research aims to develop the technology that can help inexperienced urban farmers manage their crops in a more efficient way. To do so, we utilized the Image Captioning model that presents evidence sentences of its diagnosis and Object Detection that created bounding boxes on the damaged area. Although the mAP50 in the Object Detection model was quite lower than our expectation, overall performances can be improved with further research. Those two models allowed us to design an accurate crop disease diagnosis solution with detailed explanations. We anticipate a few positive social effects when our solution is combined with application service. Moreover, it can also satisfy the social needs of the increasing urban farmers, and even home farmers, raising pet plants. Consequently, we expect that our solution can create the environment for sustainable development and stimulate the agricultural economy.

Author Contributions: Conceptualization, D.I.L., J.H.L. and S.H.J.; Funding acquisition, S.J.O. and I.C.D.; Investigation and methodology, D.I.L., J.H.L. and S.H.J.; Project administration, S.J.O. and I.C.D.; Resources, S.J.O. and I.C.D.; Supervision, S.J.O. and I.C.D.; Writing of the original draft, D.I.L., J.H.L. and S.H.J.; Writing of the review and editing, S.J.O. and I.C.D.; Software, D.I.L., J.H.L. and S.H.J.; Validation, D.I.L., J.H.L. and S.H.J.; Formal analysis, D.I.L., J.H.L. and S.H.J.; Data curation, D.I.L., J.H.L. and S.H.J.; Visualization, D.I.L., J.H.L. and S.H.J. All authors have read and agreed to the published version of the manuscript.

Funding: “This research was supported by the MIST (Ministry of Science, ICT), Korea, under the National Program for Excellence in SW), supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation) in 2023” (2019-0-01816). Also This research was supported by Hankuk University of Foreign Studies Research Fund of 2023. This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2021S1A5A8065934).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Khamparia, A.; Saini, G.; Gupta, D.; Khanna, A.; Tiwari, S.; Albuquerque, V.H.C. Seasonal crops disease prediction and classification using deep convolutional encoder network. *Circuits Syst. Signal Process.* **2020**, *39*, 818–836. [[CrossRef](#)]
2. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 27.
3. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and Tell: A Neural Image Caption Generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3156–3164.

4. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; Bengio, Y. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2048–2057.
5. Li, G.; Zhu, L.; Liu, P.; Yang, Y. Entangled Transformer for Image Captioning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8928–8937.
6. Zhang, X.; Bai, L.; Zhang, Z.; Li, Y. Multi-Scale Keypoints Feature Fusion Network for 3D Object Detection from Point Clouds. *Hum.-Cent. Comput. Inf. Sci.* **2022**, *12*, 12–29.
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
8. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014 2014; pp. 580–587.
9. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
10. Park, D.; Cha, J.W. *Image Caption Generation using Object Attention Mechanism*; Korean Institute of Information Scientists and Engineers: Seoul, Republic of Korea, 2019; pp. 82–87.
11. Jo, M.; Han, S.; Jeong, C. TOD: Trash Object Detection Dataset. *J. Inf. Process. Syst.* **2022**, *18*, 524–534.
12. Kristiani, E.; Tsan, Y.T.; Liu, P.Y.; Yen, N.Y.; Yang, C.T. Binary and Multi-Class Assessment of Face Mask Classification on Edge AI Using CNN and Transfer Learning. *Hum.-Cent. Comput. Inf. Sci.* **2022**, *12*, 53.
13. Mohanty, S.P.; Hughes, D.P.; Salathe, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*, 1419. [[CrossRef](#)] [[PubMed](#)]
14. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [[CrossRef](#)] [[PubMed](#)]
15. Zhong, L.; Hu, L.; Zhou, H. Deep learning based multi-temporal crop classification. *Remote Sens. Environ.* **2019**, *221*, 430–443. [[CrossRef](#)]
16. Zhou, C.; Xing, J. Improved Deep Residual Network for Apple Leaf Disease Identification. *J. Inf. Process. Syst.* **2021**, *17*, 1115–1126.
17. Nanekaran, Y.A.; Zhang, D.; Chen, J.; Tian, Y.; Al-Nabhan, N. Recognition of plant leaf diseases based on computer vision. *J. Ambient. Intell. Humaniz. Comput.* **2020**, 1–18. [[CrossRef](#)]
18. Chen, J.; Zhang, D.; Nanekaran, Y.A.; Li, D. Detection of rice plant diseases based on deep transfer learning. *J. Sci. Food Agric.* **2020**, *100*, 3246–3256. [[CrossRef](#)] [[PubMed](#)]
19. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanekaran, Y.A. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [[CrossRef](#)]
20. Chen, J.; Zeb, A.; Nanekaran, Y.A.; Zhang, D. Stacking ensemble model of deep learning for plant disease recognition. *J. Ambient. Intell. Humaniz. Comput.* **2022**, 1–14. [[CrossRef](#)]
21. Kim, Y.M.; An, H.U.; Jeon, H.G.; Kim, J.P.; Jang, G.J.; Hwang, H.C. A Study of Tram-Pedestrian Collision Prediction Method Using YOLOv5 and Motion Vector. *KIPS Trans. Softw. Data Eng.* **2021**, *10*, 561–568.
22. AI-Hub Home Page. Available online: <https://www.aihub.or.kr/> (accessed on 19 January 2023).
23. AI-Hub; Facility Crop Disease Diagnostic Image Dataset Home Page. Available online: <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=147> (accessed on 19 January 2023).
24. AI-Hub; Outdoor Crop Disease Diagnostic Image Dataset Home Page. Available online: <https://www.aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&aihubDataSe=realm&dataSetSn=153> (accessed on 19 January 2023).
25. National Crop Pest Management System Home Page. Available online: <https://ncpms.rda.go.kr/npms/Main.np> (accessed on 19 January 2023).
26. Cho, S.W. The acquisition of word order in Korean. *Calg. Work. Pap. Linguist.* **1982**, *7*, 53–116.
27. Sennrich, R.; Haddow, B.; Birch, A. Improving Neural Machine Translation Models with Monolingual Data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; Volume 1, pp. 86–96.
28. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
29. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
31. Bahdanau, D.; Cho, K.H.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.

32. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, 7–12 July 2002; pp. 311–318.
33. Google Cloud Home Page. Available online: <https://cloud.google.com/translate/automl/docs/evaluate> (accessed on 19 January 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.