remote sensing

MDPI

Article

Transformer Neural Network for Weed and Crop Classification of High Resolution UAV Images

Reenul Reedha [1], Eric Dericquebourg [1,*], Raphael Canals [2] and Adel Hafiane [1]

[1] INSA CVL, University of Orleans, PRISME Laboratory EA 4229, 18022 Bourges, France; reenul.reedha@insa-cvl.fr (R.R.); adel.hafiane@insa-cvl.fr (A.H.)
[2] INSA CVL, University of Orleans, PRISME Laboratory EA 4229, 45067 Orleans, France; raphael.canals@univ-orleans.fr
* Correspondence: eric.dericquebourg@insa-cvl.fr

Abstract: Monitoring crops and weeds is a major challenge in agriculture and food production today. Weeds compete directly with crops for moisture, nutrients, and sunlight. They therefore have a significant negative impact on crop yield if not sufficiently controlled. Weed detection and mapping is an essential step in weed control. Many existing research studies recognize the importance of remote sensing systems and machine learning algorithms in weed management. Deep learning approaches have shown good performance in many agriculture-related remote sensing tasks, such as plant classification, disease detection, etc. However, despite the success of these approaches, they still face many challenges such as high computation cost, the need of large labelled datasets, intra-class discrimination (in growing phase weeds and crops share many attributes similarity as color, texture, and shape), etc. This paper aims to show that the attention-based deep network is a promising approach to address the forementioned problems, in the context of weeds and crops recognition with drone system. The specific objective of this study was to investigate visual transformers (ViT) and apply them to plant classification in Unmanned Aerial Vehicles (UAV) images. Data were collected using a high-resolution camera mounted on a UAV, which was deployed in beet, parsley and spinach fields. The acquired data were augmented to build larger dataset, since ViT requires large sample sets for better performance, we also adopted the transfer learning strategy. Experiments were set out to assess the effect of training and validation dataset size, as well as the effect of increasing the test set while reducing the training set. The results show that with a small labeled training dataset, the ViT models outperform state-of-the-art models such as EfficientNet and ResNet. The results of this study are promising and show the potential of ViT to be applied to a wide range of remote sensing image analysis tasks.

Keywords: computer vision; deep learning; self-attention; vision transformers; remote sensing; drone; image classification; agriculture

Citation: Reedha, R.; Dericquebourg, E.; Canals, R.; Hafiane, A. Transformer Neural Network for Weed and Crop Classification of High Resolution UAV Images. *Remote Sens.* **2022**, *14*, 592. https://doi.org/10.3390/rs14030592

Academic Editor: Jianxi Huang

Received: 12 November 2021
Accepted: 19 January 2022
Published: 26 January 2022

1. Introduction

Agriculture is at the heart of scientific evolution and innovation to face major challenges for achieving high yield production while protecting plants growth and quality to meet the anticipated demands on the market [1]. However, a major problem arising in modern agriculture is the excessive use of chemicals to boost the production yield and to get rid of unwanted plants such as weeds from the field [2]. Weeds are generally considered harmful to agricultural production [3]. They compete directly with crop plants for water, nutrients and sunlight [4]. Herbicides are often used in large quantities by spraying all over agricultural fields which has, however, shown various concerns like air, water and soil pollution and promoting weed resistance to such chemicals [2]. If the rate of usage of herbicides remains the same, in the near future, weeds will become fully resistant to these products and eventually destroy the harvest [5]. This is why weed and crop control management is becoming an essential field of research nowadays [6].

Automated crop monitoring system is a practical solution that can be beneficial both economically and environmentally. Such a system can reduce labour costs by making use of robots to remove weeds and hence minimising the use of herbicides [7]. The foremost step to an automatic weed control system is the detection and mapping of weeds on the field which can be a challenging part as weeds and crop plants often have similar colours, textures, and shapes [4]. The use of Unmanned Aerial Vehicles (UAVs) has proved significant results for mapping weed density across a field by collecting RGB images ([8–12]) or multispectral images ([13–17]) covering the whole field. As UAVs fly over the field at an elevated altitude, the images captured cover a large ground surface area and these large images can be split into smaller tiles to facilitate their processing ([18–20]) before feeding them to learning algorithms to identify and classify a weed from a crop plant.

In the agricultural domain, the main approach to plant detection is to first extract vegetation from the image background using segmentation and then distinguish crops from the weeds [21]. Common segmentation approaches use multispectral information to separate the vegetation from the background (soil and residuals) [22]. However, weeds and crops are difficult to distinguish from one another even while using spectral information because of their strong similarities [23]. This point has also been highlighted in [6], in which the authors reported the importance of using both spectral and spatial features to identify weeds in crops. In traditional machine learning approaches, features are handcrafted and then algorithms like support vector machines (SVM) are used to generate discriminative models. For example, the authors in [24,25] used this method to detect weeds in potato fields. Literature reviews of this type of approach for weed detection can be found in [26,27].

Classical machine learning approaches depend on feature engineering, where one has to design feature extractors, which generally performs well on small databases but fails on larger and varied data. In contrast, deep learning (DL) approaches rely on learning feature extractors and have shown much better performance compared to traditional methods. Therefore, DL becames an essential approach in image classification, object detection and recognition [28,29] notably in the agricultural domain [30]. DL models with architectures based on Convolutional Neural Network (CNN), have been applied to various domains as they yield high accuracy for image classification and object detection tasks [31–33]. CNN uses convolutional filters on an image to extract important features to understand the object of interest in an image with the help of convolutional operations covering key properties such as local connection, parameters (weight) sharing and translation equivariance [28,34]. Numerous papers covering weed detection or classification make use of CNN-based model structures [35–37] such as AlexNet [32], VGG-19, VGG-16 [38], GoogLeNet [39], ResNet-50, ResNet-101 [33] and Inception-v3 [40].

On the other hand, attention mechanism has seen a rapid development particularly in natural language processing (NLP) [41] and has shown impressive performance gains when compared to previous generation of models [42]. In vision applications, the use of attention mechanism has been much more limited, due to the high computational cost as the number of pixels in an image is much larger than the number of units of words in NLP applications. This makes it impossible to apply standard attention models to images. A recent survey of applications of transformer networks in computer vision can be found in [43]. The recently proposed vision transformer (ViT) appears to be a major step towards adopting transformer-attention models for computer vision tasks [44]. Where image patches are considered as units of information for training, whereas CNN-based methods operate on image pixel level. ViT incorporates image patches into a shared space and learns the relation between these patches using self-attention modules. Given massive amounts of training data and computational resources, ViT was shown to surpass CNNs in image classification accuracy [44]. Vision transformer models have not been explored yet for the task of weeds and crops classification of high resolution UAV images. To our best knowledge, there is no study that has examined their potential for such a task.

In this paper, we propose a methodology to automatically recognize weeds and crops in drone images using the vision transformer approach. We set up an acquisition system

with a drone and a high resolution camera. The images were captured in real-world conditions on plots of different crops: red leaf beet, green leaf beet, parsley and spinach. The main objective was to study the paradigm of transformers architectures for specific tasks such as plant recognition in UAV images, where labeled data are not available in large quantities. Data augmentation and transfer learning were used as a strategy to fill the gap of labeled data. To evaluate the performance of the self-attention mechanism via vision transformers, we fluctuated the proportions of data used for training and for testing within cross-validation scheem. The contributions are summarized in the following points:

- Low-altitude aerial imagery based on UAVs and self-attention algorithms for crop management.
- First study to explore the potential of transformers for classification of weed and crop images.
- Evaluation of the generalization capabilities of deep learning algorithms with regard to train set reduction, in crop plants classification task.

The rest of the paper is organised as follows: Section 2 presents the materials and methods used as well as a brief description of the self-attention mechanism and the vision transformer model architecture. The experimental results and analysis are presented in Sections 3 and 4. We discuss the results in Section 5. Section 6 summarizes our study and provides some perspectives.

2. Materials and Methods

This section outlines the acquisition, preparation and labeling of the dataset acquired using a high resolution camera mounted on a UAV, and describes both: the self-attention paradigm and the vision transformer model architecture.

2.1. Image Collection and Annotation

The study area is composed of crop fields of beet, parsley and spinach located in the Centre-Val de Loire Region, in France. It is a highly agricultural region as it presents many pedo-climatic advantages: the region has limited rainfall and clay-limestone soils with good filtering capacity. Irrigation is also offered on 95% of the plots, enabling controlled water conditions.

To survey the study area, a "*Starfury*", Pilgrim UAV was equipped with a *Sony ILCE-7R*, 36 mega pixel camera as shown in Figure 1. The camera is mounted to the drone using a 3-axis stabilized brushless gimbals in order to keep the camera axis stable even during strong winds. The drone flight altitude was respectively of 30 m for the beet field and 20 m for the parsley and spinach fields. These altitudes where selected to minimize drone flight times while maintaining sufficient image quality. The beet plants being more developed, a higher altitude was selected. The aerial image acquisitions of the 3 fields were also conducted at different times depending of the weed levels reported by ground field experts. Acquiring images over multiple days resulted in adding variability in the images, as the beet field was flown by a light morning fog and the parsley and spinach fields under sunnier weather conditions.

The drone followed a specific flight plan and the camera captured RGB images at regular intervals as shown in Figure 2 and Figure 3. The images captured have respectively a minimum longitudinal and lateral overlapping of 70% and 50–60% depending on the fields vegetation coverage and homogeneity, assuring a better and complete coverage of the whole field of 4 ha (40,000 m^2) and improving the accuracy of the orthorectified image of the field.

(**a**) (**b**)

Figure 1. Apparatus used for data acquisition. (**a**) Starfury Drone; (**b**) Sony ILCE-7R Camera.

Figure 2. Overlay of the orthophoto on google earth of the spinach plot (**left**) and the flight plan (**right**) across a spinach field (the images are taken along the yellow lines at regular intervals to ensure sufficient overlapping).

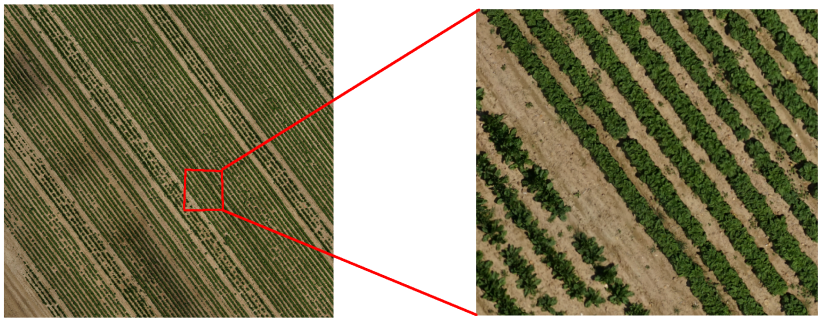

Figure 3. Example of image captured from a spinach study site.

The data were manually processed using the annotation tool LabelImg (https://github.com/tzutalin/labelImg, accessed on 7 September 2021) on the tiles of the orthorectified image. Weeds and crops were annotated using bounding boxes, which may have various sizes and contain a portion of the object of interest. We extract the crop and weed image patches from the bounding boxes. Then, the image patches are resized to 64 × 64 pixels. This image size was chosen because the bounding box dimensions were centered around 64 × 64 pixels, which may be proportionally related to the flight height of the UAV and the size of the crops observed in the study fields. Resizing the patches to the average bounding box dimensions also limits width and height distortions in the input images. We divided the crop and weed labels into 5 classes as shown in Figure 4. We have a class for each of the studied crops, an overall weed class, and an off-type green leaf beet class.

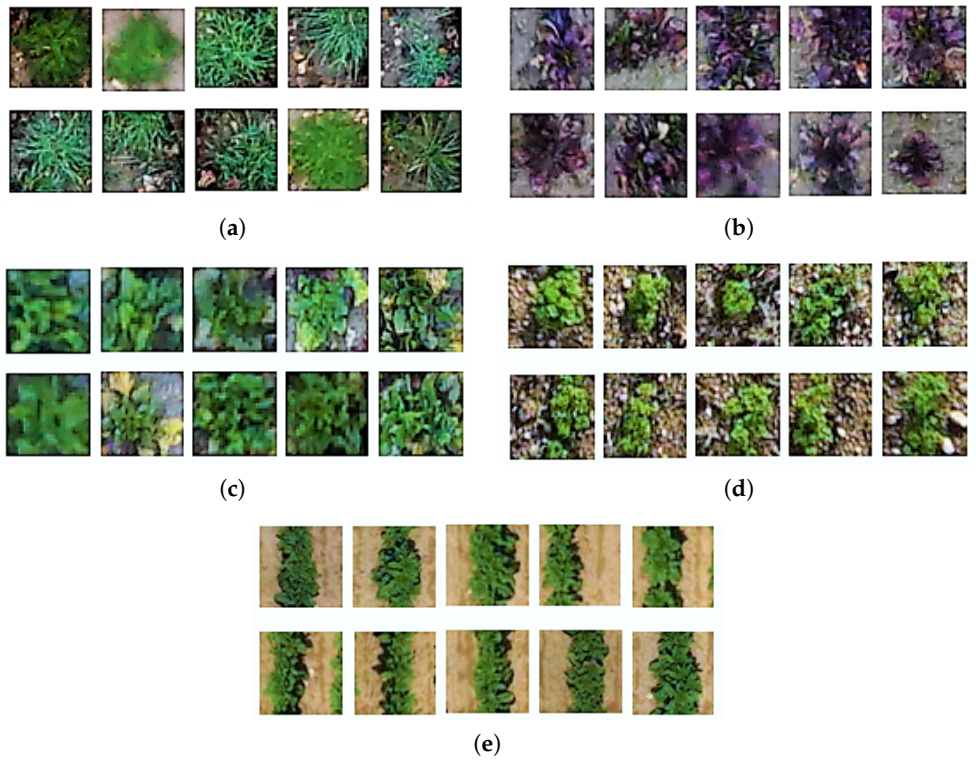

Figure 4. This overview shows sample images patches of all 5 classes of our custom dataset. The images measure 64 × 64 pixels. Each class contains 3200 to 4000 images. (**a**) Weeds; (**b**) Beet; (**c**) Off-type green leaves beet; (**d**) Parsley; (**e**) Spinach.

2.2. Image Preprocessing

Manual image labeling being a very time consuming task which implying huge labor costs, therefore, we limited the manual labeling to 4000 samples for each crop and weed classes. Off-type green leaf beet is not as well represented as the other 4 classes, with only 653 labeled samples. In order to tackle this class imbalance, we upsampled four times the off-type beet class up to 3265 samples, by performing random flips and rotations. Resulting in a dataset distribution of 16.9% of off-type beet plants, and equally 20.8% images for the four other classes as presented in Table 1, for a total of 19,265 images of size 64 × 64.

Table 1. Class Distribution.

Class	Number
Weed	4000
Beet	4000
Off-type Beet	3265
Parsley	4000
Spinach	4000

Images have been rescaled to 0–1 range and then normalized by scaling the pixels values to have a zero mean and unit variance before being divided into training, validation and testing sets.

During the training phase, we employed data augmentation strategies to enrich the datasets as it plays an important role in deep learning [45]. The augmentations applied can be summed up as random resized crop, colour jitters and rand augments [46]. This technique is implemented using *Keras ImageDataGenerator*, generating augmented images on the fly.Data augmentations were used to help improve the robustness of the model

and generalisation capabilities by expanding the training dataset and simulate real-world agricultural scenarios as they can vary a lot depending on the soil, environment, season and climate conditions.

2.3. ViT Self-Attention

The attention mechanism is becoming a key concept in the deep learning field [47]. Attention was inspired by the human perception process where the human tends to focus on parts of information, ignoring other perceptible parts of information at the same time. The attention mechanism has had a profound impact on the field of natural language processing, where the goal was to focus on a subset of important words. The self-attention paradigm has emerged from the concepts attention showing improvement in the performance of deep networks [42].

Let us denote a sequence of n entities $(\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n})$ by $X \in \mathbb{R}^{n \times d}$, where d is the embedding dimension to represent each entity. The goal of self-attention is to capture the interaction amongst all n entities by encoding each entity in terms of the global contextual information. This is done by defining three learnable weight matrices, Queries ($W^Q \in \mathbb{R}^{n \times d_q}$), Keys ($W^K \in \mathbb{R}^{n \times d_k}$) and Values ($W^V \in \mathbb{R}^{n \times d_v}$). The input sequence X is first projected onto these weight matrices to get $Q = XW^Q, K = XW^K$ and $V = XW^V$.

The attention matrix $A \in \mathbb{R}^{n \times d_v}$ indicates a score between N queries Q and K^T keys representing which part of the input sequence to focus on.

$$A(Q, K) = \sigma(QK^T) \tag{1}$$

where σ is an activation function, usually $softmax()$. To capture the relations among the input sequence, the values V are weighted by the scores from Equation (1). Resulting in [44],

$$\begin{aligned} SelfAttention(Q, K, V) &= A(Q, K) \cdot V \\ \Rightarrow SelfAttention(Q, k, V) &= softmax(\frac{QK^T}{\sqrt{d_k}}) \cdot V \end{aligned} \tag{2}$$

where d_k is dimension of the input queries.

If each pixel in a feature map is regarded as a random variable and the covariances are calculated, the value of each predicted pixel can be enhanced or weakened based on its similarity to other pixels in the image. The mechanism of employing similar pixels in training and prediction and ignoring dissimilar pixels is called the self-attention mechanism. It helps to relate different positions of a single sequence of image patches in order to gain a more vivid representation of the whole image [48].

The transformer network is an extension of the attention mechanism from Equation (2) based on the Multi-Head Attention operation. It is based on running k self-attention operations, called "heads", in parallel, and project their concatenated outputs [42]. This helps the transformer jointly attend to different information derived from each head. The output matrix is obtained by the concatenation of each attention heads and a dot product with the weight W^O. Hence, generating the output of the multi-headed attention layer. The overall operation is summarised by the equations below [42].

$$\begin{aligned} MultiHead(Q, K, V) &= Concat(head_1, ..., head_h)W^O \\ where head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{3}$$

where W_i^Q, W_i^K, W_i^V are weight matrices for queries, keys and values respectively and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$.

By using the self-attention mechanism, global reference can be realised during the training and prediction of models. This helps in reducing by a considerable amount training time of the model to achieve high accuracy [44]. The self-attention mechanism is an integral

component of transformers, which explicitly models the interactions between all entities of a sequence for structured prediction tasks. Basically, a self-attention layer updates each component of a sequence by aggregating global information from the complete input sequence. While, the convolution layers' receptive field is a fixed $K \times K$ neighbourhood grid, the self-attention's receptive field is the full image. The self-attention mechanism increases the receptive field compared to the CNN without adding computational cost associated with very large kernel sizes [49]. Furthermore, self-attention is invariant to permutations and changes in the number of input points. As a result, it can easily operate on irregular inputs as opposed to standard convolution that requires grid structures [43].

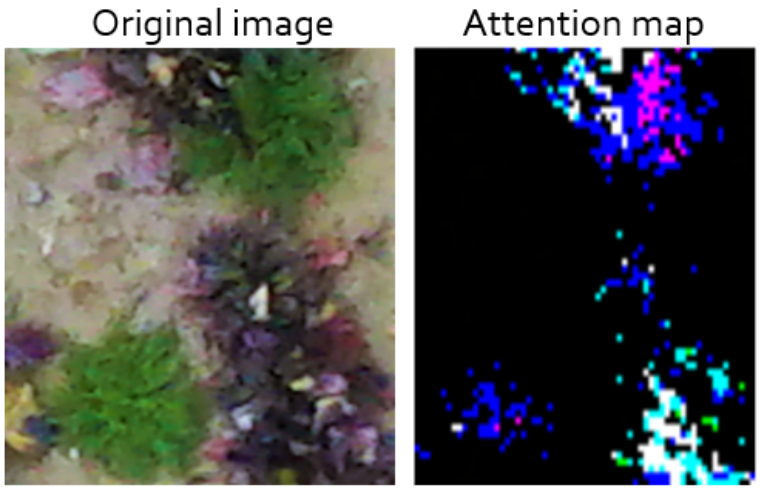

Figure 5. Attention mechanism on an image patch (**left**) containing weeds (in green) and beet plant (in red). With the original image on the left and the attention map (**right**) obtained with ViT-B16 model. The attention map shows the model's attention on the different plants: with a dark blue and purple colour pixel representing the attention on the weeds and a light blue colour pixel representing the beet plant.

Average attention weights of all heads mean heads across layers and the head in the same layer. Basically, the area has every attention in the transformer which is called attention pattern or attention matrix. When the patch of the weed image is passed through the transformer, it will generate the attention weight matrix for the image patches (see Figure 5). For example, when patch 1 is passed through the transformer, self-attention will calculate how much attention should pay to others (patch 2, patch 3, ...). In addition, every head will have one attention pattern as shown in Figure 6 and finally, they will sum up all attention patterns (all heads). We can observe that the model tries to identify the object (weed) on the image and tries to focus its attention on it (as it stands out from the background).

An attention mechanism is applied to selectively give more importance to some of the locations of the image compared to others, for generating caption(s) corresponding to the image. In addition, consequently, this helps to focus on the main differences between weeds and crops in an images and improves the learning of the model to identify the contrasts between these plants. This mechanism also helps the model to learn features faster, and eventually decreases the training cost [44].

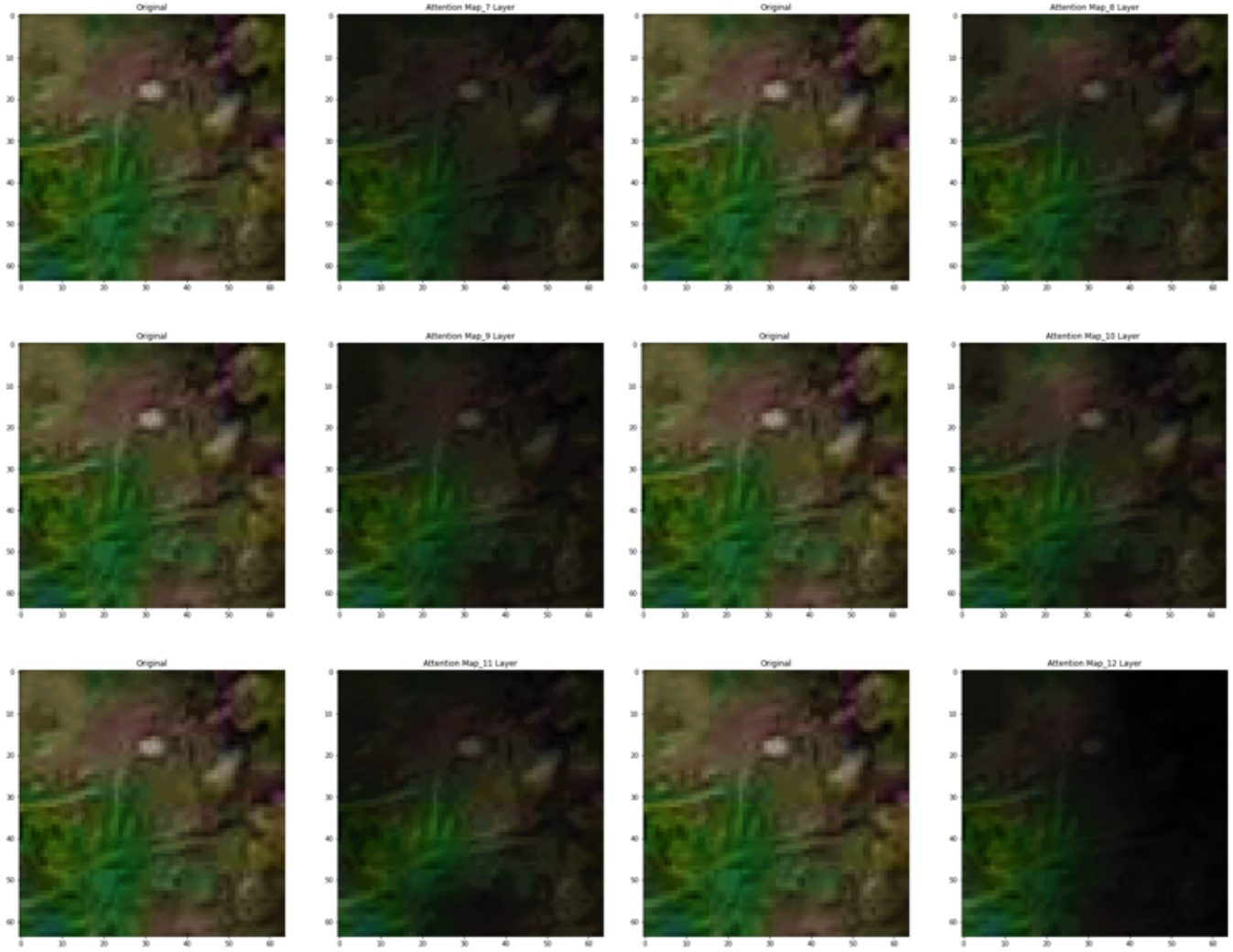

Figure 6. Attention map generated from layers 7 to 12 of the ViT-B16 model on an image of a weed.

2.4. Vision Transformers

Transformer models were major headway in NLP. They became the standard for modern NLP tasks and they brought spectacular performance yields when compared to the previous generation of state-of-the-art models [42]. Recently, it was reviewed and introduced to computer vision and image classification aiming to show that this reliance on CNNs is not necessary anymore in object detection or image classification and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks [44].

Figure 7 presents the architecture of the vision transformer used in this paper for weed and crop classification. It is based on the first developed ViT model by Dosovitskiy et al. [44]. The model architecture consists of 7 main steps. Firstly, the input image is split into smaller fixed-size patches. Then each patch is flattened into a 1-D vector. The input sequence consists of the flattened vector (2D to 1D) of pixel values from a patch of size 16×16.

For an input image,

$$(x) \in \mathbb{R}^{H \times W \times C} \tag{4}$$

and patch size P, N image patches are created

$$(x)_P \in \mathbb{R}^{N \times P \times P \times C} \tag{5}$$

with

$$N = \frac{HW}{P \times P} \tag{6}$$

where N is the sequence length (token) similar to the words of a sentence, (H, W) is the resolution of the original image and C is the number of channels [44].

Afterwards, each flattened element is then fed into a linear projection layer that will produce what is called the "patch embedding". There is one single matrix, represented as 'E' (embedding) used for the linear projection. A single patch is taken and first unrolled into a linear vector as shown in Figure 8. This vector is then multiplied with the embedding matrix E. The final result is then fed to the transformer, along with the positional embedding. In the 4^{th} phase, the position embeddings are linearly added to the sequence of image patches so that the images can retain their positional information. It injects information about the relative or absolute position of the image patches in the sequence. The next step is to attach an extra learnable (class) embedding to the sequence according to the position of the image patch. This class embedding is used to predict the class of the input image after being updated by self-attention. Finally, the classification is performed by stacking a multilayer perceptron (MLP) head on top of the transformer, at the position of the extra learnable embedding that has been added to the sequence.

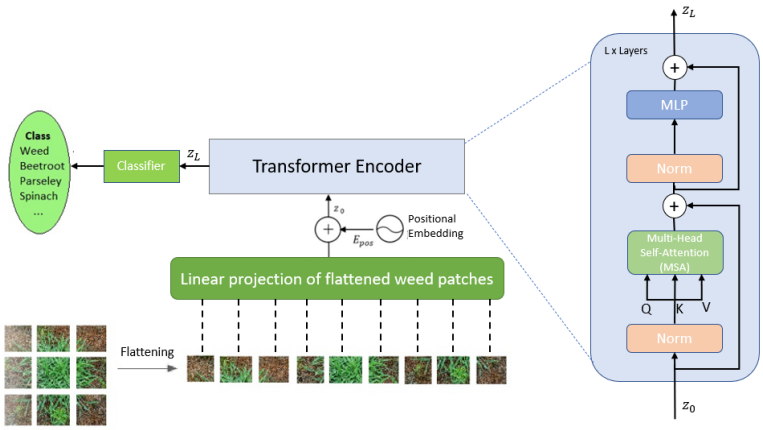

Figure 7. ViT model architecture based on original ViT model [44].

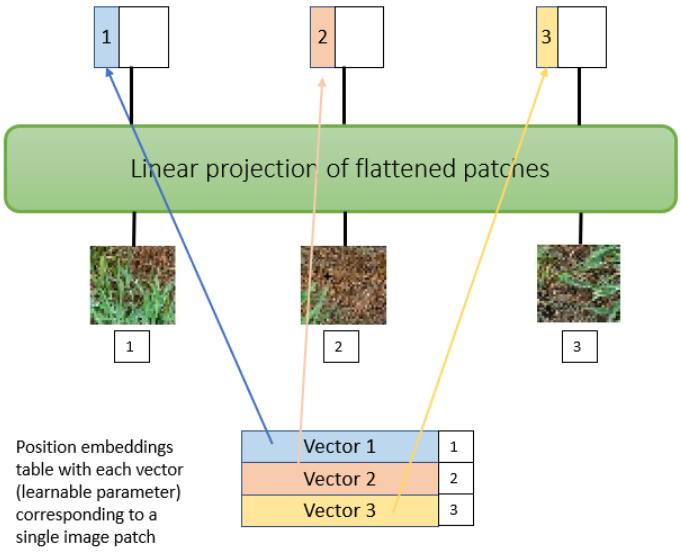

Figure 8. Positional embeddings as vector representations.

3. Performance Evaluation

We made use of recent implementations of ViT-B32 and ViT-B16 models as well as EfficientNet and ResNet models. The algorithms were built on top of a Tensorflow 2.4.1 and Keras 2.4.3 frameworks using Python 3.6.9. To run and evaluate our methods, we used the following hardware; an Intel Xeon(R) CPU E5-1620 v4 3.50 GHz x 8 processor (CPU) with 16 GB of RAM, and a graphics processing unit (GPU) NVIDIA Quadro M2000 with an internal RAM of 4 GB under the Linux operating system Ubuntu 18.04 LTS (64 bits).

All models were trained using the same parameters in order to have an unbiased and reliable comparison between their performance. The initial learning rate was set to 0.0001 with a reducing factor of 0.2. The batch size was set to 8 and the models were trained for 100 epochs with an early stopping after a wait of 10 epochs without better scores. The models used, ViT-B16, ViT-B32, EfficientNet B0, EfficientNet B1 and ResNet 50 were loaded from the keras library with pre-trained weights of "ImageNet".

We limited the comparison of the ViT Based models with ResNet and EfficientNet CNN architectures as they are widely used CNN architectures and have been applied to various study domains. More specifically, the ResNet architecture [33] was the first CNN architecture introducing residual blocks. Where the residual blocks use skip connections between layers providing alternative paths for the gradient backpropagation, resulting in improving accuracy. We selected the ResNet-50 version for the residual architecture using 3-layer building blocks which yields better results compared to 2-layer building blocks as used in ResNet-34. The second CNN architecture considered is the EfficientNet architecture [50], the particularity of the EfficientNet neural network family is that is has highly optimised parameters and yields equivalent or higher Top-1 results depending on the version of the network used.

3.1. Cross-Validation

The experiments have been carried out using the cross-validation technique to ensure the integrity and accuracy of the models. Cross-validation is a widely used technique for assessing models as the performance evaluation is carried out on unseen test data [51], the method also presents the advantage of being a low bias resampling method [52].

As our dataset classes are not perfectly balanced we applied stratified K-Fold. By applying stratification, each randomly sampled fold will have an equal class distribution in respect to the total dataset distribution. From these folds we then test the performance of the models using the k-fold cross-validation leaving k folds as validation set.

To assess the performance of ViT models with respect to the selected CNN architectures, we performed 3 workflows. First, we performed cross-validation with one validation set (k = 1) to maximize the size of the training dataset and evaluate the performance at a fixed test fold (see Figure 9). Second, we decreased the number of training folds and increased the size of the validation set while keeping the same test fold. Finally, we reduced the number of training folds while maintaining a single validation fold (k = 1) and increasing the size of the test set, to evaluate the predictive performance of the models when trained on small data sets.

Using the stratified five-folds cross-validation leaving k folds as validation set (where $1 \leq k \leq 4$), Figure 10 shows how the dataset is splitted with $n = 5$ and $k = 2$ where n represents the total number of cross-validation folds, resulting in the training of 10 models. Increasing the value of k, decreases the number of folds used for training and thus forces the model to train on a smaller dataset. This helps to evaluate how well the models perform on reduced training datasets and their capacities to extract features from fewer image samples. The number of combination (splits) of the train-validation is as follows:

$$C_k^n = \frac{n!}{k!(n-k)!} \tag{7}$$

where n is the number of folders and k is the number of validation folds.

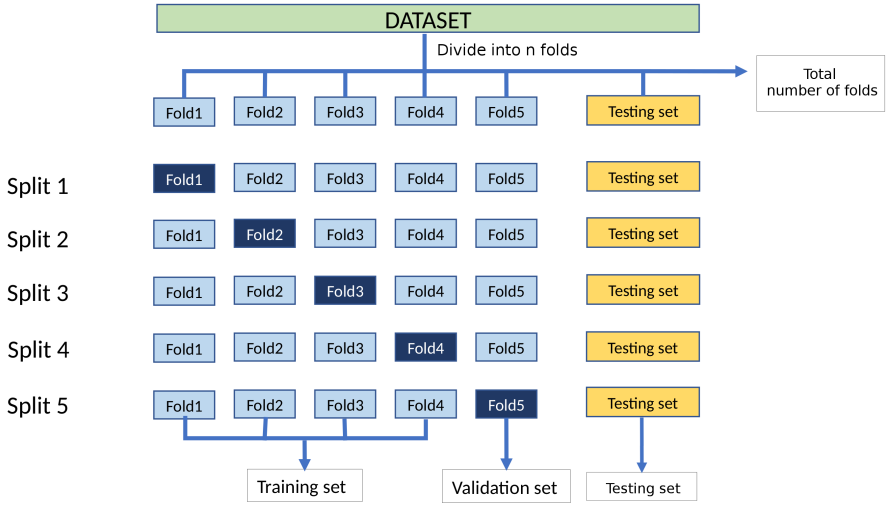

Figure 9. Stratified five-folds cross-validation, leaving one out for validation and the remaining 4 folds are used for training. Dark blue representing validation folds, light blue colour folds are used as training set and yellow colour folds are used as testing set containing unprocessed images. This generates 5 trained models.

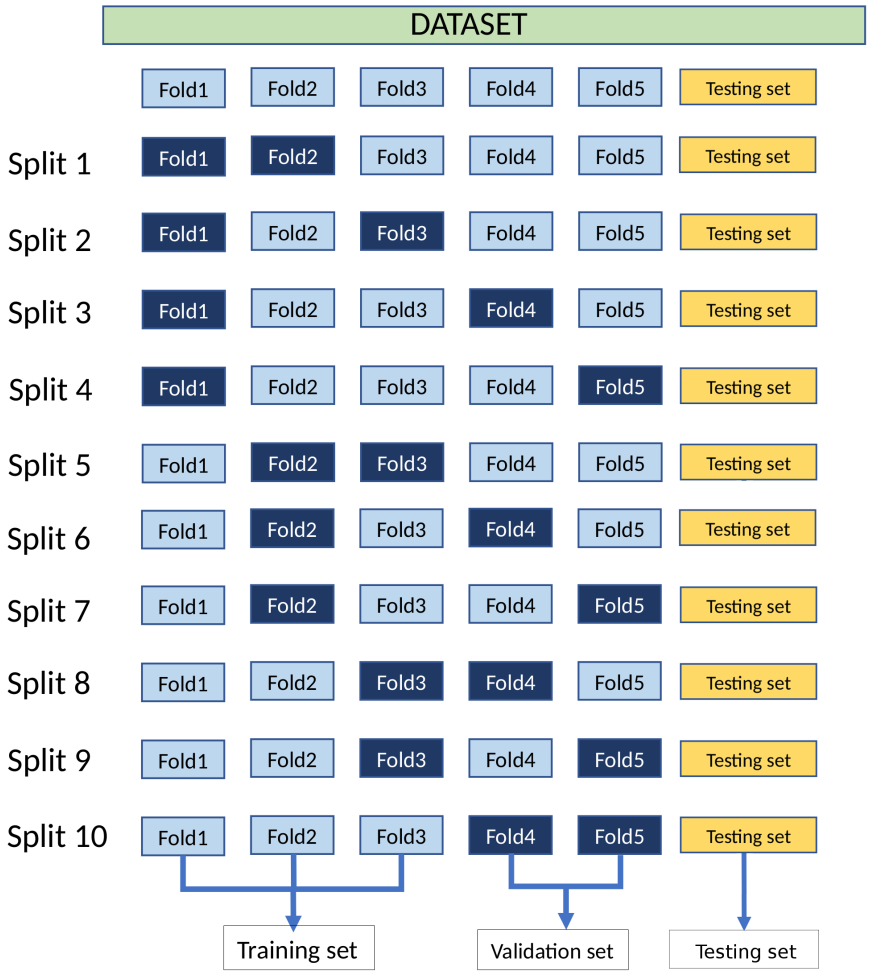

Figure 10. Stratified five-folds cross-validation and leaving two out as validation set and the rest are used for training resulting in 10 different models. Dark blue representing validation folds and light blue colour folds are used as training set.

For the third workflow, we conducted three experiments. The number of testing images is increased for each experiment, consequently decreasing the number of training images. In experiment 1, the dataset was split into 9633 training and 6421 testing images. In experiment 2, the dataset was divided into 6422 training and 9633 testing images. Experiment 3 contains only 3211 training images for 12,843 testing images. Each set up of experiments is then trained using the cross-validation technique (see Figure 11).

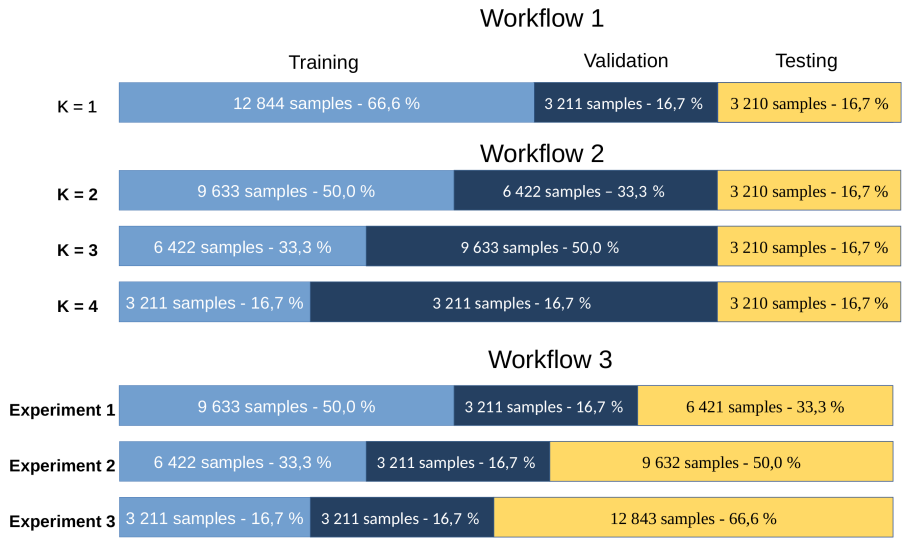

Figure 11. Variation of training/validation set and testing image set for the 3 workflows. The training/validation set is used for the cross-validation as shown in Figure 9.

3.2. Evaluation Metrics

In the collected dataset, each image has been manually classified into one of the categories: weeds, off-type beet (green leaves beet), beet (red leaves), parsley or spinach, called ground-truth data. By running the classifiers on a test set, we obtained a label for each testing image, resulting in the predicted classes. The classification performance is measured by evaluating the relevance between the ground-truth labels and the predicted ones resulting in classification probabilities of true positives (TP), false positives (FP) and false negatives (FN). We then calculate a recall measure representing how well a model correctly predicts all the ground-truth classes and a precision representing the ratio of how many of the positive predictions were correct relative to all the positive predictions.

$$\begin{aligned} Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \end{aligned} \tag{8}$$

The metrics used in the evaluation procedure were the precision, recall and F1-Score [53], the latter being the weighted average of precision and recall, hence considering both false positive and false negatives. Comparison studies have shown that these metrics are relevant for evaluation of classification model performance [54].

These metrics were also selected as in opposition to accuracy, they are invariant to class distribution. This invariance property is due to the consideration of only TP and not TN predictions in the computation of precision and recall [55]. Not taking TNs into account can sometimes cause issues in particular classification tasks where TNs have a significant impact in certain domains. This is not the case in our agricultural application, since an example of a TN would be to predict a crop sample as a weed, when it is more desirable

to not classify a weed as a crop. In other words, it is better to over-detect weeds than to under-detect them.

$$F1 - Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \tag{9}$$

Since we used cross-validation techniques to evaluate the performance of each model, we calculated the mean (μ) and standard deviation (σ) of the F1-scores of the model in order to have an average overview of its performance. The equations used are presented below:

$$\begin{aligned} \mu_{F1-Score} &= \frac{\sum_{i=1}^{\mathcal{N}}(F1 - Score_i)}{\mathcal{N}} \\ \sigma_{F1-Score} &= \sqrt{\frac{\sum_{i=1}^{\mathcal{N}}(F1 - Score_i - \mu_{F1-Score})^2}{\mathcal{N}}} \end{aligned} \tag{10}$$

where $\mathcal{N}$ is the number of splits generated from the cross validation procedure. For instance, leave one out generates five splits ($\mathcal{N} = 5$) using Equation (7) as shown in Figure 9.

As for the loss metrics, we used the cross-entropy loss function between the true classes and predicted classes.

4. Results

CNN-Based architectures, ResNet and EfficientNet were trained along the ViT-B16 and ViT-B32 in order to compare their performance on our custum dataset comprising of 5 classes (weeds, beet, off-type beet, parsley and spinach). All models have been trained using the five-folds cross-validation leaving one out technique. With this technique, the models were trained using 12,844 samples (66.6 %), validated with 3211 (16.7 %) and tested on 3210 (16.7 %) image samples. The accuracies and losses of the models tend to be flat after the 30th epoch. The average F1-Scores and losses obtained for the considered models are reported in Table 2.

Table 2. Comparison between state-of-the-art CNN-based models and vision transformer models on agricultural image classification. The F1-Score has been calculated using Equation (10) with $\mathcal{N} = 5$.

Model	$\mu_{F1-Score}$	μ_{Loss}
ViT B-16	**0.994 ± 0.002**	**0.656**
ViT B-32	**0.992 ± 0.002**	**0.672**
EfficientNet B0	0.987 ± 0.005	0.735
EfficientNet B1	0.989 ± 0.005	0.720
ResNet 50	0.992 ± 0.005	0.716

From these experimental results, we notice the outperformance of the ViT models compared to the CNN models, with a best F1-Score of 99.4 % for the ViT B-16 model although the ViT B-32 models's performance is very close behind at 99.2 % with a minimum loss of 0.656. The EfficientNet and ResNet models fall behind compared to the ViT models but with high scores nevertheless, having been trained on a large dataset (12,844 training images). The experimental results confirm vision transformers high performance compared to state of the art models ResNet and EfficientNet as presented by [44]. Although all network families obtain high accuracy and F1-Score, the classification of crops and weed images using vision transformer yields the best prediction performance.

Influence of the Training Set Size

In the next stage, we tried to answer the question of which network family yields the best performance with a smaller training dataset. We did so by carrying out a five-folds

cross-validation leaving k out where k is a varying parameter from 1 to 4 while keeping the testing set to 3210 images to evaluate the performance of the models.

Varying the number of training images has a direct influence on the performance of the trained ViT model, as shown in Table 3. The results obtained with the five-folds cross-validation, leaving two out as a validation set ($k = 2$) are promising, with a mean F1-Score of 99.28 % and a standart deviation of 0.1% showing a very small decrease in the performance of the ViT B-16 model while reducing the number of training images. We note a very light decrease of 0.1% in the accuracy of the ViT B-16 model while training only with 2/5 of the dataset (6422 images, $k = 3$) and validating on the remaining 3/5. With $k = 4$, the ViT B-16 model was trained with a smaller dataset of 3211 images (75% reduction), and its performance decreased as expected but by a small margin of only 0.44 % for an overall accuracy of 99.63 %. These experimental results show how well the vision transformer models perform with small datasets and transfer learning.

We also compared the performance of the ViT B-16 model to CNN-based models ResNet and EfficientNet with a deacreasing number of training images. The experimental results of their F1-Scores are reported in Figure 12. We notice a deacrease in the F1-Scores of the ResNet50, EfficientNet B0 and EfficientNet B1 with a reduction in the number of training images. In contrast, the ViT B-16 model keeps its high performance in the set of experiments, specially with the smallest number of training images, achieving an F1-Score of 99.07%. On the other hand, ResNet 50 scores an accuracy of 97.54%, EfficientNet B0 scores 96.53% and EfficientNet B1 with the worst score of 95.91%. EfficientNet B1 has the worst decrease in performance of 3.07% (from 98.98%—with 12,844 training images to 95.91%—with 3211 training images). Even though EfficientNet B1 achieves better results with the largest dataset (98.98% accuracy) than EfficientNet B0 (98.78%), its performance falls off the most with the smallest train dataset. While the F1-Scores of ResNet and EfficientNet B0 and B1 declines with a reduction of training images by 25% (from 12,844 images to 9633 images), the ViT B-16 model still achieves a high performance of 99.28% (a slight decrease from 99.44%). These experimental results show the outperformance of vision transformer models over current CNN-based models ResNet and EfficientNet in agricultural image classification when dealing with small training datasets.

Furthermore, we compared the performance of the models with by varying the number of testing images while using a 5-folds leaving one fold out cross-validation technique. The ViT results for each class are reported in Table 4. It can be observed that there is a slight decrease in performance along the reduction of the train set and the increase of the test set, indicating a good stability of ViT with the variation of the dataset size. As shown in Figure 13, there is a notable decrease in the F1-Scores of the four models while testing on 9632 and 12,843 images and training with only 33.3% and 16.7% of the labelled dataset. On the third experiment, models were trained on only 3211 images and also validating on 3211 images, which explains the decrease in their performances. Even though all models have a decrease in their F1-Scores with an increasing number of testing images, the ViT B-16 model still achieves higher performance than EfficientNet B0, EfficientNet B1 and ResNet50. The ViT B-16 model had the smallest decrease in performance from 99.44% (from 3210 testing images and 12,844 training images) to 98.63% (from 12,844 testing images an 3211 training images).

Table 3. Comparison of classification reports generated from 5-Fold cross-validation leaving k folds out (where k-folds stands for the number of validation folds) with $1 \leq k \leq 4$. k = 1 represents the most number of training images (12,844) and k = 4 represents the lowest number of training images (3211). The average precision, recall and F1-Score, obtained using Equation (10) are reported for each class obtained with the ViT B-16 model.

k-Folds / **Classes**	k **= 1**			k **= 2**			k **= 3**			k **= 4**		
	$\mu_{Precision}$	μ_{Recall}	$\mu_{F1-Score}$	$\mu_{Precision}$	μ_{Recall}	$\mu_{F1-Score}$	$\mu_{Precision}$	μ_{Recall}	$\mu_{F1-Score}$	$\mu_{Precision}$	μ_{Recall}	$\mu_{F1-Score}$
Weeds	0.996	0.979	0.988 ± 0.001	0.989	0.980	0.984 ± 0.002	0.988	0.972	0.980 ± 0.001	0.984	0.977	0.981 ± 0.001
Off-Type Beet	0.977	0.996	0.986 ± 0.001	0.978	0.987	0.983 ± 0.002	0.969	0.986	0.977 ± 0.002	0.973	0.980	0.977 ± 0.001
Beet	0.998	1.000	0.999 ± 0.000	0.998	0.999	0.998 ± 0.003	0.998	1.000	0.999 ± 0.003	0.997	0.998	0.998 ± 0.001
Parsley	1.000	1.000	1.000 ± 0.000	0.999	1.000	0.999 ± 0.003	0.999	1.000	0.999 ± 0.003	0.999	1.000	0.999 ± 0.003
Spinach	0.999	1.000	0.999 ± 0.003	1.000	1.000	1.000 ± 0.000	1.000	1.000	1.000 ± 0.000	0.999	1.000	0.999 ± 0.001

Table 4. Comparison of classification reports generated from 5-Fold cross-validation leaving $k = 1$ fold out while reducing training and augmenting test sample size. Where experiment 3 represents the largest test sample size Figure 11. The average precision, recall and F1-Score are reported for each class obtained with the ViT B-16 model.

Test Fold / **Classes**	**Experiment 1**			**Experiment 2**			**Experiment 3**		
	$\mu_{Precision}$	μ_{Recall}	$\mu_{F1-Score}$	$\mu_{Precision}$	μ_{Recall}	$\mu_{F1-Score}$	$\mu_{Precision}$	μ_{Recall}	$\mu_{F1-Score}$
Weeds	0.993	0.980	0.987 ± 0.005	0.987	0.981	0.984 ± 0.006	0.985	0.957	0.971 ± 0.004
Off-Type Beet	0.978	0.992	0.985 ± 0.006	0.979	0.984	0.982 ± 0.005	0.954	0.979	0.966 ± 0.003
Beet	0.998	1.000	0.999 ± 0.000	0.997	0.999	0.998 ± 0.003	0.994	0.999	0.997 ± 0.001
Parsley	1.000	1.000	1.000 ± 0.000	1.000	1.000	1.000 ± 0.000	0.997	1.000	0.998 ± 0.001
Spinach	1.000	1.000	1.000 ± 0.000	1.000	1.000	1.000 ± 0.000	1.000	1.000	1.000 ± 0.000

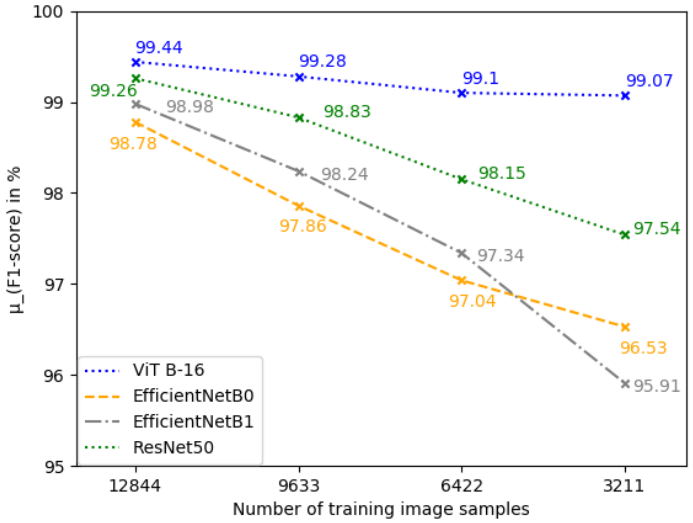

Figure 12. Comparison between ViT B-16, EfficientNet B0, EfficientNet B1 and ResNet50 on their respective performance with different number of training images.

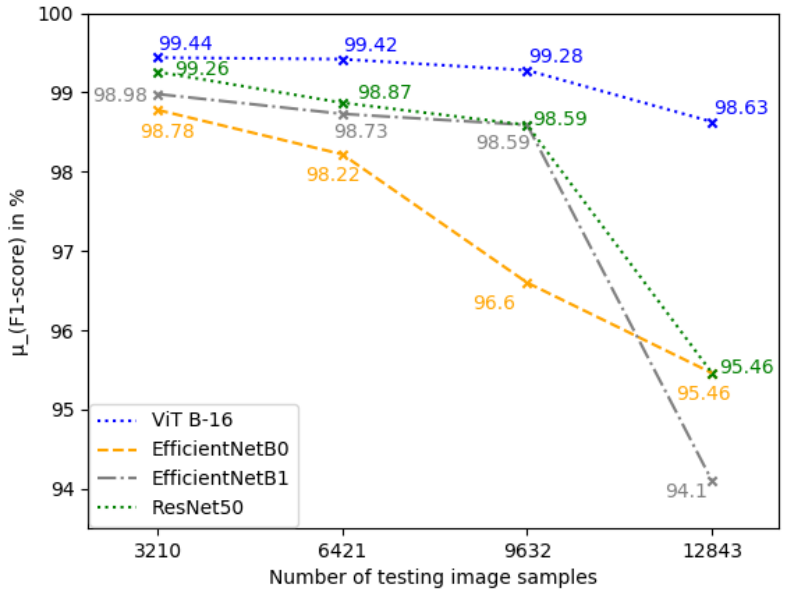

Figure 13. Comparison between ViT B-16, EfficientNet B0, EfficientNet B1 and ResNet50 on their respective performance with different number of testing samples while keeping 5-Fold cross-validation leaving one fold out as validation set.

5. Discussion

This study aimed to deploy and analyze self-attention deep learning approaches, in the context of a drone-based weed and crop recognition system. The classification models were evaluated on our aerial image dataset to select the best architecture. As discussed earlier, the ViT B-16 architecture achieved better performance compared to the CNN architectures. This observation implies that the self-attention mechanism may be more effective for weed identification because image patches are interpreted as units of information, whereas with CNN-based models, information is extracted via convolutional layers. Another observed advantage of interpreting images as information units, via self-attention, is the stable performance of the visual transformation model while reducing the number of training samples and increasing the number of test samples.

In the first workflow, all models studied achieved high accuracies and F1 scores, indicating that with a sufficient number of examples for each class, the difference in performance is not significant, and that the variations may be due to the dataset used for the experiments. Unfortunately, creating datasets large enough for weed identification by UAV, can be difficult depending on the crops being studied. Weeds must be removed from the field quickly by the grower and the costs of acquiring aerial images by drone can be high depending on the sensor and the area to be photographed.

In response to this difficulty, and in order to optimize future data acquisition campaigns. We decreased the number of training samples and increased the number of validation samples (Workflow 2). Reducing the number of training images while increasing the number of validation samples will force the model to extract general features for the images and track its training progress with a large number of validation samples. As can be observed in Figure 12, the performance of CNN models is proportional to the number of training samples while the performance of ViT is more stable. Moreover, the F1-score for each class predicted using the self-attention mechanism decreases only slightly and uniformly for all five classes, and does not decrease only for specific classes (Table 3). In addition to decreasing the training samples, in workflow 3 we increase the number of test samples and keep a fixed number of validation samples. Increasing the number of test samples from 3210 to 12,843 unseen samples simulates the behavior of the model as it would be in a production inference, as the larger the test set, the more representative it is. In this experimental setup, as summarized in Figure 13, the ViT B-16 model also maintains steady metric scores as the decrease for the CNN is greater the higher the number of test samples.

We have showed that applied to our five class agricultural dataset for weed identification, the ViT B-16 architecture pre-trained on ImageNet dataset outperforms other architectures and is more robust to a varying number of samples in the dataset. The application of the ViT for weed classification shows promising results for a limited number of classes. In futur experiments, we will add extra classes to cover more number of crop types. Adding extra classes will probably lower the classification top-1 score especially if classifying similar plants in shape and color. But should still yield better results than CNNs as the ViT was shown to be more robust.

There are also some limitations in the acquisition and preparation of the data sets. First, the data augmentations used are large, especially for the off-type beet class, where the rotation augmentations were applied before the training augmentations. On the other hand, the other augmentations performed during training facilitate model convergence and generalization by transforming the samples, which can represent different variations in outdoor brightness, for example. This may ensure the generalization capabilities of the models when the image acquisition conditions are similar to the augmentations performed. If the image acquisition conditions are very different, the models could lose score points, a most important environmental change may be photographing plants after a rain where the plants will not have the same vigor/shape as when they are capturing sunlight. Therefore, additional image acquisitions are planned for next season to address these different conditions.

6. Conclusions

In this study, we used the self-attention paradigm via the ViT (vision transformer) models to learn and classify custom crops and weed images acquired by UAV in beet, parsley and spinach fields. The results achieved with this dataset indicate a promising direction in the use of vision transformers with transfer learning in agricultural problems. Outperforming current state-of-the-art CNN-based models like ResNet and EfficientNet, the base ViT model is to be preferred over the other models for its high accuracy and its low computation cost. Furthermore, the ViT B-16 model has proven better with its high performance specially with small training datasets where other models failed to achieve such high accuracy. This shows how well the convolutional-free, ViT model interprets an image as a sequence of patches and processes it by a standard transformer encoder,

using the self-attention mechanism, to learn patterns between weeds and crops images. It is worth mentioning that certain findings of the current study do not support some previous researches, where it is indicated the transformers perform better only with large datasets. It is possible that the high performance obtained here with small dataset is due to the low number of classes, transfer learning, and data augmentation. In this respect, we come to conclusion that the application of vision transformer could change the way to tackle vision tasks in agricultural applications for image classification by bypassing classic CNN-based models. Despite these promising results, questions remain, such as the viability of the vision-transformers in the recognition task after a significant change in the image acquisition conditions in the fields (resolutions, luminosity, plant development phase, etc.), large number of plant classes, etc. Further research should be undertaken to study these aspects. In future works, we plan to use vision transformer classifier as a backbone in an object detection architecture to locate and identify weeds and plants on UAV orthophotos, with different acquisition conditions.

Author Contributions: Conceptualization, E.D., R.C. In addition, A.H.; methodology, R.R. In addition, A.H.; software, R.R.; validation, R.R., E.D., R.C. In addition, A.H.; formal analysis, R.R., E.D., R.C. In addition, A.H.; investigation, E.D., R.C. In addition, A.H.; data acquisition, E.D.; writing—original draft preparation, R.R.; writing—review and editing, E.D., R.C. In addition, A.H.; visualization, R.R.; supervision, A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Region Centre-Val de Loire France.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This work was carried out as a part of DESHERBROB project. We gratefully acknowledge Region Centre-Val de Loire for its support. We thank FRASEM company, partner of this project for their valuable provision of plots and data.

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

References

1. Radoglou-Grammatikis, P.; Sarigiannidis, P.; Lagkas, T.; Moscholios, I. A compilation of UAV applications for precision agriculture. *Comput. Netw.* **2020,** *172*, 107148. [CrossRef]
2. Ustuner, T.; Sakran, A.; Almhemed, K. Effect of Herbicides on Living Organisms in The Ecosystem and Available Alternative Control Methods. *Int. J. Sci. Res. Publ. (IJSRP)* **2020**, *10*, 633641. [CrossRef]
3. Patel, D.D.; Kumbhar, B.A. Weed and its management: A major threats to crop economy. *J. Pharm. Sci. Biosci. Res.* **2016**, *6*, 453–758.
4. Iqbal, N.; Manalil, S.; Chauhan, B.; Adkins, S. Investigation of alternate herbicides for effective weed management in glyphosate-tolerant cotton. *Arch. Agron. Soil Sci.* **2019**, *65*, 1885–1899. [CrossRef]
5. Vrbničanin, S.; Pavlović, D.; Božić, D. Weed Resistance to Herbicides. In *Herbicide Resistance in Weeds and Crops*; IntechOpen Limited: London, UK, 2017. [CrossRef]
6. A. Wang, W.Z.; Wei, X. A review on weed detection using ground-based machine vision and image processing techniques. *Comput. Electron. Agric.* **2019**, *158*, 226–240. [CrossRef]
7. Wu, X.; Aravecchia, S.; Lottes, P.; Stachniss, C.; Pradalier, C. Robotic weed control using automated weed and crop classification. *J. Field Robot.* **2020**, *37*, 322–340. [CrossRef]
8. Donmez, C.; Villi, O.; Berberoglu, S.; Cilek, A. Computer vision-based citrus tree detection in a cultivated environment using UAV imagery. *Comput. Electron. Agric.* **2021**, *187*, 106273. [CrossRef]
9. Bah, M.D.; Hafiane, A.; Canals, R. CRowNet: Deep Network for Crop Row Detection in UAV Images. *IEEE Access* **2020**, *8*, 5189–5200. [CrossRef]
10. Huang, H.; Deng, J.; Lan, Y.; Yang, A.; Deng, X.; Zhang, L. A fully convolutional network for weed mapping of unmanned aerial vehicle (UAV) imagery. *PLoS ONE* **2018**, *13*, e0196302. [CrossRef]
11. Huang, H.; Lan, Y.; Yang, A.; Zhang, Y.; Wen, S.; Deng, J. Deep learning versus Object-based Image Analysis (OBIA) in weed mapping of UAV imagery. *Int. J. Remote Sens.* **2020**, *41*, 3446–3479. [CrossRef]
12. Petrich, L.; Lohrmann, G.; Neumann, M.; Martin, F.; Frey, A.; Stoll, A.; Schmidt, V. Detection of Colchicum autumnale in drone images, using a machine-learning approach. *Precis. Agric.* **2020**, *21*, 1291–1303. [CrossRef]

13. Puerto, A.; Pedraza, C.; Jamaica-Tenjo, D.A.; Osorio Delgado, A. A Deep Learning Approach for Weed Detection in Lettuce Crops Using Multispectral Images. *AgriEngineering* **2020**, *2*, 471–488. [CrossRef]
14. Ramirez, W.; Achanccaray Diaz, P.; Mendoza, L.; Pacheco, M. Deep Convolutional Neural Networks for Weed Detection in Agricultural Crops using Optical Aerial Images. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *XLII-3/W12-2020*, 551–555. [CrossRef]
15. Patidar, S.; Singh, U.; Sharma, S.; Himanshu. Weed Seedling Detection Using Mask Regional Convolutional Neural Network. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020.
16. Sa, I.; Chen, Z.; Popovic, M.; Khanna, R.; Liebisch, F.; Nieto, J.; Siegwart, R. WeedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming. *IEEE Robot. Autom. Lett.* **2017**, *3*, 588–595. [CrossRef]
17. Sa, I.; Popovic, M.; Khanna, R.; Chen, Z.; Lottes, P.; Liebisch, F.; Nieto, J.; Stachniss, C.; Walter, A.; Siegwart, R. WeedMap: A Large-Scale Semantic Weed Mapping Framework Using Aerial Multispectral Imaging and Deep Neural Network for Precision Farming. *Remote Sens.* **2018**, *10*, 1423. [CrossRef]
18. dos Santos Ferreira, A.; Matte Freitas, D.; Gonçalves da Silva, G.; Pistori, H.; Theophilo Folhes, M. Weed detection in soybean crops using ConvNets. *Comput. Electron. Agric.* **2017**, *143*, 314–324. [CrossRef]
19. Milioto, A.; Lottes, P.; Stachniss, C. Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *IV-2/W3*, 41–48. [CrossRef]
20. Sivakumar, A.N.V.; Li, J.; Scott, S.; Psota, E.; Jhala, A.J.; Luck, J.D.; Shi, Y. Comparison of object detection and patch-based classification deep learning models on mid-to late-season weed detection in UAV imagery. *Remote Sens.* **2020**, *12*, 2136. [CrossRef]
21. Kang, J.; Liu, L.; Zhang, F.; Shen, C.; Wang, N.; Shao, L. Semantic segmentation model of cotton roots in-situ image based on attention mechanism. *Comput. Electron. Agric.* **2021**, *189*, 106370. [CrossRef]
22. Kerkech, M.; Hafiane, A.; Canals, R. Vine disease detection in uav multispectral images with deep learning segmentation approach. *Comput. Electron. Agric.* **2020**, *174*, 105446. [CrossRef]
23. Hamuda, E.; Glavin, M.; Jones, E. A survey of image processing techniques for plant extraction and segmentation in the field. *Comput. Electron. Agric.* **2016**, *125*, 184–199. [CrossRef]
24. Sabzi, S.; Abbaspour-Gilandeh, Y.; Arribas, J.I. An automatic visible-range video weed detection, segmentation and classification prototype in potato field. *Heliyon* **2020**, *6*, e03685. [CrossRef] [PubMed]
25. Saha, D. Development of Enhanced Weed Detection System with Adaptive Thresholding, K-Means and Support Vector Machine. In *Electronic Theses and Dissertations*; South Dakota State University: Brookings, SD, USA, 2019; p. 49. Available online: https://openprairie.sdstate.edu/cgi/viewcontent.cgi?article=4399&context=etd (accessed on 10 November 2021).
26. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
27. Wu, Z.; Chen, Y.; Zhao, B.; Kang, X.; Ding, Y. Review of Weed Detection Methods Based on Computer Vision. *Sensors* **2021**, *21*, 3647. [CrossRef]
28. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
29. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–80. [CrossRef]
30. Hasan, A.S.; Sohel, F.; Diepeveen, D.; Laga, H.; Jones, M.G. A survey of deep learning techniques for weed detection from images. *Comput. Electron. Agric.* **2021**, *184*, 106067. [CrossRef]
31. Lecun, Y. Generalization and network design strategies. *Connect. Perspect.* **1989**, *19*, 143–155.
32. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2012**, *60*, 84–90. [CrossRef]
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
34. Lecun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990; pp. 396–404.
35. Nkemelu, D.; Omeiza, D.; Lubalo, N. Deep Convolutional Neural Network for Plant Seedlings Classification. *arXiv* **2018**, arXiv:1811.08404.
36. Suh, H.K.; IJsselmuiden, J.; Hofstee, J.W.; van Henten, E.J. Transfer learning for the classification of sugar beet and volunteer potato under field conditions. *Biosyst. Eng.* **2018**, *174*, 50–65. [CrossRef]
37. Dian Bah, M.; Hafiane, A.; Canals, R. Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote Sens.* **2018**, *10*, 1690. [CrossRef]
38. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.
39. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
40. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]

41. Hu, D. An Introductory Survey on Attention Mechanisms in NLP Problems. In Proceedings of the IntelliSys, London, UK, 5–6 September 2019. [CrossRef]
42. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Available online: http://papers.neurips.cc/paper/7181-attention-is-all-you-need.pdf (accessed on 10 November 2021).
43. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in Vision: A Survey. *arXiv* **2021**. [CrossRef]
44. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929.
45. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
46. Cubuk, E.D.; Zoph, B.; Shlens, J.; Le, Q.V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 3008–3017. [CrossRef]
47. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*. [CrossRef]
48. Cheng, J.; Dong, L.; Lapata, M. Long Short-Term Memory-Networks for Machine Reading. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, USA; 1–5 November 2016; pp. 551–561. [CrossRef]
49. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, CA, USA, 10–15 June 2019.
50. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2020**, arXiv:1905.11946.
51. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*. [CrossRef]
52. Molinaro, A.M.; Simon, R.; Pfeiffer, R.M. Prediction error estimation: a comparison of resampling methods. *Bioinformatics* **2005**, *21*, 3301–3307. [CrossRef]
53. Tharwat, A. Classification assessment methods. *Appl. Comput. Inform.* **2020**, *17*, 168–192. [CrossRef]
54. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]
55. Sokolova, M.; Lapalme, G. Performance Measures in Classification of Human Communications. *Advances in Artificial Intelligence* **2007**, *4509*, 159–170. [CrossRef]