Classifying Cover Crop Residue from RGB Images: a Simple SVM versus a SVM Ensemble

Parth C. Upadhyay[1], Lokesh Karanam[2], John A. Lory[3], Guilherme N. DeSouza[4]

Abstract—**Plant-derived crop residue on soil surface provides many important advantages including preventing erosion and conserving soil moisture. In that sense, making accurate determination on the percent of crop residue cover using RGB images can be a fundamental tool in protecting the soil. In our research, we approach the determination of such percentages as a classification problem, and in this paper, we compare two of these approaches. Both approaches relied on support vector machines (SVM) as the classifier of choice, and the same set of features, which were selected in our previous studies on the same topic. In this paper we developed a SVM ensemble with a hierarchical structure and compared it against a single, multi-class SVM classifier. In the SVM ensemble framework, four two-class SVMs and one five-class SVM were combined in sequence to better separate adjacent levels of residue cover. The rationale of the ensemble was to allow each of the two-class SVMs to find the hyperplanes that maximize the margin between the corresponding two consecutive classes. Then, based on the distance of the samples to these hyperplanes, probabilistic estimates of the data-point belonging to the class were computed and added as extra inputs for the last SVM. In order to enhance the performance of the ensemble, other considerations such as the use of Grid Search method for optimizing the hyperparameters were employed in the tuning of the SVMs. Numerical experiments were conducted over a dataset of 4,400 images, which were collected from 88 locations in 40 row crop fields in five Missouri counties between mid-April and early July in 2018 and 2019. The images were collected using a camera mounted on a tripod, with a spatial resolution of 0.014 cm pixel^{-1} GSD (Ground Sampling Distance). The experiments highlighted the better performance of the proposed hierarchical ensemble classifier, which achieved a cross-validation accuracy of 86.3% vs an accuracy of 80.4% for the single SVM, while the testing accuracy was 83.8% when compared to the accuracy of 80.9% from the single SVM. Other metrics, such as precision, recall and F1 score, were also highly favorable towards the ensemble SVM.**

I. Introduction

Crop residue – or the remainder plant material left on the surface of the soil after cultivation – plays an important role in protecting soil. Benefits of more residue include increasing soil carbon sequestration, reducing water-based and wind-based soil erosion, and improving the quality of surface water runoff [1], [2]. Consequently, maintaining better crop residue level provide significant increment of crop yields [3]. Considering these advantages, in the US, The Food Security Act of 1985 established a requirement to maintain "sustainable erosion rates" on cropland, hayland and pasture defined as "highly erodible land" (HEL). The Natural Resource Conservation Service (NRCS) makes determination if an agricultural tract is HEL complaint and these determinations have economic consequences when farmers fail conservation-plan-compliance assessments, which can in turn lead to Federal agriculture support payments being withheld by the Farm Service Agency (FSA) [4]. Residue assessment is part of this compliance protocol.

Line-transect method is the most common technique performed by NRCS employees to assess residue cover in the field [5]. There are some other conventional methods such as: the meter stick, spiked wheel, visual estimate, photograph comparison, and photographic-grid [6]–[9]. However, there are well-known issues with these methods, including the fact that they are time consuming, reader biased, labor intensive, and unable to provide continuous estimates of residue across the field [7]–[10]. So, a reliable automated approach that eradicates the tediousness and subjectivity of the current methods for crop residue estimation would be of great benefit to farmers and government technical staff.

There has been substantial research performed on multispectral imagery to map crop residue cover in agricultural fields [11]–[13], whereas there has been a comparatively limited research done on the RGB imagery of the same crop residue cover. Computer vision based methods, like color-based threshold segmentation were found to be effective for segmenting the plant structures [14], but they do not perform well in general in under various lighting conditions, and they are only suitable for segmentation under one dominant color, e.g. green, hence they are not very effective for segmenting crop residue [15] since soil and residue do not have very distinctive colors. The better alternative is the use of a machine learning approach, that may use feature descriptors to represent image components, which can then be classified using random forest (RF), support vector machine (SVM), etc. [15], [16].

The usefulness of SVM had already been established for crop/weeds identification in maize fields as well as to estimate the plant density of wheat crops [17], [18]. For example, in our previous work [16], we showed that a machine learning approach using a three-class SVM classifier has been found effective with color and texture features for the residue estimation from RGB imagery at different GSDs (Ground Sampling Distance). In this problem, each class belongs to different range of percentage residue, that makes it difficult to find classification boundaries among classes. In general, maximum misclassification happens with the data-points which are nearer

Parth C. Upadhyay, Lokesh Karanam and Guilherme N. DeSouza are with the Vision-Guided and Intelligent Robotics (ViGIR) Laboratory in the Department of Electrical Engineering and Computer Science (EECS). John A. Lory is with the Division of Plant Sciences: all at the University of Missouri, Columbia, MO, 65211. (email: pcuv48@mail.missouri.edu[1], lkdw6@mail.missouri.edu[2], loryj@missouri.edu[3], desouzag@missouri.edu[4])

to the percentage residue boundaries for the class. So firstly, by learning classification boundary between consecutive classes to find the probability estimates of the data-points and then propagate such information to the final multi-class classifier should be helpful. So, inspired by such idea, we now propose an ensemble of SVM classifier with a hierarchical structure. Also, since dividing the residue cover percentages into five intervals/classes would lead to a more reasonable assessment of the field, four two-class SVMs were used to perform binary classification between consecutive intervals, followed by a five-class SVM classifier. Our goal and main contribution of this paper was to show that these classifiers can be ”tuned” to better estimate the hyperplanes that separate two consecutive classes. Also, as an addition contribution since our last study, we used the distance of the data-points to these same hyperplanes as probabilistic estimates of the data-points belonging to each individual class. These probabilistic estimates were used as a set of additional features by a five-class SVM to perform the final, multiclass classification of the field residue level (i.e. 5 classes instead of the previous 3).

The rest of the paper is organized as follows. In Section 2, we overview about data collection and ground truthing, and details post-data collection machine learning methods including different image feature types, Recursive Feature Elimination with Cross Validation using Support Vector Machine (RFECV-SVM) based feature selection, Grid Search cross validation based hyper parameter optimization of SVM classifier, and ensemble SVM classifier design. In section 3, numerical experiments on collected datasets, as well as comparisons between a simple SVM classifier and ensembled SVM classifier are presented. Section 4 concludes the paper.

II. Methodology

A. Data Collection

Images were collected from 40 fields in five central Missouri counties after NRCS personnel got permission for the project team to access farmer fields, during early May through early July in 2018 and 2019 in arid fields planted to corn (Zea mays [L.]) or soybean (Glycine max [L.] Merr.) prior to corn reaching growth stage V4 and soybean reaching growth stage V3 in the weather condition ranges from very cloudy to full sunny. At each field location, project staff and NRCS personnel placed one to four 15.24-m (50 ft) tapes at 45 degrees to the planted row direction based on NRCS criteria. Images were obtained 1.0-m above the ground surface over the tapes on a tripod-mounted Canon EOS Rebel T6i digital single lens reflex (DSLR) camera (Canon USA, Melville, NY) with a 24 mm stepper motor technology lens and 24.2 MP resolution which generated image size of 6000 × 4000 pixels. Estimated GSD for these images was 0.014 cm pixel^{-1} and typically 51 images per tape were obtained by moving the tripod 30 cm between images. Final dataset resulted in imagery from 60 locations (tapes) in 2018 (32 planted to corn and 28 planted to soybean) and 28 locations in 2019 (6 planted to corn and 22 planted to soybean) to be used in this project.

From the captured images over each tape, starting at the zero point on the tape, 50 sequential, unique region of interest (ROI) images were created from along the tape in 0.305-m (≈1-ft) increments as can be seen in Fig. 1. Unique ROI images were created by cropping an area representing 0.305-m (1-ft) wide by 0.20-m (0.66-ft) high area from the captured image using the software package Photoshop (Adobe Inc., San Jose, CA). These ROI images were obtained contiguous to the tape from the side of the tape with lesser tape shadow (to avoid tape shadow in the ROI image dataset) using the captured image where the section of interest was most central to the image to minimize the parallax effect. Each cropped unique image contained approximately 2,400 × 1,600 pixel area (approximately 0.305-m × 0.020-m surface area of the soil). This resulted in a dataset of 4400 images (88 tape locations × 50 image per location) for each GSD. A bullseye grid method as in [10] with n=100 grid points was used on 0.014 GSD cropped images to obtain image-wise estimates of residue cover used for ground truth.

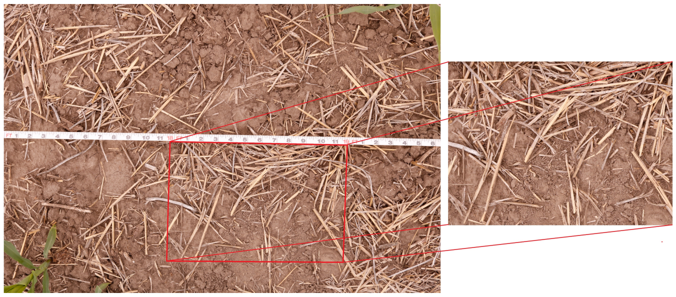


Fig. 1: Unique ROI image cropped from the collected image

Based on the residue cover ground truth of the images, they were then divided into five classes constructed on residue range (Table I).

TABLE I: Number of Crop Residue Images per Class

Class No.	Class (Residue Range)	Number of Images
Class-1	Lower (≤ 20%)	1766
Class-2	Lower-Medium (20% to 40%)	775
Class-3	Medium (40% to 60%)	549
Class-4	Higher-Medium (60% to 80%)	491
Class-5	Higher (≥ 80%)	819

B. Pipeline for Classification

The pipeline for post data collection for the classification of crop residue from the RGB images is summarized in Fig. 2. The pipeline consisted of three main steps: first, we extracted a large set of color and texture features from the entire dataset; we normalized them using a min-max scaling method; and divided the full dataset into training and testing sets, with a 80:20 ratio. In the second step, a Recursive Feature Elimination using Support Vector Machine (RFE-SVM) method was applied to the training set to determine an optimized set of features to be used for classification. Next, the selected features were used in a Grid Search to find the optimized hyper parameters of the single, multi-class SVM classifier. Finally, the single SVM and ensemble SVM classifiers were trained using the same training dataset and the selected features from the previous steps. These classifiers were validated using a 10-fold cross validation, and then tested on the remaining testing dataset. All machine learning operations were performed using the Jupyter Notebook (ver. 6.2.0) software toolkit. In the next sections, we will explain in greater details the steps of this pipeline.

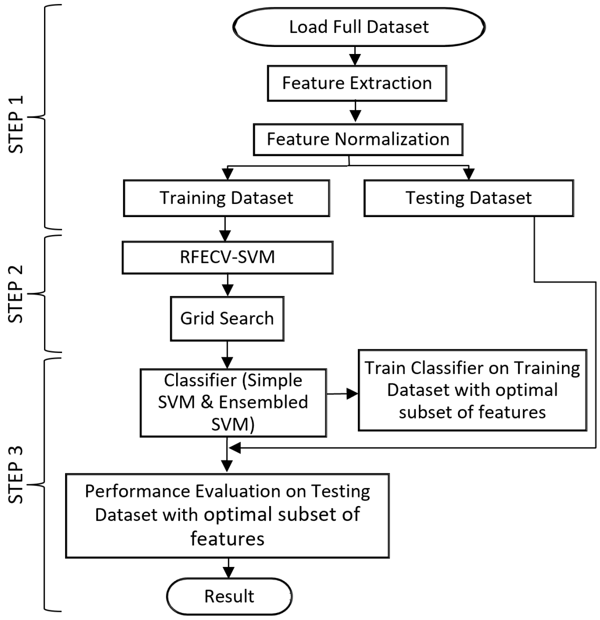

STEP 1
STEP 2
STEP 3
Load Full Dataset
Feature Extraction
Feature Normalization
Training Dataset
Testing Dataset
RFECV-SVM
Grid Search
Classifier (Simple SVM & Ensembled SVM)
Train Classifier on Training Dataset with optimal subset of features
Performance Evaluation on Testing Dataset with optimal subset of features
Result

Fig. 2: Pipeline for crop residue classification from RGB images

C. Features Extraction and Selection

Following our previous study [16], we selected and extracted eighty-one feature descriptors from all images in the dataset. The features comprised: twenty-four color, thirteen global texture and forty-four local texture features. The twenty-four color features were comprised of the image-wise mean, median, standard deviation and skewness of each of the color bands from the HSV and CIE-LAB color-spaces. The thirteen global texture features were extracted based on the gray-level co-occurrence matrix (GLCM) as defined by Haralick, which are: angular second moment, contrast, correlation, sum of squares (variance), inverse difference moment, sum average, sum variance, sum entropy, entropy, difference variance, difference entropy, information measure of correlation, and information measure of correlation squared [19]. The forty-four local texture features were represented using local binary pattern (LBP) in which, firstly, one of eighteen uniform patterns were calculated based on 16 equally spaced pixels in a circularly symmetric neighborhood around the central pixel, with two pixels radius. A similar operation, but now with a three pixels radius, provided one of twenty-six uniform patterns based on the now 24 equally spaced pixels. Consequently, histogram bins developed from the images that were created from these uniform patterns were used as the LBP features [20]. Unlike in our previous study, here we added 18 extra LBP features. These additional features were found to lead to better results for the five-class problem with the addition of radius-2 based LBP features. Once again, all image features were normalized to a range of [0,1] using the min-max scaling normalization method.

Finally, it is very important to find the subset of most relevant features and avoid redundant and irrelevant features contained in the entire set of extracted features. This helps the model accuracy and leads to optimal classification result. So, feature selection was performed using the RFE-SVM method, which showed superior results over other methods [16]. In a nutshell, RFE-SVM is a wrapper method, that implements a backward selection learning scheme to evaluate feature sets, and the accuracy of the learning scheme is estimated using k-fold cross-validation to find the best subset of features [21]. We used k=10 and SVM classifier with linear kernel as an external estimator.

D. SVM Classifier

As the reader must well know, SVM is a supervised machine learning method developed by Vapnik et al. in the mid 1990s [22]. The general and typical principle behind SVM is to project the data into a higher-dimensional, non-linear feature space using a kernel function, and then find the separating hyperplane that maximizes the margin between the data. Thus, SVM can handle a linearly inseparable problem as a linear separable problem by means of the kernel functions. Another core idea in SVM is in finding the set of nearest data points to the hyperplane, which are called the support vectors. Fig. 3 shows an example of a separating plane in two dimensions and the support vectors that define the same plane. In many applications, the radial basis function (RBF) is a good choice for kernel for its better performance when compared to other kernels, such as the linear kernel [23] or the sigmoid kernel [24]. Moreover, since the RBF kernel has fewer hyperparameters, it also possesses comparatively lower numerical complexity than polynomial kernels and their larger number of hyperparameters to be optimized. For those reasons, we chose the RBF kernel for this study, and optimized its two parameters: C and γ, where C is a penalty (or regularization) parameter and γ is a parameter that determines the radius of influence of the data.

The advantages of the SVM classifier include its enhanced generalization properties and its efficiency without direct dependence on the dimension of feature space. However, SVM was conceived for binary classification tasks, rendering it conceptually useless for multi-class classification problems. This issue has been resolved by using SVM in either an *one-against-all* or a *pair-wise* fashion – in the first case, multiple hyperplanes are obtained to separate one class from the remaining classes all together; while in the second, multiple hyperplanes are built to separate every two classes in a pair-wise manner. While both of these methods have advantages and disadvantages and the best practice is still a matter of research [25], here, we propose an ensemble SVM as a

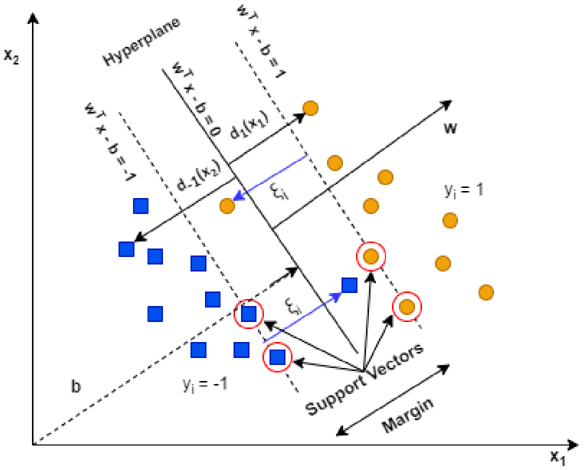


Fig. 3: Classification of dataset with two classes by support vector machine (SVM) and its geometric interpretations

method for addressing many classification problems that are not addressed by typical SVM: e.g. unbalanced data; partially noisy data; missing data; etc. In such cases, a single multi-class SVMs should lead to decrease accuracy, while the ensemble can be ”tuned” for the pair-wise classification of consecutive classes. Also, another major difference between a traditional pair-wise fashion for multi-class SVM and the one proposed here is that a traditional pair-wise approach takes into account all possible pairs. However, in this application it is very unlikely that a classifier would (or should) confuse, say, a field with 10% residue cover with one with 80% residue cover.

E. SVM Mathematical Definitions

As mentioned above, SVM was designed for two-class problems. So, in order to construct an optimal separating hyperplane with higher classification accuracy, SVM uses the structural risk minimization (SRM) approach. In other words, SVM seeks a hyperplane that is not just any separation between two classes, but one that also maximizes the separating margin [26]. Consider a training dataset consists of N number of d-dimensional samples – i.e. $x_i \in R^d$ (i = 1, 2, . . . , N) from the d-dimensional feature space X, where target label $y_i \in \{$-1, +1$\}$ is associated with each vector x_i. Since, the goal of SVM is to find the hyperplane that maximizes the margin, the solution can be expressed as the following optimization problem:

$$min\frac{1}{2} \parallel W \parallel^2 + C\sum_{i=1}^{N}\xi_i \qquad where,\ i=1,2,...,N \tag{1}$$

subject to

$$y_i[W\cdot K(x_i,x_j)+b] \geq 1-\xi_i \quad and \quad \xi_i \geq 0 \tag{2}$$

where ξ_i are slack variables, which represent the distance between the misclassified sample and the optimal hyperplane (Fig. 3), where C is the penalty parameter that allows to balance the model complexity and penalizing the non-zero ξ_i. The bias b is a scalar, that represents the bias of the hyperplane, and W is the weight vector, that defines a perpendicular direction to the hyperplane (Fig. 3). Function $K(x_i,x_j)$ is the kernel function, and in our case, it is a Gaussian RBF, which can be expressed as:

$$K(x_i,x_j) = e^{-\|x_i-x_j\|^2/2\gamma^2} \tag{3}$$

From the equations (1), (2) and (3), it is clear that the optimization problem of SVM requires a pair of parameters, C and γ. Specifically, penalty factor C characterizes the trade-off between the complexity and classification accuracy of the classifier, whereas γ controls the radial effect range (or influence) of the data. The popular and commonly used optimization method of SVM classifier is grid search method accompanied by a k-fold cross-validation [27].

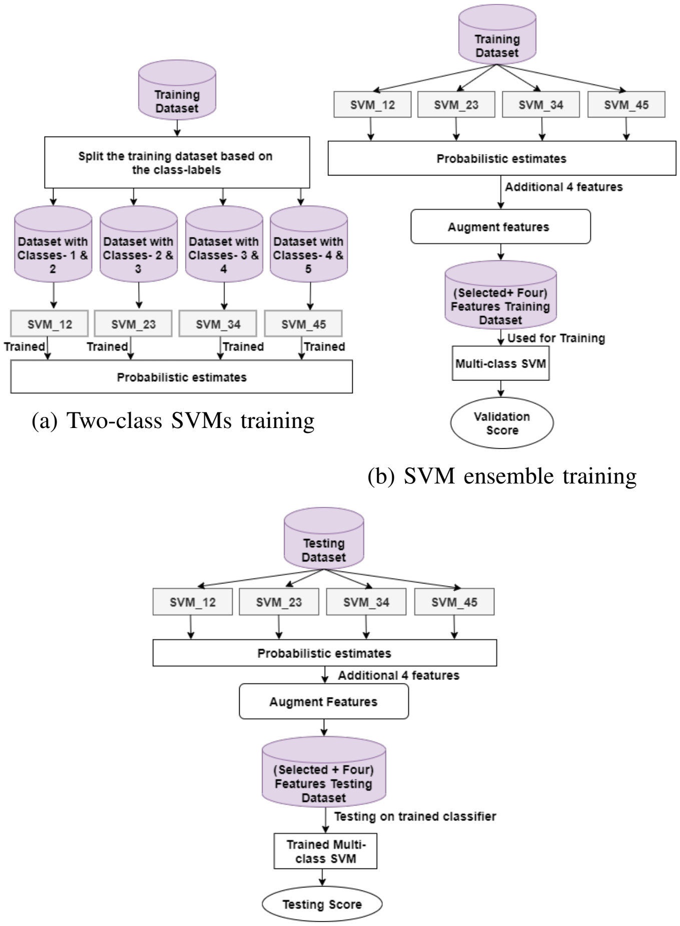
1) Grid Search Method: Grid search is a method for the hyperparameter optimization that exhaustively search based on defined subset of the hyperparameter space [28]. The goal of this step is to identify better (C, γ) so that the classifier can accurately predict unknown testing data. We used k-fold cross-validation to evaluate the performance of every combination with k=5. In this method, various pairs of (C, γ) values were tried and the one with the best cross-validation accuracy was picked.

F. Ensembled SVM

In general, ensemble learning approaches in machine learning require multiple base classifiers to be trained and then aggregated to construct the final classifier. The result is a classifier with stronger generalization power than that of a single classifier [29]–[31]. In this same sense, we developed a hierarchical SVM-ensemble classification model under the following assumptions: 1) SVM was conceived for binary classification and the traditional one-against-all and pair-wise methods for multi-class SVM cannot capture the nuances of our application (i.e. the smaller number of pair-wise cases that need to be considered; and the imbalance between classes); and 2) an ensemble approach allows us to change the feature space to be used by the individual classifiers in the ensemble, leading to an even better ”tuning” of the individual classifiers, which should also further increase the accuracy and generalization power of the final solution.

Here, the idea was to add four extra relevant features to the input of the final multi-class SVM classifier derived from the four two-class SVM classifiers, as it is shown in Fig. 4. To understand these extra features in the proposed SVM ensemble, it is important to understand the output of an SVM under a probabilistic point of view.

1) Outputs as Probabilistic Estimates: One of the major advantages of a SVM is that its outputs can be regarded as a probabilistic estimate of the data-point belonging to each class. In this paper, such probabilistic estimates were computed for all four two-class (binary) SVMs and fed as features for the final multi-class SVM in the ensemble. To compute these probabilistic estimates, the distances $d_k(x)$ of each data-point x to the separating hyperplane corresponding to the pair of



(a) Two-class SVMs training

(b) SVM ensemble training

(c) SVM ensemble testing

Fig. 4: Training and Testing Algorithm for the proposed SVM ensemble for crop residue level classification

classes *k* were computed, where *k*=1 corresponds to the pair of classes (1,2); *k*=2 to the pair (2,3), and so forth. Based on these distances $d_k(x)$, probabilistic estimates $p_k(x)$ of a data-point belonging to one of the classes in the pair *k* were then calculated using a sigmoid function [32] with equation:

$$p_k(x) = \frac{1}{1 + e^{-(A_k \cdot d_k(x) + B_k)}} \tag{4}$$

where, A_k and B_k are parameters estimated for the two-class SVM for the pair of classes *k* by minimizing the mean square error between the original label and the output of the sigmoid function on the training data. The reader should notice that if, according to the two-class SVM for, say, the pair of classes *k*=1 (i.e. SVM_12), a point *x* has a probabilistic estimate $p_1(x)$ of belonging to class 1, then that same point has probabilistic estimate *1*-$p_1(x)$ of belonging to class 2 according to the same two-class SVM_12. For that reason, only four additional features were needed to augment the feature vector going into the final, multi-class SVM in the ensemble.

2) Procedure for Training and Testing: The procedure used for training and testing the SVM ensemble consisted of three main steps. The detailed process is illustrated in Fig. 4a-c.

- Step 1: The training data was split into four separate datasets, which were used to train the individual two-class SVMs classifiers (Fig. 4a). The split-datasets were formed by the same selected features obtained from using the single SVM classifier, as explained earlier and illustrated in Fig. 2. The number of samples, ($x_i \in R^d$), in each class are presented in Table II.

TABLE II: Combined Training and Testing Dataset for Each Binary SVM Classifier

k	Binary SVM Classifier	Dataset with Classes	Number of Images in each Dataset
1	SVM_12	Classes-1 & 2	2033
2	SVM_23	Classes-2 & 3	1059
3	SVM_34	Classes-3 & 4	832
4	SVM_45	Classes-4 & 5	1048

- Step 2: The entire training dataset was now fed into the four already-trained two-class SVMs – Fig. 4b. Next, probabilistic estimates $p_k(x)$ were computed using eq. (4) and two probabilistic values were obtained, as shown in Fig. 5 – even though only the first one was kept. Therefore, for each data-point, four new features representing $p_1(x)$, $p_2(x)$, $p_3(x)$ and $p_4(x)$ were added to the feature vector, and the final multi-class SVM was trained (Fig. 4b).

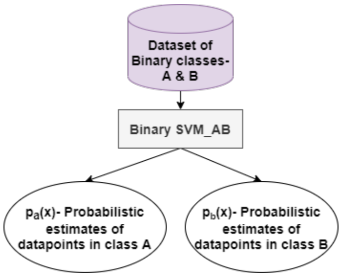

Dataset of Binary classes- A & B
Binary SVM_AB
Pa(x)- Probabilistic estimates of datapoints in class A
Pb(x)- Probabilistic estimates of datapoints in class B

Fig. 5: Probabilistic estimates of data-points based on binary-class SVM

- Step 3: Similarly to the previous step, the testing dataset with the augmented features was now fed into the two-class SVMs, ad then into the final multi-class SVM classifier leading to one of five possible predicted classes (Fig. 4c).

G. Performance Evaluation Methods

Four performance evaluation metrics were computed based on the 10-fold cross-validation score: accuracy, precision, recall, and F_1 score. Also, the confusion matrices for the single SVM and for the ensemble of SVMs were computed and are presented here. These metrics are defined as follows:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \tag{5}$$

$$Precision = \frac{(TP)}{(TP+FP)} \tag{6}$$

$$Recall = \frac{(TP)}{(TP+FN)} \tag{7}$$

$$F_1\ Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)} \tag{8}$$

Where, *TP*, *TN*, *FP* and *FN* are the number of true positives, true negatives, false positives and false negatives estimated by the classifiers, respectively.

III. Experimental Results and Discussion

As explained earlier, the first experiment performed was for feature selection. Fig. 6 shows the relationship between the mean 10-fold cross-validation score and the number of features included in the model based on the RFE-SVM feature selection method. It was observed that the mean 10-fold cross-validation score reached the highest value when the number of features was 42. Therefore, this optimized subset of 42 features was used for all subsequent experiments, including the single SVM classifier, and the ensemble of SVMs.

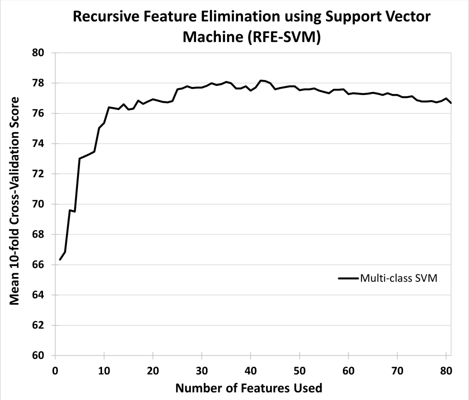


Fig. 6: RFE-SVM based features selection

After feature selection, grid search was applied to optimize the hyper-parameters of the classifiers. In this case, a 10-fold cross-validation was also used and pairs (C,γ) were determined for each classifier. As expected, the unique characteristics of each classifier resulted in different optimal (C,γ) pairs, which are summarized in Table III.

TABLE III: Hyper-Parameters For Each SVM Classifier After Grid-Search

No.	SVM Classifier	C	γ
1	SVM_12	10	1
2	SVM_23	1000	1
3	SVM_34	1000	0.1
4	SVM_45	10	1
5	Ensembled SVM (Master)	1	1
6	Simple SVM	1000	0.1

Next, we evaluated the performance of the two-class SVM classifiers by performing 10-fold cross validation and then observing their effectiveness – i.e. to detect if over-fitting occurred during model training. Table IV reports those results, and as it can be observed, none of the two-class SVM classifiers presented problems, but instead were satisfactorily trained.

TABLE IV: Training Accuracy and 10-Fold Cross-Validation Scores of Individual Two-Class SVM Classifiers

Metrics / Models	Training Score	Cross Validation Score
SVM_12	90.2	89.4
SVM_23	85.8	83.2
SVM_34	81.7	79.1
SVM_45	86.9	85.2

Further, we evaluated and compared the performances of the simple SVM versus the ensembled SVM using the already mentioned metrics: accuracy, precision, recall, and F1 score. Table V presents those results using cross validation, whereas Table VI reports the performance of the two methods using the testing dataset. The comparisons shown in both tables indicate, the ensembled SVM is superior than the simple SVM.

TABLE V: 10-Fold Cross-Validation Scores of Ensembled SVM and Simple SVM

Models / Metrics	Ensembled SVM	Simple SVM
Accuracy	86.3	80.4
Precision	86.8	81.5
Recall	86.3	80.4
F1 Score	86.5	80.9

TABLE VI: Performance of Ensembled SVM and Simple SVM on Final Test Dataset

Models / Metrics	Ensembled SVM	Simple SVM
Accuracy	83.8	80.9
Precision	84.1	81.8
Recall	84.1	80.9
F1 score	84.0	81.2

Finally, Tables VII and VIII depict the confusion matrices that summarizes the average performance of the simple SVM and ensembled SVM, respectively. As these tables show, the confusion matrices also support the claim that ensembled SVM outperformed the simple SVM. Based on these numbers, we notice that the SVM ensemble produced a small, but still relevant gain of up to 6% compared to the simple SVM.

TABLE VII: Confusion Matrix for Simple SVM

Predicted	Actual				
	1	2	3	4	5
1	316	37	0	0	0
2	20	118	17	0	0
3	0	22	76	12	0
4	0	1	24	62	11
5	0	0	0	24	140

TABLE VIII: Confusion Matrix for Ensembled SVM

Predicted	Actual				
	1	2	3	4	5
1	335	18	0	0	0
2	31	113	11	0	0
3	0	19	82	7	2
4	0	1	21	60	16
5	0	0	0	14	150

IV. Conclusion

In this research, we built an ensemble SVM classifier and compared it with a simple SVM for crop residue classification. This study represented a gain with respect to our previous approach not only in terms of the number of classes used for training (from 3 to 5 classes), but also in terms of the accuracy achieved (more than 80% in testing). In terms of the actual use of an simple SVM versus an ensemble SVM, the results were also satisfactory and justified the use of the ensemble. Indeed, the ensemble method increased all four metrics, while reducing the confusion between classes. In the future, we will increase the number of classes for an even finer crop residue classification. Also, we will test the generalization power of the ensemble by applying the models trained using the datasets from 2018, to the data from 2019 – a problem that we observed in our previous studies, which, again, we expect to be minimized by the use of the ensemble.

Acknowledgment

We appreciate the work of research specialists David Kleinsorge, Krystal Burkett-Tysdale and Theresa Musket; and student workers Ahmed Krgo, Isaac Shee, Robert Phillips, and Dewi Kharismawati. We also thank Missouri USDA NRCS for financial and logistical support with particular appreciation to Ron Miller and Glenn Davis.

References

[1] R Lal. Soils and sustainable agriculture. a review. agron sustain dev 28: 57–64, 2008.

[2] Lalaina Ranaivoson, Krishna Naudin, Aude Ripoche, Francois Affholder, Lilia Rabeharisoa, and Marc Corbeels. Agro-ecological functions of crop residues under conservation agriculture. a review. *Agronomy for Sustainable Development*, 37(4):1–17, 2017.

[3] Shan Huang, Yongjun Zeng, Jianfu Wu, Qinghua Shi, and Xiaohua Pan. Effect of crop residue retention on rice yield in china: A meta-analysis. *Field Crops Research*, 154:188–194, 2013.

[4] Lewrene K Glaser. *Provisions of the Food Security Act of 1985*. Number 498. US Department of Agriculture, 1986.

[5] United States Department of Agriculture. Subpart 503e crop residue. natural resources conservation service, 2011.

[6] JM Laflen, M Amemiya, and EA Hintz. Measuring crop residue cover. *Journal of soil and Water Conservation*, 36(6):341–343, 1981.

[7] Elbert C Dickey, David P Shelton, GE Meyer, and KT Fairbanks. Determining crop residue cover with electronic image analysis. 1989.

[8] John E Morrison, Chi-Hua Huang, David T Lightle, and Craig ST Daughtry. Residue measurement techniques. *Journal of soil and water conservation*, 48(6):478–483, 1993.

[9] A Laamrani, P Joosse, and N Feisthauer. Determining the number of measurements required to estimate crop residue cover by different methods. *Journal of Soil and Water Conservation*, 72(5):471–479, 2017.

[10] JA Lory, P Upadhyay, TAP Lagaunne, C Spinka, R Miller, G Davis, and GN DeSouza. Capability of high-resolution rgb imagery to accurately document residue in row-crop fields. *Journal of Soil and Water Conservation*, 2021.

[11] Miguel Quemada and Craig ST Daughtry. Spectral indices to improve crop residue cover estimation under varying moisture conditions. *Remote Sensing*, 8(8):660, 2016.

[12] Jibo Yue and Qingjiu Tian. Estimating fractional cover of crop, crop residue, and soil in cropland using broadband remote sensing data and machine learning. *International Journal of Applied Earth Observation and Geoinformation*, 89:102089, 2020.

[13] Payam Najafi, Bakhtiar Feizizadeh, and Hossein Navid. A comparative approach of fuzzy object based image analysis and machine learning techniques which are applied to crop residue cover mapping by using sentinel-2 satellite and uav imagery. *Remote Sensing*, 13(5):937, 2021.

[14] Esmael Hamuda, Martin Glavin, and Edward Jones. A survey of image processing techniques for plant extraction and segmentation in the field. *Computers and Electronics in Agriculture*, 125:184–199, 2016.

[15] Peter Riegler-Nurscher, Johann Prankl, Thomas Bauer, Peter Strauss, and Heinrich Prankl. A machine learning approach for pixel wise classification of residue and vegetation cover under field conditions. *biosystems engineering*, 169:188–198, 2018.

[16] PC Upadhyay, JA Lory, GN DeSouza, TAP Lagaunne, and C Spinka. Classification of crop residue cover in high resolution rgb images using machine learning. submitted for publication.

[17] José Miguel Guerrero, Gonzalo Pajares, Martín Montalvo, Juan Romeo, and María Guijarro. Support vector machines for crop/weeds identification in maize fields. *Expert Systems with Applications*, 39(12):11149–11155, 2012.

[18] Xiuliang Jin, Shouyang Liu, Frédéric Baret, Matthieu Hemerlé, and Alexis Comar. Estimates of plant density of wheat crops at emergence from very low altitude uav imagery. *Remote Sensing of Environment*, 198:105–114, 2017.

[19] Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, (6):610–621, 1973.

[20] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[21] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

[22] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[23] S Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

[24] Hsuan-Tien Lin and Chih-Jen Lin. A study on sigmoid kernels for svm and the training of non-psd kernels by smo-type methods. *submitted to Neural Computation*, 3(1-32):16, 2003.

[25] Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 44(8):1761–1776, 2011.

[26] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[27] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[28] Matthias Feurer and Frank Hutter. Hyperparameter optimization. In *Automated machine learning*, pages 3–33. Springer, Cham, 2019.

[29] Gang Wang, Jinxing Hao, Jian Ma, and Hongbing Jiang. A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications*, 38(1):223–230, 2011.

[30] Xin Huang and Liangpei Zhang. An svm ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. *IEEE transactions on geoscience and remote sensing*, 51(1):257–272, 2012.

[31] Rui Wang, Wei Li, Rui Li, and Liang Zhang. Automatic blur type classification via ensemble svm. *Signal processing: image communication*, 71:24–35, 2019.

[32] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.