A Deep Ensemble Classifier for Surface Defect Detection in Aircraft Visual Inspection

Ivan Ren[1,*], Feraidoon Zahiri[2], Gregory Sutton[2], Thomas Kurfess[1], and Christopher Saldana[1]
[1]The George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, Georgia, 30332, USA
[2]The Warner Robins Air Logistics Complex, Robins Air Force Base, Warner Robins, Georgia, 31088, USA
*Corresponding Author – Email: iren3@gatech.edu, Address: Manufacturing Related Disciplines Complex, Georgia Institute of Technology, 801 Ferst Drive, Atlanta, Georgia, 30332, USA

A Deep Ensemble Classifier for Surface Defect Detection in Aircraft Visual Inspection

ABSTRACT

Visual inspection is critical in many maintenance, repair, and overhaul operations and is often the primary defense against premature failure caused by unresolved surface defects. Traditionally, visual inspection is conducted by human operators in a time-consuming and subjective process. Recent advancements in deep learning have the potential to provide accurate detection of defects, leading to reduced inspection times. Several methodologies have been developed using convolutional neural networks (CNNs) to classify surface defects; however, these methods often rely on singular models to make detections. This makes them susceptible to inherent biases and variances introduced during the training process. Ensembling is a technique used to minimize the errors of CNNs through combining the outputs of multiple models. This paper presents an automated inspection methodology utilizing stacked ensembles of CNNs to classify defects on images of aircraft surfaces. The proposed framework is evaluated with images obtained from a borescope inspection of aircraft propeller blade bores. It is shown that the ensemble method improves inspection accuracy over conventional single-model deep learning methods. Furthermore, the error reduction provided by the ensemble method reduces false alarms at decision boundaries that minimize missed detections. The proposed method is shown to improve the reliability of automated detection systems, which can avoid catastrophic scenarios on critical systems such as aircraft propellers.

Keywords

automated inspection, ensemble classification, manufacturing surface defects, deep learning

Introduction

Visual inspections are ubiquitous within maintenance, repair, and overhaul (MRO) operations. In particular, they comprise over 80% of total inspections for large transport aircraft and are often the fastest and most economical method of detecting surface defects before they reach dangerous sizes[1]. Traditionally, visual inspections are conducted by human operators that scan the surface of the aircraft for indications of cracks, corrosion, disbonding, and incidental damage. Operators are equipped with visual aids such as mirrors and borescopes and recent advancements have introduced automated imaging technologies that allow for remote inspections. While these technologies have increased the amount and quality of available data, humans are still the primary decision makers on the acceptability of a part or surface. Human visual inspection is costly, time-consuming, and subject to human errors that can be exacerbated by mental fatigue or boredom. This has inspired research into the development of autonomous inspection systems to provide accurate and objective defect detection for improved inspection efficiency and reduced maintenance downtimes.

Prior literature on automated inspection systems with computer vision can be broadly categorized into heuristic-based methods[2-5] and machine learning (ML) methods[6-8,12-17]. Heuristic-based methods seek to identify and extract textural and color features from images using rudimentary algorithms such as edge detection, histogram thresholding, and wavelet transforms[9]. Predefined discriminant functions are used to classify various defects from the extracted features. ML-based methods consist of various shallow learning algorithms trained on handcrafted image features to obtain a classification function. The performance of both methods is highly reliant on the quality of the input features and saturates as the amount of data increases[10]. They are difficult to apply to a wide range of defects as feature extraction becomes increasingly cumbersome as the number of defect classes increase. Furthermore, it is difficult to represent the complex defects and backgrounds found in many aircraft inspections with simple features.

Deep learning (DL) has received substantial interest in recent years in applications of automated visual inspection. DL algorithms contain multiple trainable layers of feature representation that suppress irrelevant variations in background and lighting[11]. Unlike conventional ML algorithms, DL eliminates the need for human feature engineering by combining feature extraction and classification within the model structure. Convolutional neural networks (CNNs) are considered the current state-of-the-art for many image classification tasks and have shown promising results in defect detection for aircraft visual inspections[12-14]. Such inspections typically encounter uncertain imaging conditions and large variability in defect and surface morphology that limits the effectiveness of heuristic and ML-based methods. Prior CNN-based methods have focused on frameworks consisting of a single CNN model; however, it is known that ensembles of classifiers can improve classification performance. Ensembles are designed to increase accuracy by combining the predictions of multiple classifiers through a higher-level function. Ensemble-based methodologies have previously been developed for PCB assembly[15], pipeline health monitoring[16], and fatigue crack detection[17]. Ensembles of CNNs have been used in wagon component inspection[18] and car body inspection[19].

This paper proposes a computer vision inspection method utilizing ensembles of CNNs for the automated classification of aircraft surface defects. The CNNs are pre-trained on a general image set and fine-tuned for the inspection task to reduce training data requirements. The proposed method shows improved detection accuracy over existing single-model CNN frameworks. The component CNNs vary in structure and training data as increasing network diversity has been shown to improve ensemble performance[20]. Three variations of stacked ensembles are constructed, comprising of combinations of homogenous models, dimensionally-diverse models, and structurally-diverse models. The decision boundaries of the ensembles are adjusted to minimize missed detections for critical inspections where the cost of missed detections is high. The proposed method is validated using image data collected from a borescope inspection of aircraft propeller blades. The problem is approached as an image classification task due to the low dataset though a similar ensemble approach may be applied to object detection or segmentation networks for aircraft inspection.

The remainder of this paper is organized as follows: “Background” reviews the related works in computer vision for defect detection and provides a brief introduction to ensembling techniques. “Methods” describes the ensemble methodology used to detect surface defects. This includes a description of the training and evaluation process, selection of hyperparameters, and dataset generation. The experimental results and analysis are presented in “Results and Discussion”. This section provides a comparison of the performance of ensembles and individual CNNs. Finally, conclusions and recommendations for future work are provided in “Conclusions”.

Background

RELATED WORKS

Computer vision methods have been used in automated inspection for many years. Lee et al.[2] utilized digital color analysis to develop linear multi-feature discriminant functions to detect the presence of rust on steel bridges. Features were extracted through statistical analysis of the red, green, and blue (RGB) color channels. Wang and Cheng[3] employed Hough's circular transform to locate and characterize pitting corrosion in microscopic images. The method achieved greater than 95% accuracy in pit detection with less than 10% error in its estimates of pit radius and position. It was also shown to be capable of differentiating pits from other surface defects such as scratches and inclusions on simulated data. Guo et al.[4] combined morphological filters and Fisher's discriminant analysis on gray level pixel gradients to segment highlight scratches on steel surfaces. Gunatilake et al.[5] extracted gray level features from images of corroded aircraft skins with discrete wavelet transform to train a 1-nearest neighbor clustering algorithm to segment corroded and non-corroded surfaces. The algorithm was able to detect 95% of the corrosion samples in the testing set. Hoang and Tran[6] utilized a sliding window with a support vector machine (SVM) classifier to localize instances of corrosion within water pipelines. Features were extracted using histogram analysis and gray-level co-occurrence matrices (GLCMs). While accurate, the above techniques are often sensitive to varied illumination conditions and the presence of complex backgrounds. Moreover, shallow machine learning techniques are limited by the complexity of the handcrafted features and most lack generality and reliability beyond specific, targeted applications.

With recent advancements in graphics processing units (GPUs) and the success of CNNs in image classification challenges[21], there has been increased interest in CNNs for automated defect detection. Nonetheless, the lack of expansive datasets continues to hinder the development of these systems as most deep learning models require a training set of at least 1000 images per defect class to obtain acceptable performance. As defect rates are low for aircraft inspections, dataset generation is time-consuming and can be prohibitive. Prior works have implemented transfer learning[22,23] and data augmentation[24] to mitigating the issues associated with limited data. Malekzadeh et al.[12] used two pre-trained CNNs as feature extractors for an SVM classifier to inspect aircraft fuselages. Experimental results showed a detection accuracy of 96.38% and sensitivity of 96.48% on a testing set of 12 images. Shen et al.[13] trained a fully convolutional network (FCN) to identify and localize engine cracks and burns from borescope images with high accuracy. Ramalingham et al.[14] developed a mobile robot inspection platform equipped with a lightweight CNN network to differentiate between stains and defects on aircraft skins. The platform achieved a test accuracy of 96.2% with an average prediction confidence of 97%. These prior studies have demonstrated the excellent performance of CNNs in aircraft inspection; however, all of these methods utilize single CNN frameworks which are sensitive to poor model selection and improper training. An ensemble framework is a possible solution for these issues.

ENSEMBLES

Ensembles combine the predictions of multiple independent machine or deep learning models using a higher-level algorithm. The models which comprise an ensemble are referred to as base learners and are selected to maximize accuracy and diversity. A common strategy for designing ensembles, *overproduce and choose*, introduced by Giacinto and Roli[25] involves training a large set of base learners and selecting the most error-diverse models. The main methods to obtain sufficient diversity include varying the initial weights, varying the network architecture, varying the network type, and varying the training data. Diverse models tend to make errors on different subsets of the problem domain and the combination of their predictions is effective in reducing the overall error and variability in the system[26]. This protects against the selection of a base learner with sub-optimal generalization performance.

There are two common approaches to deep learning ensembles, bagging[27] and stacking[28]. In bagging, the predictions of the base learners are combined using an average or majority vote function. In stacking, a second-level machine learning algorithm, known as a meta-learner, is trained on the predictions of the component models to output the final prediction. Unlike bagging, stacking is resistant to differences in base learner accuracy. Ensembles can be extended to multiple levels to further improve performance but are often limited in size by computational restraints.

Several ensemble methodologies have been proposed for automated visual inspection. Wu, Liu, and He[16] compared the performance of four machine learning ensembles in the automated inspection of sewer pipes. The highest classification accuracy was obtained using a RotBoost ensembling method for decision trees. Dworakowski et al.[17] proposed an ensemble of artificial neural networks (ANNs) to detect fatigue cracks on aircraft from the readings of embedded piezoelectric sensors. The component ANNs were selected for their structural diversity and it was shown that the ensembles had greater mean classification accuracy than any of the individual ANNs. Fernandes et al.[18] trained a bagged ensemble of homogenous CNNs for railway wagon pad inspection using a small, unbalanced dataset of 334 grayscale images. The method showed low precision on the majority no-defect class relative to its performance on the smaller defect class. Chang et al.[19] combined the region proposals of double YOLOv3 object detection networks to identify defects on manufactured car bodies. The ensemble was shown to outperform both the single model YOLOv3 network and the trained human inspectors. A paired imaging system was developed to minimize illumination variation and enhance defect saliency.

Methods

This study proposes using a stacked ensemble of CNNs to classify defect and defect-free aircraft surfaces for MRO operations. To this end, five CNN architectures are trained on a dataset comprised of borescope images of propeller bore surfaces. Error-diverse base learners are selected to form three different stacked ensemble combinations. A logistic regression algorithm is selected as the meta-learner. The architecture of the proposed stacked ensembles is shown in figure 1. The ensembles are evaluated on a common testing set and a comparison of the classification results between the ensembles and base learners are presented in “Results and Discussion”. Figure 2 illustrates the overall framework of the proposed method.

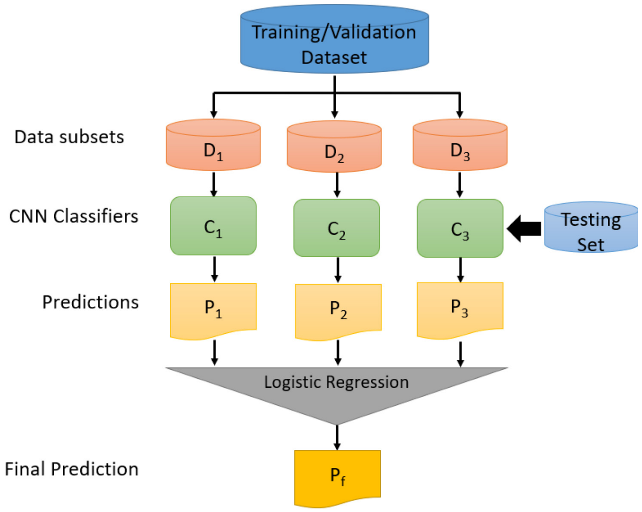

Fig. 1. Structure of the proposed stacked ensembles

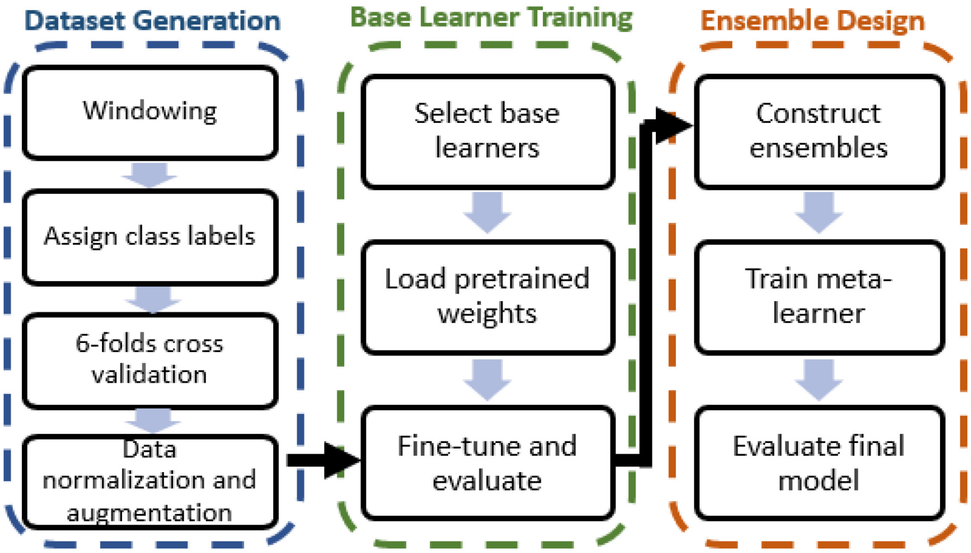

Fig. 2. Overall architecture of the proposed automated inspection framework

DATASET GENERATION

To create the dataset, images are gathered from a borescope inspection of aircraft propeller bores. A central rectangular region of 672 x 448 pixels is extracted from each original image and further divided into six 224 x 224 images. The original images are distorted and poorly illuminated at the extremities of the field of view due to a manually controlled light source and fisheye lens, hence the initial rectangular center crop. The split images of 224 x 224 pixels are manually annotated by a subject matter expert. Defects are defined as incongruities in the image surface. Cracks are usually long and thin whereas corrosion, which comprises the majority of the defect samples, is characterized by regions of lower pixel intensity with clear boundaries. Other defects include scratches and gouges. The annotated dataset is balanced to limit bias towards either class and consists of 300 defect and 300 defect-free. Given the limited number of defect samples, the defect class was not further differentiated into different defect types. Examples of defect and defect-free surfaces included in the dataset can be found in figure 3. To combat overfitting, K-fold cross validation is applied to divide the dataset into discrete subsets. A test set of 100 images is randomly selected from the full dataset and withheld from training. 5-folds cross validation is applied to the remaining 500 images to form the training and validation sets. The ratio of training, validation, and testing sets is 4:1:1. Five combinations of training and validation sets are created. The models are tuned for the validation set and then evaluated on the common testing set. This allows for a direct comparison of model performance.

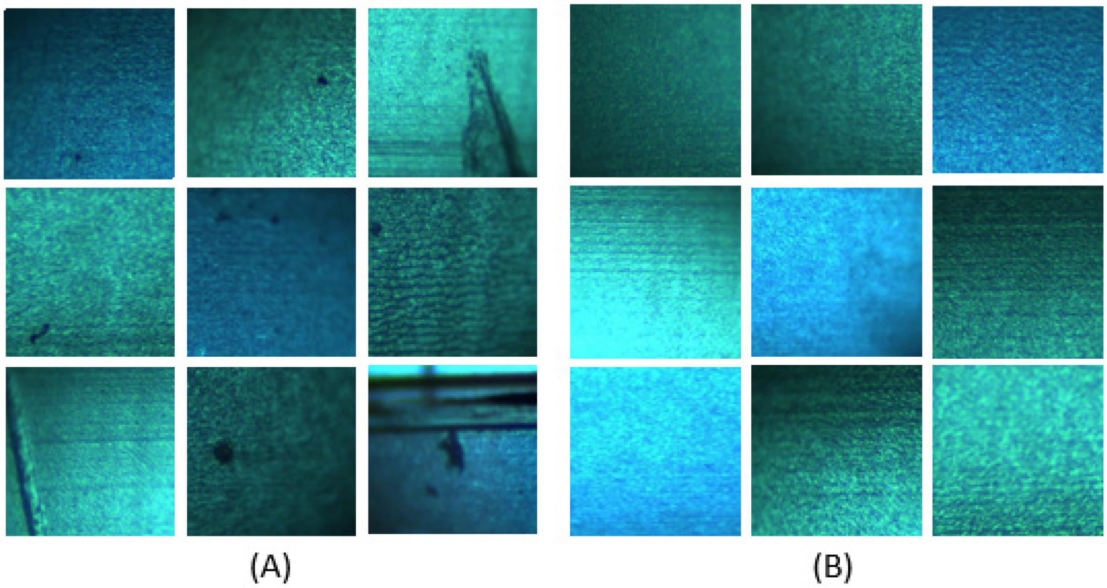

Fig. 3. Sample images: (A) defect images and (B) defect-free images

BASE LEARNERS

Several CNN architectures previously developed for the ImageNet challenge[21] are selected as the base learners. The Visual Geometry Group developed the VGG[29] architecture in 2014 and showed that network depth is critical in improving classification performance with CNNs. GoogLeNet[30] was designed with small receptive kernels, decreasing the number of network parameters per layer which allowed for deeper networks and improved computational efficiency. ResNet[31] was designed to combat the vanishing gradient issue of large, deep networks and achieved better than human classification performance on the ImageNet challenge. ResNet models with 18, 34, and 50 layers are used in this study. Varying the structure and training data of the selected CNNs is used to induce error diversity. Batch normalization is applied to the VGG network to improve speed and performance through regularization[32].

The base learners are pre-trained on the ImageNet dataset before being fine-tuned for the borescope detection task. To increase the size and diversity of the training set, data augmentation is implemented by randomly flipping the training images horizontally. This allows the CNN to learn more invariant features and helps prevent overfitting. Image rotations, brightness and contrast adjustment, and the addition of Gaussian noise were also tested but found to result in no improvement in model performance. To update the model weights, a cross entropy loss function is used to define the error between the model output and the target label during training. The weights are updated in the direction of decreasing error through a stochastic gradient descent (SGD) optimization algorithm with momentum. Momentum accelerates the SGD function towards convergence, allowing it to escape local minima or saddle points[33,34]. Recently, adaptive optimization algorithms such as Adam have become popular in deep learning but SGD with momentum has been shown in a recent study to outperform adaptive methods on a binary classification task with the same amount of hyperparameter tuning[35].

Several hyperparameters are defined in the training of the base learners. These include learning rate, batch size, and number of epochs. The learning rate determines the step size of the weight update as defined by the SGD function. A small learning rate converges slowly whereas a large learning rate may result in fluctuations of the optimizer at the minimum[36]. The initial learning rate is set to 0.001 and a scheduler decays the learning rate by a factor of 10 every seven epochs. This allows for faster convergence early in the training while preventing overshooting in the later iterations. Batch size refers to the number of training images per iteration. Batch size is set to 25 for the ResNet and GoogLeNet networks. VGG11_bn is trained with a batch size of 5 due to lack of memory on the training device. The number of epochs refers to the number of full passes through the training set. As training for too many epochs may lead to overfitting, the networks were trained for 50 epochs to determine an early stopping point where the validation accuracy has not changed for several epochs. From this analysis, the number of epochs was set to 25. Each of the base learners is trained twice on the five combinations of training and validation sets for a total of 50 models. During training, the hyperparameters are optimized to maximize performance on the validation set. This increases the risk of overfitting, a training error that occurs when the trained model is too closely fit to the training data and is consequently unable to make accurate classifications on the testing data. The inclusion of both a validation and testing set allows for easy detection of overfitting through comparison of the model performance on both sets. The addition of data augmentation and early stopping of training also help prevent overfitting. For the duration of this paper, networks will be used to denote the different CNN architectures whereas models will be used to describe unique trained iterations of the networks

ENSEMBLE DESIGN

Stacked ensembles consisting of three CNN base learners and a logistic regression meta-learner are constructed from the set of trained base learners. The logistic regression algorithm is a small ML algorithm suitable for binary classification problems and allows for easy fine-tuning of the decision boundary. Moreover, given the small data size, the algorithm is less prone to overfitting than other techniques such as SVMs. The logistic regression is trained on the outputs of the base learners to update the regression coefficients as defined by:

$$p = \frac{1}{1 + \exp^{\left(\frac{1}{1+\exp^{(b_0+b_1X_1+b_2X_2+b_3X_3)}}\right)}} \tag{1}$$

where p is the class probability, X_n are the distinct independent inputs, and b_k are the regression coefficients. To evaluate the impact of network diversity on ensemble performance, three variations of stacked ensembles are constructed as shown in Table 1. Ensemble A is homogeneous whereas Ensembles B and C vary in depth and structure, respectively. When constructing the ensembles, the base learners are selected to maximize accuracy and error diversity.

Table 1. Component base learners of the three stacked ensembles

Ensemble	Base Learners
Ensemble A	3x ResNet18
Ensemble B	ResNet18, Resnet34, Resnet50
Ensemble C	ResNet18, GoogLeNet, VGG11_bn

TOOLS

PyTorch[37] is an open-source deep learning framework developed by Facebook and used to program the networks. The logistic regression algorithm was trained using the scikit-learn[38] machine learning library. Training and evaluation are conducted on a laptop computer with a GeForce GTX 1050Ti GPU, Intel i7-8750H CPU, and 8GB of RAM.

Results and Discussion

Standard performance metrics such as accuracy, recall, precision, and F1 score are used to evaluate the developed models. Precision measures the proportion of positive detections that contained defects while recall describes the proportion of defects that were detected. F1 score is the harmonic mean of precision and recall and reflects the tradeoff between the two component metrics. Defect images are considered the positive class while defect-free images form the negative class. The equations for precision, recall, and F1 score are defined in equations (2)-(4) respectively. *TP*, *FP*, and *FN* represent the total number of true positives, false positives, and false negatives respectively. For most aircraft inspections, false positives are preferred as false negatives may lead to critical failures.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

$$recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = 2\left(\frac{precision \times recall}{precision + recall}\right) \tag{4}$$

BASE LEARNER PERFORMANCE

Table 2 shows the detection results of the base learners. The metrics presented are averaged over 10 runs. The Shapiro-Wilks test with a significance level of 0.05 determined that the data is non-normal, therefore, the non-parametric Mann-Whitney *U*-test is used to compare the mean performances of the base learners. The *U*-test assumes that the data is independent and non-normal. A sample *U*-test with an α-level of 0.05 comparing ResNet18 to the other base learners is shown in Table 3.

Table 2. Comparison of performance of different single-model CNN architectures on test images

Network	Accuracy (%)		Precision (%)		Recall (%)		F1 Score (%)	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
ResNet18	98.10	0.99	98.20	1.48	98.03	1.60	98.10	0.99
ResNet34	96.80	1.40	97.40	1.90	96.26	1.70	96.82	1.40
ResNet50	97.40	0.97	97.80	1.75	97.06	1.62	97.41	0.97
GoogLeNet	93.60	1.96	96.40	1.58	91.43	3.38	93.81	1.85
VGG11_bn	95.00	2.36	98.40	1.35	91.52	4.00	95.25	2.14

Table 3. Mann-Whitney *U*-test comparing ResNet18 to the other base learners

H_0: Group 1 ≤ Group 2, H_1: Group 1 > Group 2
$\alpha = 0.05$, N = 10 for both groups

Group 1	Group 2	Accuracy		F1 Score	
		Z	*p*	Z	*p*
ResNet18	ResNet34	2.103	0.018	1.797	0.036
ResNet18	ResNet50	1.349	0.088	1.033	0.151
ResNet18	GoogLeNet	3.770	8.15E-5	3.753	8.73E-5
ResNet18	VGG11_bn	2.926	0.002	2.466	0.007

The results in Table 3 show that ResNet18 has a greater mean, accuracy, recall, and F1 score than all of the networks expect ResNet50. The *U*-test did not find any significant difference between the mean results of ResNet18 and ResNet50. VGG11_bn has the greatest mean precision but its accuracy suffers due to its low recall rate. The large standard deviation in the recalls of GoogLeNet and VGG11_bn suggests overfitting due to the unstable performance commonly exhibited by overfit models. However, this is unlikely as the networks simultaneously show consistent high precision and overfitting would result as a performance decline in both precision and recall. Rather, the imbalance between Type I and Type II error indicates that the decision boundary that determines the final class prediction is misaligned. The output of the networks contains the likelihoods that the input images belongs to either the defect or defect-free class. For binary classification, the decision boundary is typically set at 50% and the majority class becomes the final prediction. When the Type I and Type II errors are unbalanced, such as with the GoogLeNet and VGG networks, lowering the decision boundary serves to balance the errors and decrease missed detections at the cost of increased false alarms.

A comparison of network performance on several commonly misclassified images is shown in figure 4. The values of the heatmap indicates the proportion of the ten models in each network that misclassified the corresponding image. The intra-network error diversity is a result of variations in training and validation data while error diversity between networks may be attributed to variations in both data and network structure. VGG11_bn has the added complexity of a different batch size, which influences the weight update of the SGD optimizer. Image 1 in figure 4 is a false positive and the remainder are false negatives. The defects are highlighted in red. The missed defects are either small with ambiguous edges, located in the extremities of the image, suffer from low contrast, or some combination of the previous listed factors. Small salient features tend to disappear within the filtering and downsampling in the convolutional and pooling layers of a CNN. While the networks showed overall high precision, a single defect-free image was misclassified by all of the networks. The false positive can be attributed to the rough surface finish as the majority of the other defect-free surfaces present in the dataset are smooth.

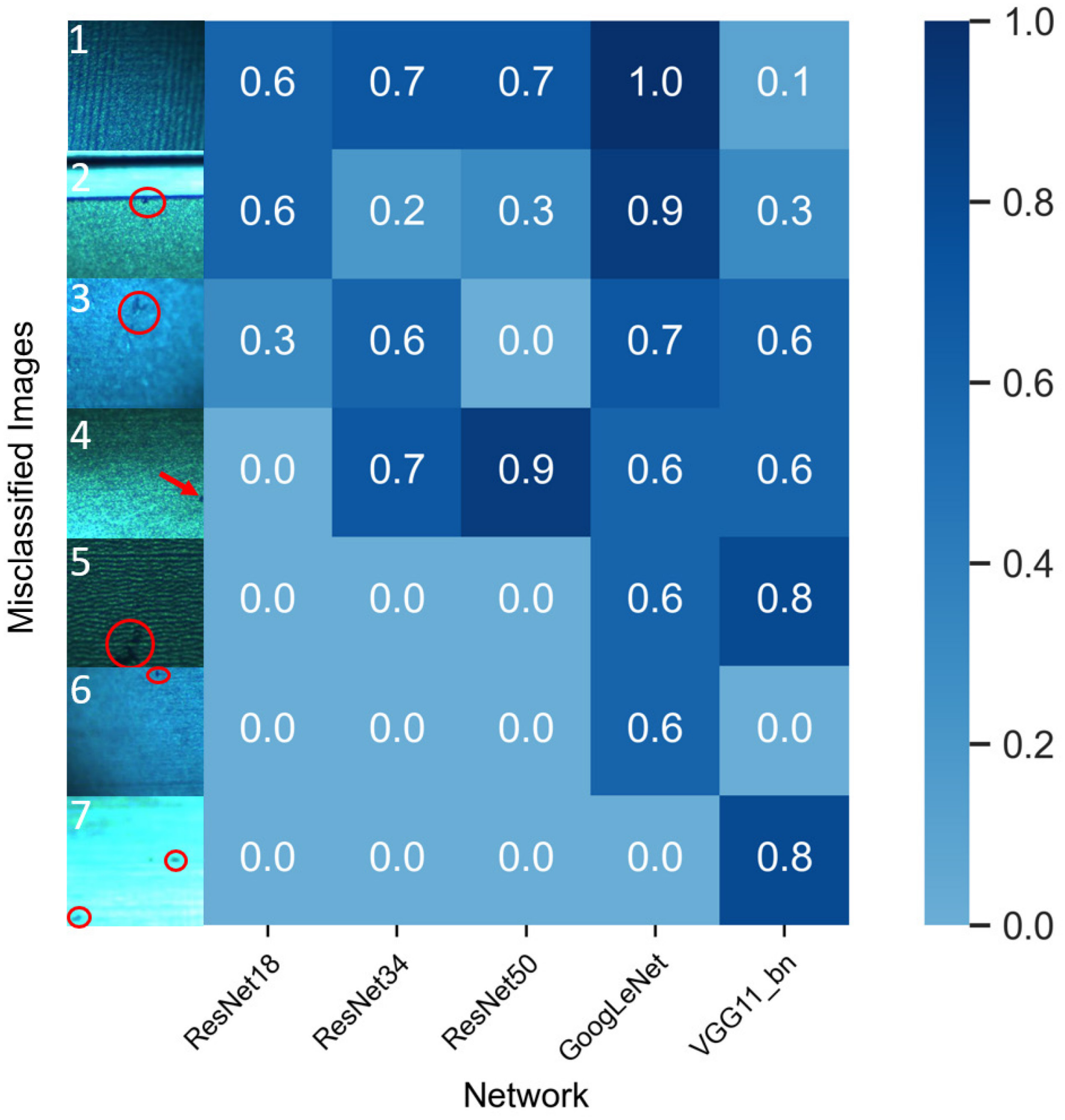

Fig. 4. Comparison of misclassifications by various networks. Values represent the proportion of models in each network that misclassified each image.

To further illustrate the impact of defect saliency, occlusion maps are generated for several defect images by sliding a gray 16 x 16 region across the image and mapping the correct class probability as a function of gray region position. The mapping was performed on a ResNet18 model and the features most relevant to the classification are shown in dark red. Defects with high contrast are clearly defined in the corresponding heatmap and the contributions of the surrounding surface to the class prediction are negligible. Occlusion maps of defects that are low contrast, small, or have poorly defined edges see greater influence from the surroundings which may lead to low prediction confidence and misclassifications.

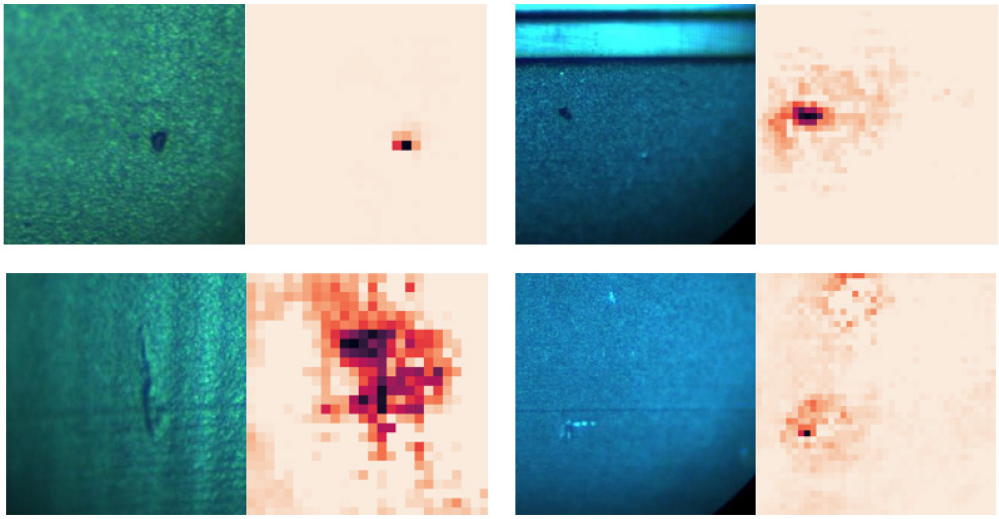

Fig 5. Occlusion maps of several defect images with a ResNet18 model.

COMPARISON OF RESULTS

The detection performance of the stacked ensembles can be seen in Table 4. The results are averaged over 10 runs. Again, a Shapiro-Wilks test was used to determine non-normality and the one-tailed Mann-Whitney *U*-test with a significance level of 0.05 is used to compare the mean performance of the ensembles. As the base learners of Ensemble C had lower mean accuracies than those of Ensembles A and B, it was expected that Ensemble C would be outperformed by the other two ensembles. However, the *U*-test showed that the mean performance metrics of the various ensembles are not significantly different. This is attributed to the accuracy of the ResNet18 network and the optimized regression coefficients of the logistic regression meta-learner. ResNet18 outperforms the VGG11_bn and GoogLeNet networks and is weighted more heavily by the logistic regression algorithm in the ensemble. This minimizes the errors of the

GoogLeNet and VGG11_bn models but also makes the ensemble more susceptible to training biases in ResNet18. To improve ensemble reliability, the base learners should be diverse in errors and similar in performance. Many of the constructed ensembles achieved perfect accuracy but given the limited dataset, this does not necessarily reflect operational performance. The results indicate that the proposed ensemble method achieves greater accuracy than the methods developed by Malekzadeh et al.[12] and Ramalingham et al.[14]; however, it is difficult to compare the performance of the various methods due to the differences in datasets and data domains.

The Wilcoxon signed-rank test is used to compare the performance of the base learners and ensembles. The test assumes non-normality and dependence. The test was performed with a α-level of 0.05 and the resulting *Z* scores and *p*-values for the accuracy and F1 score are found in Table 6. The Wilcoxon test shows that the stacked ensembles have greater mean accuracies, precisions, recalls, and F1 scores than their component base learners.

Table 4. Performance of different stacked ensembles on test images

Ensemble	Accuracy (%)		Precision (%)		Recall (%)		F1 Score (%)	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
Ensemble A	99.80	0.42	99.61	0.83	100.00	0.00	99.70	0.48
Ensemble B	99.50	0.527	99.22	1.01	99.80	0.63	99.5	0.52
Ensemble C	99.60	0.52	99.80	0.62	99.40	0.97	99.6	0.52

Table 5. Wilcoxon ranked-test results between base learners (Group 1) and ensembles (Group 2)

H_0: Group 1 ≥ Group 2, H_1: Group 1 < Group 2
α = 0.05, N = 10 for all Group 1 and 2

Group 1	Group 2	Accuracy		F1 Score (%)	
		Z	*p*	Z	*p*
ResNet18	Ensemble A	-2.641	0.004	-2.611	0.004
ResNet34		-2.768	0.003	-2.754	0.003
ResNet50		-2.796	0.002	-2.756	0.003
GoogLeNet		-2.768	0.003	-2.754	0.003
VGG11_bn		-2.771	0.003	-2.756	0.003
ResNet18	Ensemble B	-2.663	0.004	-2.611	0.004
ResNet34		-2.773	0.003	-2.756	0.003
ResNet50		-2.819	0.002	-2.756	0.003
GoogLeNet		-2.757	0.003	-2.754	0.003
VGG11_bn		-2.773	0.003	-2.754	0.003
ResNet18	Ensemble C	-2.401	0.008	-2.347	0.009
ResNet34		-2.627	0.004	-2.754	0.003
ResNet50		-2.627	0.004	-2.754	0.003
GoogLeNet		-2.762	0.002	-2.752	0.003
VGG11_bn		-2.763	0.002	-2.756	0.003

In aircraft visual inspection, it is critical to eliminate false negatives. With single CNN-based frameworks, this often entails either retraining with modified parameters and data or adjusting the decision boundary to eliminate false negatives at the cost of additional false positives. Retraining results cannot be guaranteed due to the stochastic nature of the algorithms and additional data is often unavailable. Ensembling can eliminate false negatives by selecting error-diverse learners. This can be done with little to no increase in the rate of false positives due to the overall error reduction provided by ensembles. An ensemble constructed to eliminate false positives is shown in figure 6.

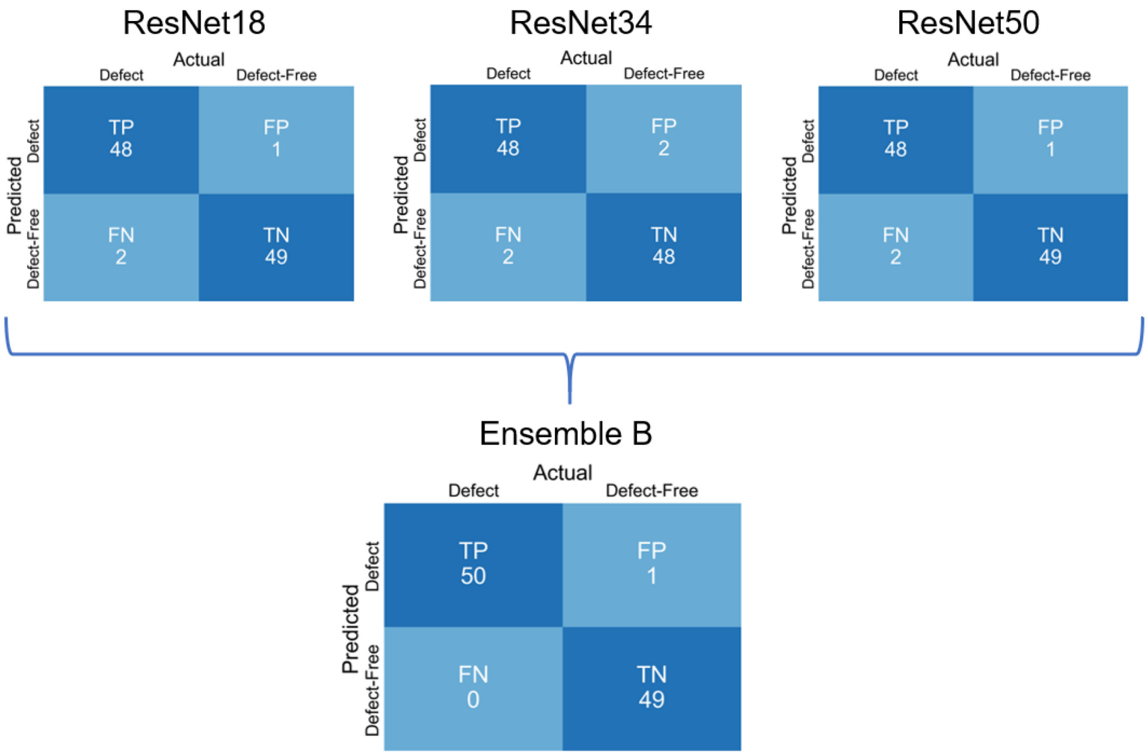

Fig. 6. Elimination of false negatives in ensemble through selection of error-diverse base learners

Decision boundary adjustment may also be applied to the meta-learner to further fine-tune performance. This offers two main advantage over boundary adjustment at the base learner level: 1) The error reduction provided by ensembles allows for a lower rate of false positives at high recall rates and vice versa. 2) Boundary adjustment at the meta-learner level is simpler and less cumbersome than adjusting the thresholds for each of the base learners. Precision-recall (PR) curves are plotted in figure 7 to illustrate the effects of boundary adjustment on mean precision and recall. The decision boundary is incremented from 0 to 100 % at intervals of 5 % and model performance is evaluated at each step. At a recall rate of 100 %, Ensembles A, B, and C attain precisions of 99.61, 96.91, and 82.69 % respectively. ResNet34 attains the highest precision for a base learner at 77.43 %. The area under the PR curve (AUC) is also commonly used to compare the overall performance of the classifier. A perfect classifier has an AUC of 1.0 and of the models in this study, Ensemble A has the greatest AUC of 0.99.

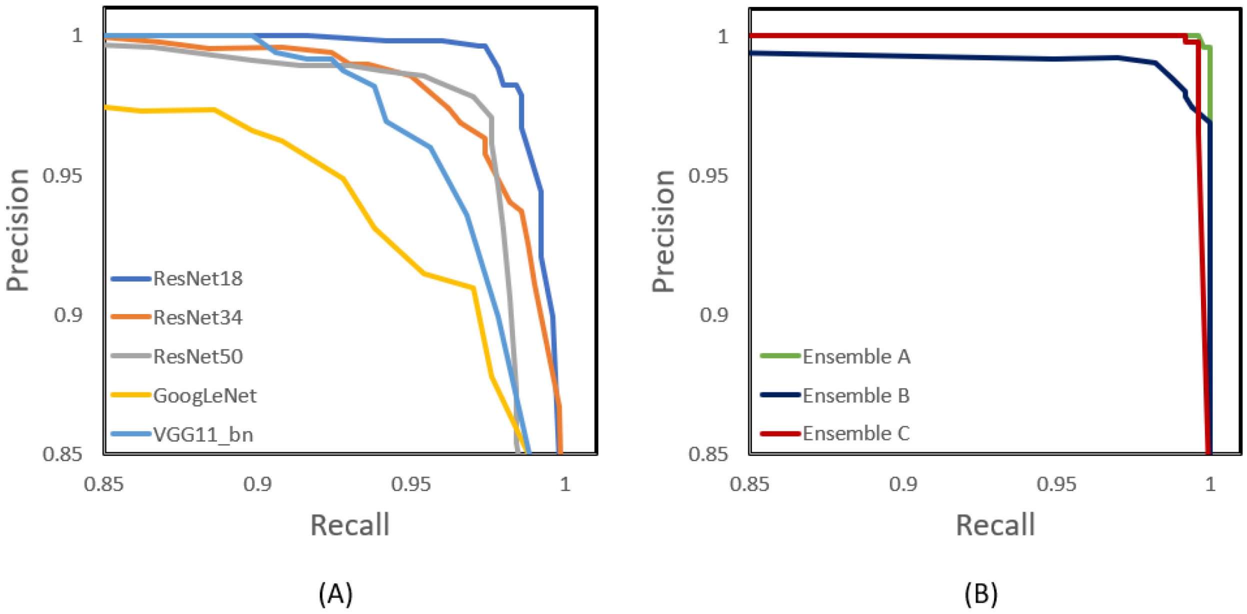

Fig. 7. Precision-recall curves of (A) base learners and (B) ensembles

COMPUTATIONAL PERFORMANCE COMPARISON

A comparison of the computational costs for each network is presented in Table 6. The networks were trained for 25 epochs with a batch size of 5 to obtain training times. The ensemble components were trained sequentially rather than in parallel. As expected, the computational resources scale with the complexity of the network. The ensembles require the most computation resources as they are a combination of the various base learners and the logistic regression meta-learner. The logistic regression algorithm contributes very little to the overall computational costs of the ensembles, adding 3 ms to the training time and 0.753 KB to the network size. The complexity of the ensembles can be reduced through the selection of lightweight base learners and pruning of network parameters to remove weights of low relevance.

Table 6. Comparison of computation costs of different networks. Training times are reported for 25 epochs and batch size of 5. Inference times are for a single 224 x 224 image.

Network	Network size (MB)	Training time (s)	Inference time (ms)
ResNet18	106	350	34
ResNet34	178	430	38
ResNet50	376	558	42
GoogLeNet	119	344	34
VGG11_bn	689	619	42
Ensemble A	318	1050	102
Ensemble B	660	1338	114
Ensemble C	915	1303	110

Conclusions

In this study, a stacked ensemble framework is proposed for automated detection of surface defect for aircraft visual inspection. Stacked ensembles comprising of three CNN base learners are trained to identify cracks, corrosion, scratches, and gouges in propeller bores using transfer learning and data augmentation to reduce dataset requirements. Using the Wilcoxon signed-rank test, the resulting ensembles are shown to have greater mean accuracies than the individual base learners. The ensembles also compare favorably against single model CNN-based detection methods in prior literature. There were no significant differences in performance between the three ensemble combinations despite diversity in the structure and depth of their component models. It is shown that error-diverse stacked ensembles can compensate for variability in base learner performance and protect against inherent biases or variances present within individual models, providing increased accuracy and reliability in detection tasks. Furthermore, it is shown that decision threshold adjustment at the meta-learner level is effective in maintaining precision for applications that require maximum recall. At decision boundaries corresponding to 100 %

recall, the ensembles achieved a mean precision 15.64 % higher than that of the highest precision base learner, ResNet34.

Several limitations exist for the proposed method. Ensembles are incapable of improving the generalization performance of its base learners, making them reliant on accurate, diverse component models for their performance. Sufficient base learners may be difficult to obtain in practice due to general limitations of CNN methods as well as the existence of correlated errors across models. Moreover, it is difficult to automate the selection of ensemble components and the process to develop ensembles for new inspection tasks is tedious. Ensembles are computationally expensive, limiting their performance for high speed applications.

In future work, the use of ensembles in this study can be expanded to detect multiple defect classes to evaluate the error reduction provided in multi-class applications. Furthermore, moving beyond classification, ensembles can be applied to improve the performance of detection algorithms that seek to localize objects of interest using bounding boxes. As object detection methods such as YOLO or SSD rely on CNNs as classifiers, an ensemble approach could see similar performance benefits. This would remove the inefficiency associated with windowing to localize and highlight the defect within the original image.

Acknowledgements

This work was supported in part by DE-EE0008303, NSF CMMI-1646013, NSF CMMI-1825640 and NSF IIP-1631803.

References

1. Federal Aviation Administration. “Visual Inspection for Aircraft” August 14, 1997.

2. S. Lee, L. Chang, and M. Skibniewski, “Automated Recognition of Surface Defects Using Digital Color Image Processing” *Automation in Construction* 15, no. 4 (July 2006): 540–49, https://doi.org/10.1016/j.autcon.2005.08.001

3. Y. Wang and G. Cheng, “Application of Gradient-Based Hough Transform to the Detection of Corrosion Pits in Optical Images” *Applied Surface Science* 366 (2016): 9–18, https://doi.org/10.1016/j.apsusc.2015.12.207

4. J. H. Guo, X. D. Meng, and M. D. Xiong, “Study on Defection Segmentation for Steel Surface Image Based on Image Edge Detection and Fisher Discriminant” *Journal of Physics: Conference Series* 48 (October 1, 2006): 364–68, https://doi.org/10.1088/1742-6596/48/1/068

5. P. Gunatilake, M. Siegel, A. G. Jordan, and G. W. Podnar, “Image Understanding Algorithms for Remote Visual Inspection of Aircraft Surfaces” (paper presentation, Proc. SPIE 3029, Machine Vision Applications in Industrial Inspection V, San Jose, CA, 1997).

6. N. Hoang and V. Tran, “Image Processing-Based Detection of Pipe Corrosion Using Texture Analysis and Metaheuristic-Optimized Machine Learning Approach” *Computational Intelligence and Neuroscience* 2019 (2019), https://doi.org/10.1155/2019/8097213

7. L. A. O. Martins, F. L. C. Pádua, and P. E. M. Almeida, “Automatic Detection of Surface Defects on Rolled Steel Using Computer Vision and Artificial Neural Networks” (paper presentation, IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society, 2010).

8. X. Zhiwei, C. Muhua, and G. Qingji, “The Structure and Defects Recognition Algorithm of an Aircraft Surface Defects Inspection Robot” (paper presentation, 2009 International Conference on Information and Automation, Macau, China, 2009), https://doi.org/10.1109/ICINFA.2009.5205019

9. X. Xie, “A Review of Recent Advances in Surface Defect Detection Using Texture Analysis Techniques” *Electronic Letters on Computer Vision and Image Analysis* 7, no. 3 (June 26, 2008): 1–22, https://doi.org/10.5565/rev/elcvia.268

10. P. R. Kumar and E. B. K. Manash, “Deep Learning: A Branch of Machine Learning.” *Journal of Physics: Conference Series* 1228 (May 2019), https://doi.org/10.1088/1742-6596/1228/1/012045

11. Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning” *Nature* 521, no. 7553 (May 2015): 436–44, https://doi.org/10.1038/nature14539

12. T. Malekzadeh, M. Abdollahzadeh, H. Nejati, and N. Cheung, “Aircraft Fuselage Defect Detection Using Deep Neural Networks” *ArXiv:1712.09213 [Cs]*, (December 26, 2017).

13. Z. Shen, X. Wan, F. Ye, X. Guan, and S. Liu, “Deep Learning Based Framework for Automatic Damage Detection in Aircraft Engine Borescope Inspection” (paper presentation, 2019 International Conference on Computing, Networking and Communications (ICNC), 2019), https://doi.org/10.1109/ICCNC.2019.8685593

14. B. Ramalingam, V. Manuel, M. R. Elara, A. Vengadesh, A. K. Lakshmanan, M. Ilyas, and T. J. Y. James, “Visual Inspection of the Aircraft Surface Using a Teleoperated Reconfigurable Climbing Robot and Enhanced Deep Learning Technique” *International Journal of Aerospace Engineering* 2019 (September 12, 2019): 1–14, https://doi.org/10.1155/2019/5137139

15. B. Luo, Y. Zhang, G. Yu, and X. Zhou, “ANN Ensembles Based Machine Vision Inspection for Solder Joints” (paper presentation, 2007 IEEE International Conference on Control and Automation, 2007), https://doi.org/10.1109/ICCA.2007.4376934

16. W. Wu, Z. Liu, and Y. He, “Classification of Defects with Ensemble Methods in the Automated Visual Inspection of Sewer Pipes” *Pattern Analysis and Applications* 18, no. 2 (May 2015): 263–76, https://doi.org/10.1007/s10044-013-0355-5

17. Z. Dworakowski, K. Dragan, and T. Stepinski, “Artificial Neural Network Ensembles for Fatigue Damage Detection in Aircraft” *Journal of Intelligent Material Systems and Structures* 28, no. 7 (April 2017): 851–61, https://doi.org/10.1177/1045389X16657428

18. E. Fernandes, R. Rocha, B. Ferreira, E. Carvalho, A. C. Siravenha, A. C. S. Gomes, S. Carvalho, and C. R. B. de Souza, “An Ensemble of Convolutional Neural Networks for Unbalanced Datasets: A Case Study with Wagon Component Inspection.” (paper presentation, 2018 International Joint Conference on Neural Networks (IJCNN), 2018), https://doi.org/10.1109/IJCNN.2018.8489423

19. F. Chang, M. Liu, M. Dong, and Y. Duan, “A Mobile Vision Inspection System for Tiny Defect Detection on Smooth Car-Body Surfaces Based on Deep Ensemble Learning.” *Measurement Science and Technology* 30, no. 12 (December 1, 2019), https://doi.org/10.1088/1361-6501/ab1467

20. Y. Liu, Y. Jin, and H. Ma, “Surface Defect Classification of Steels Based on Ensemble of Extreme Learning Machines” (paper presentation, 2019 WRC Symposium on Advanced Robotics and Automation, 2019), https://doi.org/10.1109/WRC-SARA.2019.8931807.

21. J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database" (paper presentation, 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009), https://doi.org/10.1109/CVPR.2009.5206848.

22. C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on Deep Transfer Learning" (paper presentation, The 27th International Conference on Artificial Neural Networks, 2018).

23. S. J. Pan and Q. Yang, "A Survey on Transfer Learning." *IEEE Transactions on Knowledge and Data Engineering* 22, no. 10 (October 2010): 1345–59, https://doi.org/10.1109/TKDE.2009.191.

24. L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification Using Deep Learning" *ArXiv:1712.04621 [Cs]* (December 13, 2017).

25. G. Giacinto and F. Roli, "An Approach to the Automatic Design of Multiple Classifier Systems." *Pattern Recognition Letters* 22 (2001): 25–33.

26. L. K. Hansen and P. Salamon, "Neural Network Ensembles" *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12, no. 10 (October 1990): 993–1001, https://doi.org/10.1109/34.58871.

27. L. Breiman, "Bagging Predictors" *Machine Learning* 24, no. 2 (August 1996): 123–40, https://doi.org/10.1007/BF00058655.

28. D. Wolpert, "Stacked Generalization" *Neural Networks* 5, no. 2 (July 29, 1991): 241–59, https://doi.org/10.1016/S0893-6080(05)80023-1.

29. Simonyan, Karen, and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition" (paper presentation, International Conference on Learning Representations, 2015).

30. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions" (paper presentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015).

31. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition." (paper presentation, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016).

32. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." *ArXiv:1502.03167 [Cs]*, (March 2, 2015), http://arxiv.org/abs/1502.03167.

33. B. T. Polyak, “Some Methods of Speeding up the Convergence of Iteration Methods” *USSR Computational Mathematics and Mathematical Physics* 4, no. 5 (January 1964): 1–17, https://doi.org/10.1016/0041-5553(64)90137-5.

34. I. Sutskever, J. Martens, and G. Dahl, “On the Importance of Initialization and Momentum in Deep Learning” (paper presentation, Proceedings of the 30th International Conference on Machine Learning, 2013).

35. A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and Benjamin Recht, “The Marginal Value of Adaptive Gradient Methods in Machine Learning.” (paper presentation, 31st Conference on Neural Information Processing Systems, Long Beach, CA, 2017).

36. S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A Survey of Optimization Methods From a Machine Learning Perspective" (paper presentation, IEEE Transactions on Cybernetics, November 18, 2019).

37. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic Differentiation in PyTorch” (paper presentation, 31st Conference on Neural Information Processing Systems, 2017).

38. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M Blondel, et al., “Scikit-Learn: Machine Learning in Python” *Journal of Machine Learning Research* 12 (2011): 2825–30.

Figure Captions

Fig. 1. Structure of the proposed stacked ensembles

Fig. 2. Overall architecture of the proposed automated inspection framework

Fig. 3. Sample images: (A) defect images and (B) defect-free images

Fig. 4. Comparison of misclassification by various networks. Values represent the proportion of models in each network that misclassified each image.

Fig 5. Occlusion maps of several defect images with a ResNet18 model.

Fig. 6. Elimination of false negatives in ensemble through selection of error-diverse base learners

Fig. 7. Precision-recall curves of (A) base learners and (B) ensembles

Table Captions

Table 1. Component base learners of the three stacked ensembles

Table 2. Comparison of performance of different single-model CNN architectures on test images

Table 3. Mann-Whitney U¬-test comparing ResNet18 to the other base learners

Table 4. Performance of different stacked ensembles on test images

Table 5. Wilcoxon ranked-test results between base learners (Group 1) and ensembles (Group 2)

Table 6. Comparison of computation costs of different networks. Training times are reported for 25 epochs and batch size of 5. Inference times are for a single 224 x 224 image