Surface Defect Detection and Evaluation for Marine Vessels using Multi-Stage Deep Learning

Li Yu*, Kareem Metwaly*, James Z. Wang, and Vishal Monga

Abstract—Detecting and evaluating surface coating defects is important for marine vessel maintenance. Currently the assessment is carried out manually by qualified inspectors using international standards and their own experience. Automating the processes is highly challenging because of the high level of variation in vessel type, paint surface, coatings, lighting condition, weather condition, paint colors, areas of the vessel, and time in service. We present a novel deep learning-based pipeline to detect and evaluate the percentage of corrosion, fouling, and delamination on the vessel surface from normal photographs. We propose a multi-stage image processing framework, including ship section segmentation, defect segmentation, and defect classification, to automatically recognize different types of defects and measure the coverage percentage on the ship surface. Experimental results demonstrate that our proposed pipeline can objectively perform a similar assessment as a qualified inspector.

Index Terms—Defect detection, marine vessel, deep learning, image segmentation, multi-label classification.

I. INTRODUCTION

Detecting and evaluating surface coating defects are important procedures in the maintenance and repair of marine vessels. Defects, such as corrosion and fouling, have to be blasted before repainting so that it maintains the hull integrity and assures the surface poses little resistance to the water. Visual inspection of defects is typically carried out at shipyards by trained human inspectors. This procedure, however, can be costly and subjective. The quality of defect assessment depends largely on the experience and attentiveness of the inspectors. An automatic defect detection system is thus highly desired to achieve comparable accuracy as humans but with higher efficiency and consistency.

The problem of automatic detection and evaluation of vessel surface defects, however, is highly challenging because of the high level of variation in vessel type, paint surface, coatings, lighting condition, weather condition, paint colors, areas of the vessel, and time in service. The same set of algorithms is expected to provide accurate results for all types of vessels, under any reasonable photographic imaging condition, for any exterior surface area photographed, and regardless of the types

and colors of the coating layers used on the vessel and time the existing coatings have been in service. It is unlikely one can develop a conventional algorithm that can have such a high level of robustness. Thus, we take a data-driven approach using deep learning.

In this work, we propose a multi-stage deep learning framework to automatically detect three types of defects–corrosion, delamination, and fouling–from images of marine vessels and estimate the percentage of defect coverage over different sections of the marine vessel. To train the networks, we also manually annotated hundreds of images and built a complete dataset for the analysis of different stages of the framework. To our knowledge, this is the first effort to build a real-world dataset of this scale and comprehensiveness and to develop a fully automatic defect detection and evaluation system that includes ship segmentation, ship section segmentation, defect detection, and defect classification.

A. Related Work

Over the past years, researchers have tried to incorporate computer vision techniques into the assessment of surface defects of marine vessels. Navarro *et al.* [1] introduced a sensor system that estimates the gray-scale histogram of background and determines defect pixels by a threshold. Aijazi *et al.* [2] explored the HSV color space to separate the illumination invariant color component from the intensity so that histogram-based distributions and thresholds can be more effective to help spot corrosion of different shapes and sizes. Jalalian *et al.* [3] focused on the hue channel, instead of the complete HSV, that they utilized the Gaussian mixture model to estimate the circular hue histograms and the probability distribution of local entropy values [4] to identify segments of defects. Texture features are also used in the detection of defects. In [5], wavelet transform was adopted to reconstruct the image representation in terms of the frequency of components. High-frequency components, which usually happen at the coarse appearance, can be separated from low-frequency components, which correspond to flat and fine appearance. Bonnín-Pascual *et al.* [6] combined color histograms and morphological properties to detect corrosion and cracks, in which they exploited a percolation model [7] with a region-growing procedure to identify dark, narrow, and elongated sets of connected pixels.

Though effective, these earlier methods fall short in the following three aspects. First, the classical computer vision techniques only consider shallow color or texture features. They are not robust to changes in size, shape, illumination, etc. The existence of large, continuous defects can hardly

be detected by these local-feature-based methods. Second, the reliance on handcrafted features makes the algorithms difficult to tune the parameters and learn from large amounts of labeled data. Deep neural networks, on the other hand, have many more parameters to be tuned than traditional methods, giving more capability and flexibility to capture more complex features. Third, previous work only focused on the detection of rust and corrosion, whereas other defects on the surface of marine vessels, such as delamination and fouling, are also important in vessel maintenance.

The rise of deep learning has yielded a new generation of image segmentation algorithms with remarkable performance improvements. FCN [8] is one of the first fully convolutional networks to generate segmentation maps for images. Chen *et al.* [9] extended the FCN by adding the conditional random field (CRF) to help localize segment boundaries and achieved higher accuracy than pure FCNs. SegNet [10] builds on an encoder-decoder architecture to include an encoder for feature extraction and a decoder with upsampling layers for pixel-wise classification. Inspired by FCNs and encoder-decoder models, Ronneberger *et al.* [11] proposed the well-known UNet for medical image segmentation, which has multiple variations such as VNet [12] and UNet++ [13]. Mask RCNN [14] is another class of segmentation models that uses a region proposal network (RPN) to propose bounding box candidates and extract the region of interest (RoI). It also has multiple extensions, including PANet [15], MaskLab [16], and CenterMask [17]. The DeepLab family [18], [19] approaches differently by adopting the dilated convolution [20] (a.k.a. "atrous" convolution) to address the decreasing resolution problem along with the network layers and the Atrous Spatial Pyramid Pooling (ASPP) to capture objects at multiple scales.

Deep learning has been successfully applied to the field of defect or anomaly detection, such as the defects in pantograph slides [21] and road damage [22]. For the detection of surface defects of marine vessels, we adopt the UNet as the base model because our task is more similar to the tasks in medical image segmentation where objects don't have fixed size, shape, or boundary. In typical computer vision applications, objects carry clear semantic meanings. Take the working datasets of DeepLab [18] as an example, the PASCAL VOC 2012 [23] and Cityscapes [24] contain mostly well-defined object classes such as car and person. On the other hand, the coating defects of marine vessels often have intertwined and ambiguous boundaries between different defect types or with the background.

B. Contributions

We summarize our contributions as follows.

- We constructed a first-of-its-kind complete dataset for defect detection and evaluation of the surface of marine vessels, from images. It contains 730 images with pixel-level defect annotations and 350 images with ship section annotations.
- We proposed a fully automatic evaluation system to detect defects and calculate defect coverage. The system consists of four deep learning modules: whole ship segmentation, ship section segmentation, defect segmentation, and defect classification.
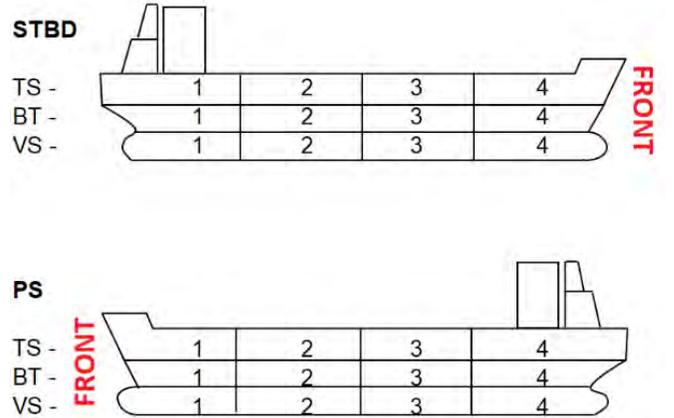
Fig. 1: Sections of a ship viewed from the starboard (STBD) and the port side (PS). The three levels we used are Top Side (TS), Boot Top (BT), and Vertical Side (VS). Diagrams courtesy of the PPG Industries, Inc.

The novel aspects of the individual modules are listed below.

- We proposed to segment three horizontal sections, namely Top Side (TS), Boot Top (BT), and Vertical Side (VS) (Figure 1), of a ship by predicting the two boundaries, which converts 2D segment prediction into 1D curve prediction. It reduced the requirement for labeled data and increased consistency in block predictions.
- We utilized a teacher-student training scheme in defect segmentation to overcome the shortage of labeled data and the problem of coarse or incomplete labels. The teacher model was first trained on manually labeled defect data and used to generate pseudo labels. The pseudo labels were then combined with the original labels to form more accurate and complete labels. The student model trained on the new labels outperformed the teacher model.
- We adopted spatial transformer networks (STN) [25] as feature extractors and proposed a multi-head architecture for multi-label prediction. STNs can effectively estimate affine transformations such that input image patches are warped for better feature extraction. A delamination detection head was added aside to the general head to cope with the need to detect the more challenging delamination defects.

The remainder of the paper is organized as follows. Section II describes the details of different defects and our collected datasets. Section III sketches the multi-stage deep learning pipeline and the subsequent Sections IV and V present the architectures of individual modules. Experiment results, including ablation studies and comparisons with baseline models, are reported in Section VI . Finally, we conclude and suggest future work in Section VII.

II. THE DATASET

We collaborated with the PPG Industries, Inc. (PPG) and collected a fully annotated real-world dataset. Figure 2 shows

some typical examples of corrosion, delamination, and fouling. They may occur in different sections of a marine vessel, as illustrated in Figure 3. Figure 3(a) displays the labels annotated by our collaborators from PPG, with different colors indicating different types of defects. To calculate the percentage of defect coverage over different sections of the ship, we need to identify sections of the ship from an image. Figure 3(b) shows the labels of three different sections. From top to bottom, they are TS, BT, and VS sections. We labeled around 730 images for defect detection and 350 images for section segmentation. Because the annotators have to manually classify defects and carefully trace the boundaries, the process demands expertise in marine coating and is time-consuming.

These photographs were collected by field inspectors over the years as they were doing on-site vessel inspections. Because the images were taken *not* for the purpose of computer-based analysis, there was no set standard on photo-taking. This poses a serious challenge to computer-based analyses because the developed algorithms must be robust enough to handle any reasonable in-the-wild photo-taking condition.

(a)

(b)

(c)
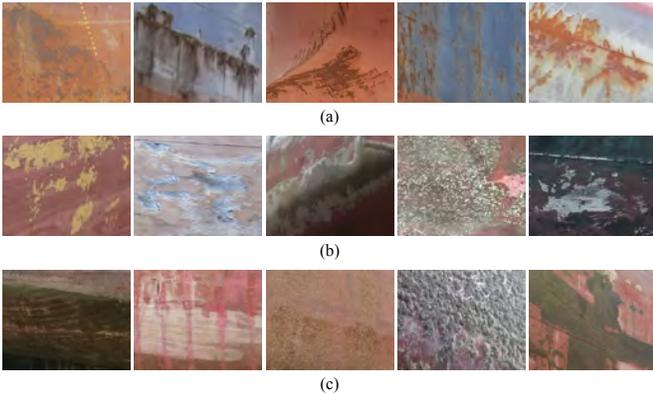
Fig. 2: Typical defects of (a) corrosion, (b) delamination, and (c) fouling.

III. MULTI-STAGE APPROACH OVERVIEW

The multi-stage approach for defect detection mainly consists of four stages, as shown in Figure 4. They are whole ship detection, section segmentation, defect segmentation, and classification. Stages 1 and 2 are required for extracting the region corresponding to the ship and subsequently different sections of the ship. Stages 3 and 4 perform defect segmentation and classification respectively. We contend that performing this as two-serial steps is superior to a single multi-class segmentation algorithm, because a region may contain multiple defect types. That is, an object in an image can have two labels, making typical multi-class segmentation models unsuitable.

IV. WHOLE SHIP AND SECTION SEGMENTATION

In order to evaluate the percentage of three defect types on different sections of the ship, we first isolate the ship within the image and segment different sections. To isolate the ship, we use an UNet [11] based image segmentation algorithm. The whole ship segmentation model is trained under

(a)

(b)
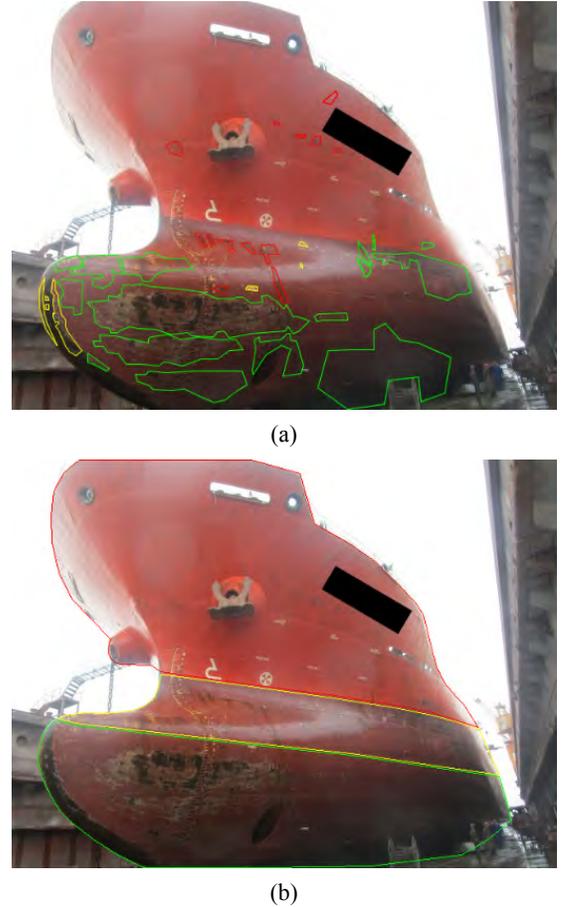
Fig. 3: Example defect labels and ship section labels: (a) corrosion (red), delamination (yellow), and fouling (green); (b) TS (red), BT (yellow), and VS (green). The ship name is blocked in black for preserving confidentiality.
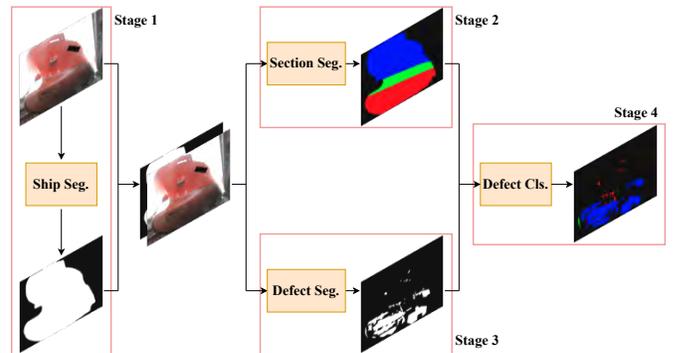
Fig. 4: The multi-stage pipeline used in our system.

the assumption that only a single prominent ship exists in an image. Those less prominent or partially blocked ships will be ignored. This ensures that we focus on one ship in each image for defect detection and evaluation.

For ship section segmentation, we propose to view the three sections (*i.e.*, TS, BT, and VS) as separated by two curves between TS and BT, and BT and VS, so that section segmentation is transformed from predicting 2D maps to

predicting 1D boundaries. An RNN-based line smoothing module is introduced to ensure consistency and smoothness of the predicted 1D line boundaries. We base our model on HorizonNet [26], which was proposed to predict room layouts from panoramic images, and made some major changes to adapt to our scenario. The following subsections will describe the model architecture and some modifications.

A. Network Architecture

The architecture for section segmentation is illustrated in Figure 5. The input is the cropped ship area from whole ship segmentation and the output are two 1D vectors representing the two boundary curves TS/BT and BT/VS. The input image is cropped and resized to 640×480 so that we can have fixed-sized input and output. We use a ResNet-based [27] convolutional neural network (CNN) as a feature extractor to extract the feature maps at four different scales. A height compression module is adopted to compress the feature maps at the vertical direction, which is achieved by four consecutive down-sampling convolutions. Each convolution layer has a stride of 2 at the vertical dimension and a stride of 1 at the horizontal dimension with proper padding to make sure that the height gets compressed while the width is maintained. The resulting flat feature vectors are then aligned to the same width and stacked into a large feature map. Different from the HorizonNet which aligns the width with direct interpolation, we use a fully connected layer to convert the width of feature maps into a uniform sequence length of 160. After we get the large feature map of size 256×160, we can use a linear layer (fully connected layer) to generate the desired output. However, as indicated in HorizonNet, we found the immediate output is not smooth and suffers from some defects. Thus we use a Bidirectional LSTM [28] to smooth the concatenated feature map, before generating the final curve lines.

B. Range-Aware Loss Function

As can be seen from Figure 5, the boundary between TS and BT or between BT and VS does not always span the full width of 640 pixels, even though we have cropped the ship and resized it to fit into the entire input image. The missing part of the green line (big chunk on the top and small chunk on the right) should not be considered in evaluating the performance of our model. Because different ships come with different lengths or spans of boundaries, we cannot make the model accept ground truth labels with variable length while predicting the two curves with exact length. In fact, both the ground truth and predicted labels of our model are of the shape 2×640. To alleviate the mismatch of real boundary length and the fixed-length required by the model, we propose a range-aware loss function that applies a mask over the range of the real boundary and only calculates loss for the range during training. The loss function can be defined as:

$$\mathcal{L} = \sum_{i=1}^{2} \sum_{j=1}^{640} m(i,j) |y_{i,j}^{t} - y_{i,j}^{p}| , \tag{1}$$

where $i \in \{1, 2\}$ is the index for the two boundaries and $j \in [1, 640]$ is the position index in the width direction. $m(i,j) \in$
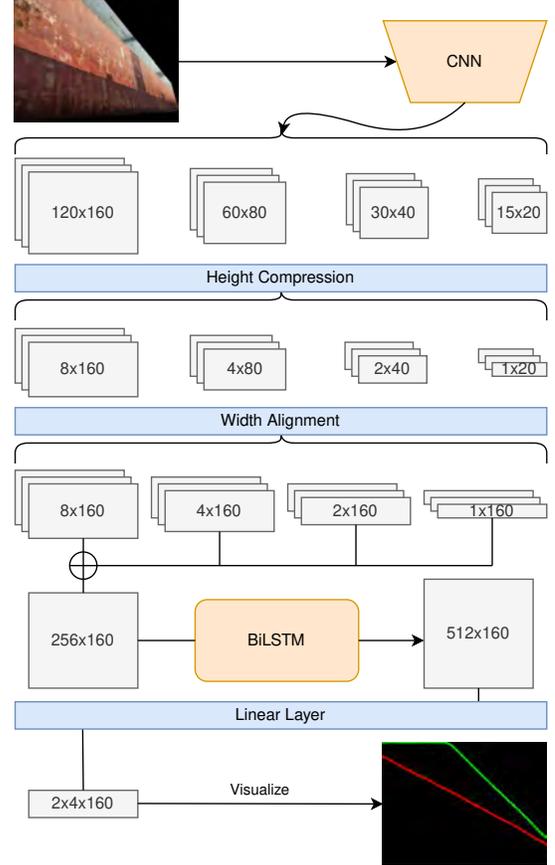
Fig. 5: Architecture of the section segmentation network.

$\{0, 1\}$ is a mask function to indicate if a prediction $y_{i,j}^{p}$ for the ith boundary at the jth position needs to be evaluated. Note that sometimes there are only two sections or even one section in an image, which means we may have one or zero boundary as ground truth, we set $m(i,j)$ as all zeros for the missing boundaries and thus the predictions for them are ignored.

C. Data Processing and Augmentation

As mentioned in Section II, the raw dataset contains both high-quality images (*e.g.* of size 4000×3000) and highly-compressed, low-quality images (*e.g.* of size 400×300), we have to preprocess the data before they can be used for training. For whole ship segmentation, we resize raw images and labeled ship masks into a uniform size 640×480 and use them as input. The output ship masks are of the same size. We then use the predicted ship masks to crop ship areas and resize the ships again to size 640×480. The cropped ships and correspondingly processed section boundary labels, as shown in Figure 5, are finally used for ship section segmentation.

We also perform active data augmentation in section segmentation to make the best use of the 350 labeled images. Because the raw images were photos taken from various angles and distances from the ships, the ship areas presented in images can be various portions of the entire ships, from different perspectives. To enrich the diversity of training data and to make the trained model more robust, we randomly rotate and shift the ship areas within a small range and repeat

(a)

(b)
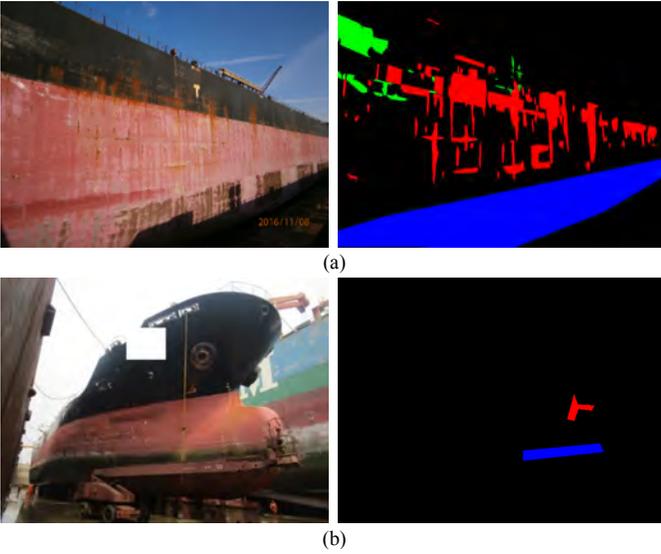
Fig. 6: Examples of weak labels: polygons too coarse (a) and not complete (b). Different colors indicate different types of defects, with red for corrosion, green for delamination, and blue for fouling.

it several times. Moreover, ships come in different colors, with the majority of ships in red color while others in blue color or white color. We believe the color of ships should not affect the determination of the three sections, so we also flip the RGB color channels to BGR as a special data augmentation strategy. As a result, we can generate 3,074 images for training.

V. DEFECT SEGMENTATION AND CLASSIFICATION

A. Defect Segmentation

We apply weakly-supervised learning to the segmentation of defects, as we find the quality of labels is not always consistent. Some labels tend to be more "coarse" than others. Take the large blue polygon in Figure 6(a) as an example, it is supposed to mark those black areas of the ship on the left as fouling, but unnecessarily includes the pale regions which should be regarded as background. On the other hand, some areas with defects are neglected and unlabeled. The ship in Figure 6(b) has a broad, greenish fouling defect at the bottom, but it is only partially labeled. The blue polygon on the right covers a small part of fouling.

To mitigate the negative effects of weak annotations, we propose to utilize a teacher-student training scheme for the defect segmentation task. Similar to GrabCut [29] and BoxSup [30], which start with bounding box annotations and gradually update segmentation masks via iterative model training, we train one model on source labels and use the predictions as input to train another model. As plotted in Figure 7, a teacher model is trained on the original label data, to generate pseudo labels. The predicted labels will get combined with the original labels and then be used to train a student model. Both teacher and student models are UNet-like image segmentation models [11], with ResNet34 [27] as the backbone. We choose UNet because it is widely used in medical image segmentation tasks and the
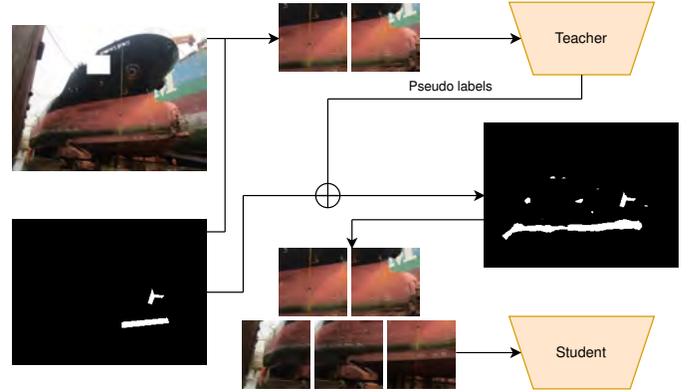
Fig. 7: The teacher-student training scheme.

form of defects in our dataset resembles more of the objects in medical imaging rather than those in natural scenes. A shallow backbone ResNet34 is used instead of deeper feature extractors such as ResNet50 or DenseNet121 [31], as defects are mostly local features which demands less high-level feature extraction and learning.

For training of the teacher model, we slice image-and-label pairs into patches of size 224×224 and remove those without defects or with defect areas less than one percent of the patch. We observed in the labeled data and during experiments that the problem of missing labels is more prominent than coarse labels. The existence of false backgrounds (unlabeled defects) in training data would impact more on the performance than false foregrounds (background labeled as defects). This is reasonable because the areas that are viewed and labeled by human annotators are more reliable than the others that are unattended. That is, though labeled polygons may contain excess regions, they are more accurate than unlabeled regions. As a result, around half of the patches are removed.

After the teacher model is trained for a few epochs, it is used to predict defect labels on the training images. The predicted labels are then combined with the original labels as a new set of labels to generate a new set of training data. Note that by taking the union of pseudo and human labels, we only attempt to solve the problem of missing labels. As aforementioned, it is more prominent than the problem of coarse labels. We also attempted to take the intersection of predicted and human labels at patches containing mostly fouling (coarse labels happen more in fouling than other two defects), the segmentation result, however, was worse than only using human labels. It suggests that even trained on refined data, the teacher model is not able to predict all real fouling correctly. We have to keep the coarse polygons in the source labels to ensure better training of the student model. Finally, the student is trained on updated labels and let run for full epochs to get the final model.

B. Defect Classification

For defect classification, we use a multi-label algorithm rather than a multi-class one. It enables the detection and classification of overlapping different types of defects rather

than only detecting the most prominent one as in the multi-class analogue. In other words, we answer three different Yes/No questions for different defect types, respectively, rather than answering one question with the type of defect.

Moreover, in analyzing defect categories, we have noticed that delamination is the most challenging one to detect. Therefore, we use an additional feature extractor (dubbed Delamination Feature Extractor or DFE) in the network architecture which pays more attention to features related to the detection of delamination defects. To ensure the effectiveness of the added (delamination) feature extractor, we regularize the loss function in training such that the added feature extractor is generating complementary information rather than the same features of the main (general) feature extractor.

Last but not least, we prepend the two feature extractors with spatial transformer networks (STN) [25]. STN can be used to estimate the parameters of affine transformations that can be used to warp the input image patch before passing by the feature extractors. This effectively enables both feature extractors to have different patches as inputs, where each generated patch focuses on the most relevant region of the original input patch. Besides, it enables the network to adaptively crop the input patch to the region that only has defects, *i.e.* focusing on the most important region of a patch. Thus, we have a better classification of the input image patch.

1) Network architecture: As depicted in Figure 8, the network takes patches of images. The preceding defect segmentation network detects regions of interest (RoI) for defects. Then, we consider patches of the input images where we have an RoI ratio > 0.1, *i.e.* we only consider patches of the input image where the defect segmentation suspects that more than 0.1 of the area of the patch is marked as defected. This way the classification algorithm would select patches that have a reasonable area of defects rather than selecting patches arbitrarily. We empirically consider patches of size 64×64 as it is not too large to cause mosaic artifact in the final classification image while being not too small to allow utilization of information from surrounding pixels (such as texture and edges).

Each input patch pass through two STN networks (more details in Section V-B3) that enable the following feature extractors to have different inputs focusing on different regions of the input image patch, if needed. The structure of the STN network is shown in Table I. The output of each STN is a warped copy of the input image patch. We use Densenet-121 [31] as the backbone for our feature extractors which is initially pre-trained over the ImageNet dataset [32]. Densenet is a deep architecture that has skip connections between each layer and all the following layers in each dense block. This ensures the flow of information from one layer to deeper layers, thus mitigating the vanishing gradient problem with deeper architectures [33]. We do not use deeper than Densenet-121 to save computational resources while mitigating the overfitting problem that arises with deeper architectures with limited training data [34]. After extracting the features, an average pool operation is used to eliminate the height and width dimensions, thus we end up with a one-dimensional vector of length 1024. The average pool operation enables
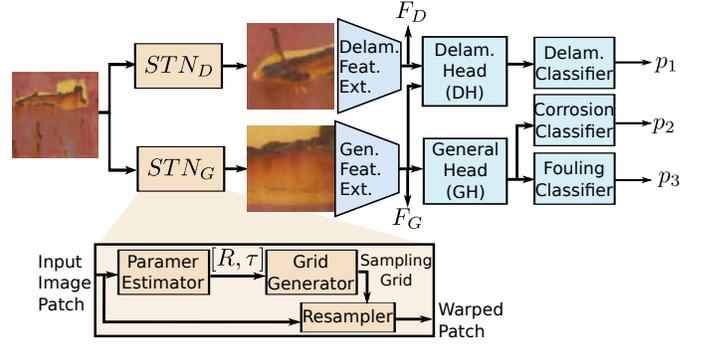
Fig. 8: Architecture of the defect classification network.

the network to work with any arbitrary size because the latter layers are fully connected. The structure of the feature extractors is detailed in Table II. The extracted two feature vectors are denoted by F_D and F_G for Delamination-specific features and general features, respectively. F_G passes then by two fully-connected layers (named General Head or GH) with a ReLU activation to reduce the feature size to 256. Similarly, we concatenate F_G and F_D and pass it by Delamination Head (DH) to reduce the size to 256. At the last stage, three fully connected networks of two layers are used as the last classification stage for each class independently. The first layer has a ReLU activation function while the last one has a Sigmoid activation. Therefore, the outputs of the sub-networks p_1, p_2 and p_3 are three values that range from 0 to 1 which represents the probabilities of different type of defects (Corrosion, Fouling or Delamination). The details of the GH, DH, and the classifier sub-networks are shown in Table III.

2) Delamination Feature Extractor and regularization: Since delamination defects are the most difficult type of defects to detect and classify, we use an additional feature extractor specifically for delamination defects. The sole purpose of that special feature extractor is to find additional features that would make it easier to detect delamination defects. However, to ensure the validity of the added feature extractor and that it is performing its expected objective, we add a regularization term to the loss function which ensures that the newly extracted features are complementary to the features extracted by the general feature extractor. Therefore, our customized training loss function is as follows.

$$\mathcal{L} = \sum_{i=1}^{3} w_i \mathcal{L}_{\text{BCE}}^{(i)}(l_i, p_i) + \lambda \cdot |\text{CosSim}\,(F_G, F_D)| \,, \quad (2)$$

where $\mathcal{L}_{\text{BCE}}^{(i)}(l_i, p_i)$ is the Binary Cross Entropy [35] which is defined as:

$$\mathcal{L}_{\text{BCE}}^{(i)}(l_i, p_i) = l_i \log p_i + (1 - l_i) \log(1 - p_i) \,, \quad (3)$$

where $l_i \in \{0, 1\}$ is the ground-truth label of class $i \in \{1, 2, 3\}$ representing corrosion, fouling, and delamination, respectively. p_i is the predicted output of class i which takes values between 0 and 1. We multiply each BCE loss term with a weight value w_i to compensate for the imbalance in the labeled dataset. We set $w_1 = 1$, $w_2 = 2$ and $w_3 = 4$. This way we implicitly give more focus to fouling and delamination

7

TABLE I: Structure of the Spatial Transformer Networks (STNs)

	STN.1	STN.2	STN.3	STN.4	GridGen	Resampler
Input	Image Patch $64 \times 64 \times 3$	STN.1	STN.2	STN.3	STN.4	[Image Patch, GridGen]
Structure	$\begin{bmatrix} 7 \times 7 \text{ conv.} \\ \text{ReLU} \end{bmatrix}$	$\begin{bmatrix} 5 \times 5 \text{ conv.} \\ \text{ReLU} \end{bmatrix}$	$\begin{bmatrix} 3 \times 3 \text{ conv.} \\ \text{ReLU} \end{bmatrix}$	$\begin{bmatrix} 1 \times 1 \text{ conv.} \\ \text{avg-pool} \end{bmatrix}$	Sample Grid Generator	2D interpolation based on the sampling grid
Output	$64 \times 64 \times 100$	$64 \times 64 \times 100$	$64 \times 64 \times 50$	$1 \times 1 \times 6$	$64 \times 64 \times 1$	$64 \times 64 \times 3$

TABLE II: Structure of the feature extractors sub-networks – General Feature Extractor and Delamination Feature Extractor

	Base	Dense.1	Trans.1	Dense.2	Trans.2	Dense.3	Trans.3	Dense.4
Input	Warped Image Patch $64 \times 64 \times 3$	Base	Dense.1	Trans.1	Dense.2	Trans.2	Dense.3	Trans.3
Structure	$\begin{bmatrix} 7 \times 7 \text{ conv.} \\ 3 \times 3 \text{ max-pool} \end{bmatrix}$	$\begin{bmatrix} 1 \times 1 \text{ conv.} \\ 3 \times 3 \text{ conv.} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv.} \\ 2 \times 2 \text{ avg-pool} \end{bmatrix}$	$\begin{bmatrix} 1 \times 1 \text{ conv.} \\ 3 \times 3 \text{ conv.} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv.} \\ 2 \times 2 \text{ avg-pool} \end{bmatrix}$	$\begin{bmatrix} 1 \times 1 \text{ conv.} \\ 3 \times 3 \text{ conv.} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv.} \\ 2 \times 2 \text{ avg-pool} \end{bmatrix}$	$\begin{bmatrix} 1 \times 1 \text{ conv.} \\ 3 \times 3 \text{ conv.} \end{bmatrix} \times 16$
Output	$16 \times 16 \times 64$	$16 \times 16 \times 256$	$8 \times 8 \times 128$	$8 \times 8 \times 512$	$4 \times 4 \times 256$	$4 \times 4 \times 1024$	$2 \times 2 \times 512$	$1 \times 1 \times 1024$

TABLE III: Structure of the classifiers

	General Head		Delamination Head		Classifiers	
	GH.1	GH.2	DH.1	DH.2	C.1	C.2
Input	F_G	GH.1	$[F_G, F_D]$	DH.1	GH.2 or DH.2	C.1
Structure	$\begin{bmatrix} 1024 \rightarrow 512 \text{ Linear} \\ \text{ReLU} \\ \text{Dropout} \end{bmatrix}$	$\begin{bmatrix} 512 \rightarrow 256 \text{ Linear} \\ \text{ReLU} \\ \text{Dropout} \end{bmatrix}$	$\begin{bmatrix} 2048 \rightarrow 512 \text{ Linear} \\ \text{ReLU} \\ \text{Dropout} \end{bmatrix}$	$\begin{bmatrix} 512 \rightarrow 256 \text{ Linear} \\ \text{ReLU} \\ \text{Dropout} \end{bmatrix}$	$\begin{bmatrix} 256 \rightarrow 128 \text{ Linear} \\ \text{ReLU} \\ \text{Dropout} \end{bmatrix}$	$\begin{bmatrix} 128 \rightarrow 1 \text{ Linear} \\ \sigma \end{bmatrix}$
Output	512×1	256×1	512×1	256×1	128×1	1

defects in the training process over the corrosion defects, because corrosion defects are the easiest type of defects to classify in comparison to delamination and fouling defects.

The regularization term $\text{CosSim}(F_G, F_D)$ is the cosine similarity [36] between the extracted features from the general feature extractor F_G and the delamination feature extractor F_D. λ is a regularization parameter to tune the importance of the regularization term in the training process in comparison to the BCE loss term. We set λ to 1 through training by cross-validation [37]. The objective of the cosine similarity regularization is to ensure that $F_D \neq \alpha \cdot F_G$ for some scale factor α. Therefore, F_D carries complementary information to F_G which helps in the final classification stage. CosSim measures the angle between the two latent feature vectors and attempts to maximize it by minimizing the inner product of the two latent vectors. In other words, we train our network in a weakly-supervised manner since the size of the training set is not large enough. $\text{CosSim}(x, y) = x^T y / \max(||x|| \cdot ||y||, \epsilon)$. The inner product is divided by $\max(||x|| \cdot ||y||, \epsilon)$ for normalization, where $\epsilon = 10^{-4}$ for numerical stability.

3) Spatial Transformer Network (STN): Before passing the input image patch by the two feature extractors, we first pass it by two STNs to perform affine transformations [38] to make it easier for each feature extractors to find the most relevant features. The Spatial Transformer Network [25] estimates 6 parameters to perform an affine transformation which consists of 4 parameters to perform rotation and scaling R and 2 parameters for translation τ as follows.

$$\begin{pmatrix} x_j^{(n)} \\ y_j^{(n)} \end{pmatrix} = \mathcal{T}_\theta \left(\begin{pmatrix} x_j \\ y_j \end{pmatrix} \right)$$
$$= \underbrace{\begin{pmatrix} r_{xx} & r_{xy} \\ r_{yx} & r_{yy} \end{pmatrix}}_{R} \underbrace{\begin{pmatrix} t_x \\ t_y \end{pmatrix}}_{\tau} \times \begin{pmatrix} x_j \\ y_j \\ 1 \end{pmatrix}, \quad (4)$$

where $x_j \& y_j$ are the coordinates of pixel j of the original input patch and $x_j^{(n)} \& y_j^{(n)}$ are the new coordinates.

Utilization of the STN modules in DFE-NET allows each feature extractor to obtain different features. It makes the problem much easier for the subsequent classifier subnetwroks. As depicted in Figure 8, each STN warps the input image differently. For instance, delamination defects usually depend on information related to edges (as it is a result of the detachment of the paint from the surface). Therefore, warping the input image patch to see edges more prominently will make the detection process of delamination defects easier. This is performed by STN_D. However, corrosion and fouling defects usually depend on information related to the texture and colors (not edges unlike delamination). Therefore, STN_G needs to warp the image differently to focus on the texture not the boundary changes from one surface to another.

VI. EXPERIMENTAL RESULTS

A. Ablation Study

We have performed an extensive set of ablation studies to verify and validate the importance of each module. We start by discussing the effectiveness of some modules related to the final classification network such as the additional encoder for delamination defects, the regularization term in the loss function, and the STN modules. Then, we study the effectiveness of employing the section segmentation results in the classification process as well.

1) Effectiveness of the Delamination Feature Extractor and regularization in the classification network: To study the effectiveness of the special feature extractor, that is the Delamination Feature Extractor (DFE), we train the classification network for 20 epochs by randomly choosing 500 images for training and 20 images for testing and repeating the same training/test split for three times then we calculate the average

TABLE IV: Effectiveness of the Delamination Feature Extractor and Regularization on the detection of Corrosion Defects

	without DFE	with DFE	
		no regularizer	with regularizer
Accuracy	0.7636	0.7834	**0.8009**
B. Accuracy	0.7854	0.7923	**0.8153**
F1-Score	0.7872	0.8076	**0.8319**

TABLE V: Effectiveness of DFE and Regularization on the detection of Delamination Defects

	without DFE	with DFE	
		no regularizer	with regularizer
Accuracy	0.6347	0.6789	**0.7103**
B. Accuracy	0.5756	0.6250	**0.6455**
F1-Score	0.5099	0.5808	**0.6061**

values of accuracy, balanced accuracy (B. Accuracy), F1-score and confusion matrices as follows.

$$\text{ACCURACY} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} , \quad (5)$$

$$\text{B. ACCURACY} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{TN} + \text{FP}} \right) , \quad (6)$$

$$F_1 = 2\frac{\text{PRECISION} \cdot \text{RECALL}}{\text{PRECISION} + \text{RECALL}} , \quad (7)$$

$$\text{PRECISION} = \frac{\text{TP}}{\text{TP} + \text{FP}} , \quad (8)$$

$$\text{RECALL} = \frac{\text{TP}}{\text{TP} + \text{FN}} , \quad (9)$$

where TP, TN, FP, and FN are acronyms for True Positive, True Negative, False Positive, and False Negative, respectively.

Tables IV to VI show the performance of the classification network for Corrosion, Delamination, and Fouling defects respectively in different scenarios: **i)** without DFE, **ii)** with DFE but no regularization term in the training loss function and **iii)** with DFE and regularization term in the training loss function. In addition, Tables VII to IX show the average confusion matrices values of the test sets at different cases of using DFE and regularization term for Corrosion, Delamination, and Fouling Defects, respectively.

The results indicate that the added Delamination Feature Extractor (DFE) can boost the performance of the network for the delamination type of defects that are most challenging to detect and classify. Moreover, it also boosts the overall performance of the network for other types of defects; namely corrosion and fouling. A justification for such a behavior can be that the General Feature Extractor (GFE) is now giving more focus to corrosion and fouling types of defects since the DFE is solely focusing on delamination defects. Moreover, since corrosion and fouling defects are generally identified by the texture of the defect unlike delamination which is typically identified by the boundaries, the GFE is focusing on extracting features related to the texture while the DFE is focusing on extracting features related to changes at the edges of a defect.

*2) **Effectiveness of the STN modules in the classification network**:* We also perform another ablation study to confirm the viability of the STN module. Although we have noticed that the STN module is not as important as the novel DFE

TABLE VI: Effectiveness of the Delamination Feature Extractor and Regularization on the detection of Fouling Defects

	without DFE	with DFE	
		no regularizer	with regularizer
Accuracy	0.7387	0.7697	**0.7883**
B. Accuracy	0.7436	0.7866	**0.7953**
F1-Score	0.7528	0.7759	**0.8013**

TABLE VII: Confusion matrices of *corrosion* defects for different cases of using DFE and regularization.

module and regularization in terms of increasing the accuracy and balanced accuracy, it was able to give a consistent increase in those metrics for all types of defects. The reason for that better classification could be the limited training set which gives more importance to the STN module. To elaborate, STN module's sole purpose is to learn an affine transformation performed over the input patch so that it can be classified accurately down the line. Its importance magnifies with the smaller size of the training set.

To validate this conclusion, we have performed a limited training case study, where we have only used 100, 200, and 300 sets of images for training and used 30 images as a test set. Then, we measured the accuracy, balanced accuracy, and F1-score for each set. The results of each test set are depicted in Figure 9, where the average is taken over the three types of defects. As can be shown from the figure, albeit the STN module boosts the performance, but its effect is much more significant by reducing the training dataset size. Because The network (without STN module) cannot generalize well with a smaller training set and it is much prone to overfitting. Therefore, the STN module has a similar-to regularization effect in the case of a smaller training set. As shown in Figure 9, the STN module allows the network to have much more graceful decay in performance when the size of the

TABLE VIII: Confusion matrices of *delamination* defects for different cases of using DFE and regularization.

TABLE IX: Confusion matrices of *fouling* defects for different cases of using DFE and regularization.

TABLE X: Mean IoU of TS/BT/VS

Method	TS	BT	VS
UNet [11]	0.7459	0.4669	0.5360
HorizonNet [26]	0.7827	0.5717	0.6800
Ours	0.8542	0.6641	0.7535

TABLE XI: Mean IoU, precision, and recall rates

Method	IoU	Precision	Recall
UNet [11]	0.4924	0.6373	0.6523
Ours	0.5205	0.6314	0.7725

training set is reduced by a factor of 3.

3) Utilization of the section segmentation output in the classification process: We have noticed that the distribution of defects varies depending on the location; whether at TS, BT, or VS. Prominently, Fouling defects never happen in the the TS section of the ship because TS is not under water where the fouling organisms are. At post-processing, we remove all fouling from the TS section. We show four different examples of a test set in Figure 10. Notably, the network (without employing section segmentation results) detects incorrect fouling defects in the top section of the ship (TS), while it is typically rare to occur. Therefore, the results of the network after employing the section segmentation results are much more consistent and close to the ground truth labels.

B. Results

1) Ship section segmentation: We compare our model with two baselines, UNet and HorizonNet. The result is evaluated as mean IoU of TS, BT, and VS over 35 test samples, as shown in Table X. From the mean IoUs, we know our method is consistently better than the two baselines. Note that the prediction from our model is two 1D vectors and they have to be combined with the whole ship segmentation to get the 2D segmentation map.

2) Defect segmentation: We also compare the result in defect segmentation with one baseline UNet. Since we use UNet architectures for both teacher and student models, the baseline here is equal to the teacher model. For the comparison, we selected 60 vessel images and the result is evaluated as three metrics: mean IoU, precision, and recall, as shown in Table XI. Because of the improvement of labels from the teacher model, our student model achieves a better recall rate and IoU. The precision does not improve much because the inclusion of pseudo labels inevitably introduced noise that makes the student tends to predict more defects than the teacher. This leads to the rise in recall rate at the sacrifice of the precision score.

3) Defect classification: To the best of our knowledge, there is no current published method that can detect defects of marine vessels. Therefore, it is not possible to compare the performance of our network with another state-of-the-art method. However, to have a justified and fair evaluation of our classification algorithm, we compare DFE-NET with both ResNet [27] and DenseNet [31] based implementation without the novel DFE additional encoder nor the STN module. Moreover, to ensure the reliability of the network in a real-life scenario, we compare the performance of our network to human experts in the detection of defects in marine vessels for 60 non-labeled images of marine vessels.

Tables XIII to XV show the confusion matrices of different types of defects for our network in comparison to DenseNet- and ResNet-based architectures. The results are calculated over 50 labeled test images. It is clear that our network performs better than ResNet and DenseNet based networks; mainly due to the DFE and the STN module as has been discussed in Sections VI-A1 and VI-A2.

The second experiment that we have performed is to test the performance of DFE-NET compared to human experts. With the help of experts in the field from PPG, we were able to obtain the percentages of defects in each section of 60 vessel images. These results were calculated by 6 human experts in the analysis of vessel defects. Table XII shows the predicted percentage of defects of different methods in comparison with the human experts. It is noteworthy that not all of the images have all sections of the vessel. Therefore, the average of defects at a section is taken considering only images that contain this section only. For instance, an image may not contain a TS section. Thus, we will not include it in the statistical calculations of the percentage of the defected area of the TS section. This leads to an effectively lower number of images for the statistical analysis and thus less confidence in the results. However, it is not that significant, as the number of images that includes a specific section is more than 40, which is more than enough for high confidence in the analysis. Figure 11 shows four examples from the test set, where DFE-NET was successful to obtain results that match the provided labeled data.

4) Discussion about multi-label vs. multi-class architecture: We have adopted a multi-label approach in the classification process of defects. However, another possible approach is to use a multi-class approach; which only detects a single class (the most prominent class) in a patch. The multi-class approach answers one question: What type of defect in the patch, corrosion, delamination, or fouling? The multi-label approach attempts to answer three questions instead: (1) Does the patch have a corrosion defect? (2) Does the patch have a delamination defect? and (3) Does the patch have a fouling defect?

We show here a comparison of both approaches. In conclusion, using a multi-label approach is better than a multi-class one. There are two reasons for that. First, the percentage of patches that have overlapping defects (at least two defects) cannot be neglected. Second, the defect segmentation algorithm may falsely detect defects. Multi-label classification can further clean some of those erroneously detected defects by answering its three questions by No.

For the comparison, we have selected 20 images as a test set. To elaborate on the importance of the multi-label approach, those selected images have overlapping defects. Thus, many of the generated patches of those images have multiple labels. Table XVII shows the accuracy, the balanced accuracy, and the F1-score in both cases. The multi-label approach is better for the detection and classification of defects in marine vessels for the aforementioned reasons.

(a) Accuracy

(b) Balanced Accuracy

(c) F1-Score
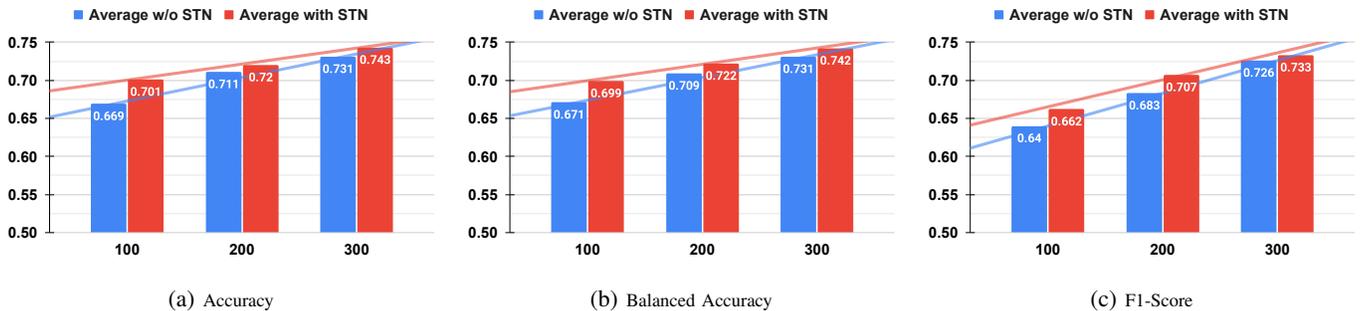
Fig. 9: Performance evaluation of the classification network with and without STN module at different sizes of the training dataset. The shown results are the average over the three defect types.

TABLE XII: Section wise average defect percentage across 60 vessel images. Note that multiple experts rate each image, hence we first compute the average over the 60 images of each inspector, then we compute the mean and standard deviation over the average ratings of the inspectors. Experts values are written as mean μ (std. dev. σ).

Defect	Corrosion			Delamination			Fouling		
section	TS	BT	VS	TS	BT	VS	TS	BT	VS
Experts	11.08 (3.34)	20.36 (5.09)	8.90 (4.98)	3.84 (1.09)	13.67 (3.90)	25.83 (8.41)	0.42 (0.01)	35.12 (4.80)	66.03 (9.17)
ResNet [27]	6.08	12.08	5.98	2.18	8.07	18.04	3.74	31.08	54.30
DenseNet [31]	8.81	15.30	5.72	3.14	7.94	19.46	5.31	29.50	56.72
DFE-NET	9.84	18.52	8.01	5.20	10.12	22.74	0	38.17	61.82

TABLE XIII: Confusion matrices of *corrosion* defects for different architectures.

TABLE XIV: Confusion matrices of *delamination* defects for different architectures.

TABLE XV: Confusion matrices of *fouling* defects for different architectures.

TABLE XVII: Numerical results of multi-class approach vs multi-label approach

	Accuracy	B. Accuracy	F1-Score
Multi-Class	0.62	0.54	0.63
Multi-Label	**0.78**	**0.76**	**0.72**

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we built the first dataset for surface defect detection of marine vessels and proposed a multi-stage deep learning framework to evaluate the percentage of three types of defects over three sections of a ship. The prediction of sections (TS, BT, VS) of a ship is converted from 2D masks to 1D boundaries so that less training data is needed and the result is more consistent. We utilized a teacher-student training scheme for defect segmentation to tackle the problem of coarse and incomplete labels. A teacher model is firstly trained on labeled data and used to generate pseudo labels. The generated labels are then combined with original labels to train a student model to achieve better performance. For defect classification, we proposed a multi-label classification network that includes two detection heads for challenging defect (delamination) detection and STNs to estimate affine transformation for input patches before feature extraction such that patches are warped for better alignment. Our individual modules achieved better performance than state-of-the-art methods and comparable performance to human inspectors.

Aspects of our approach can be further improved. An end-to-end framework in place of the defect segmentation and classification modules may be desired and possibly reduce inference cost. The domain knowledge that fouling usually happens at the lower part of a ship can be utilized in other ways. Currently, we make use of it by post-processing to remove fouling from TS. Future work may consider estimating the relative position of fouling in the ship and enforce it in loss function or remove it before final prediction.
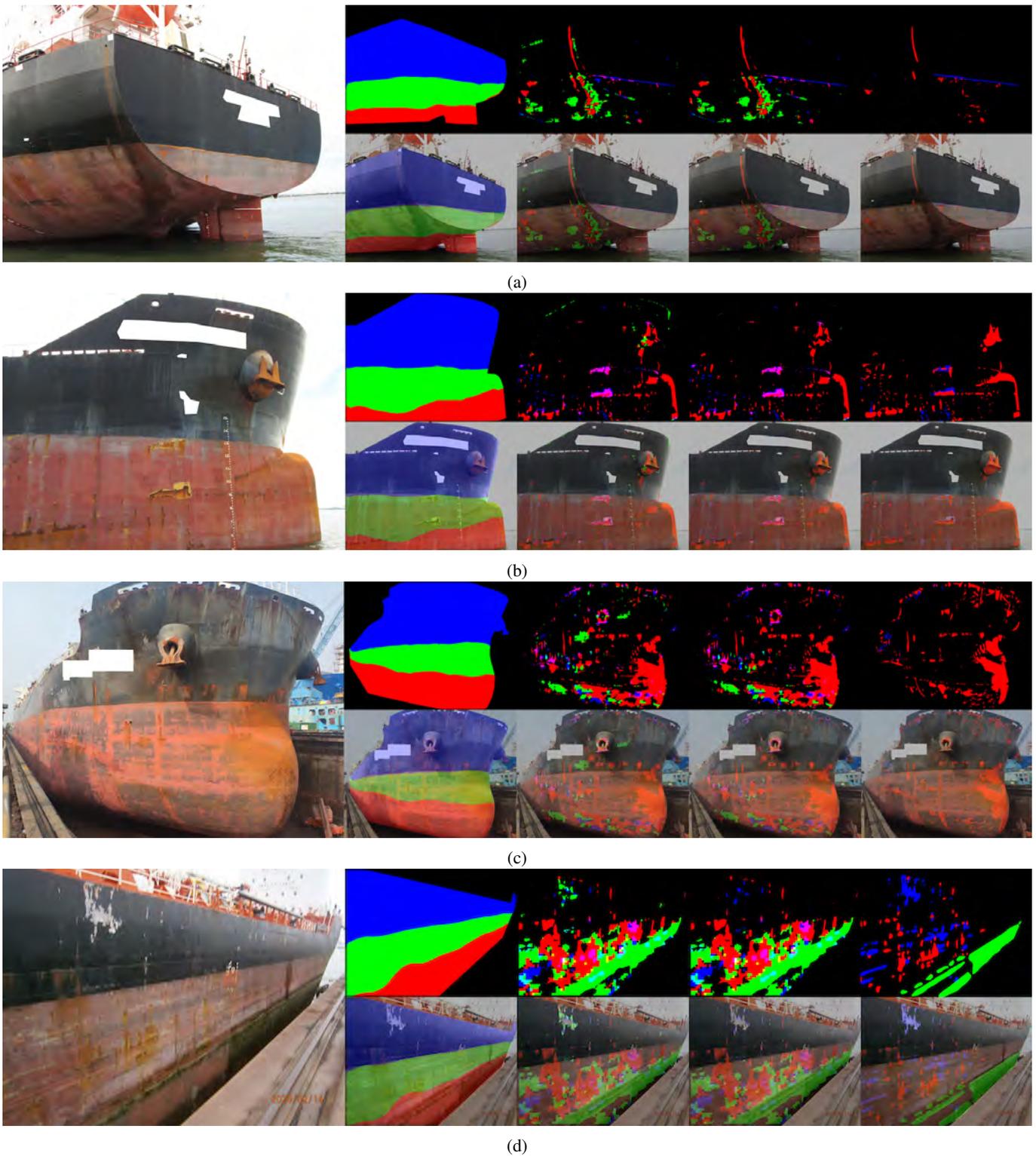
Fig. 10: Four examples of the test set. From left to right, i) input image, ii) section segmentation results, iii) defect classification without employing section segmentation, iv) defect classification employing section segmentation and v) ground-truth labels. The first row of each sample shows the results/labels and the second row shows the overlaid results/labels over the input image.
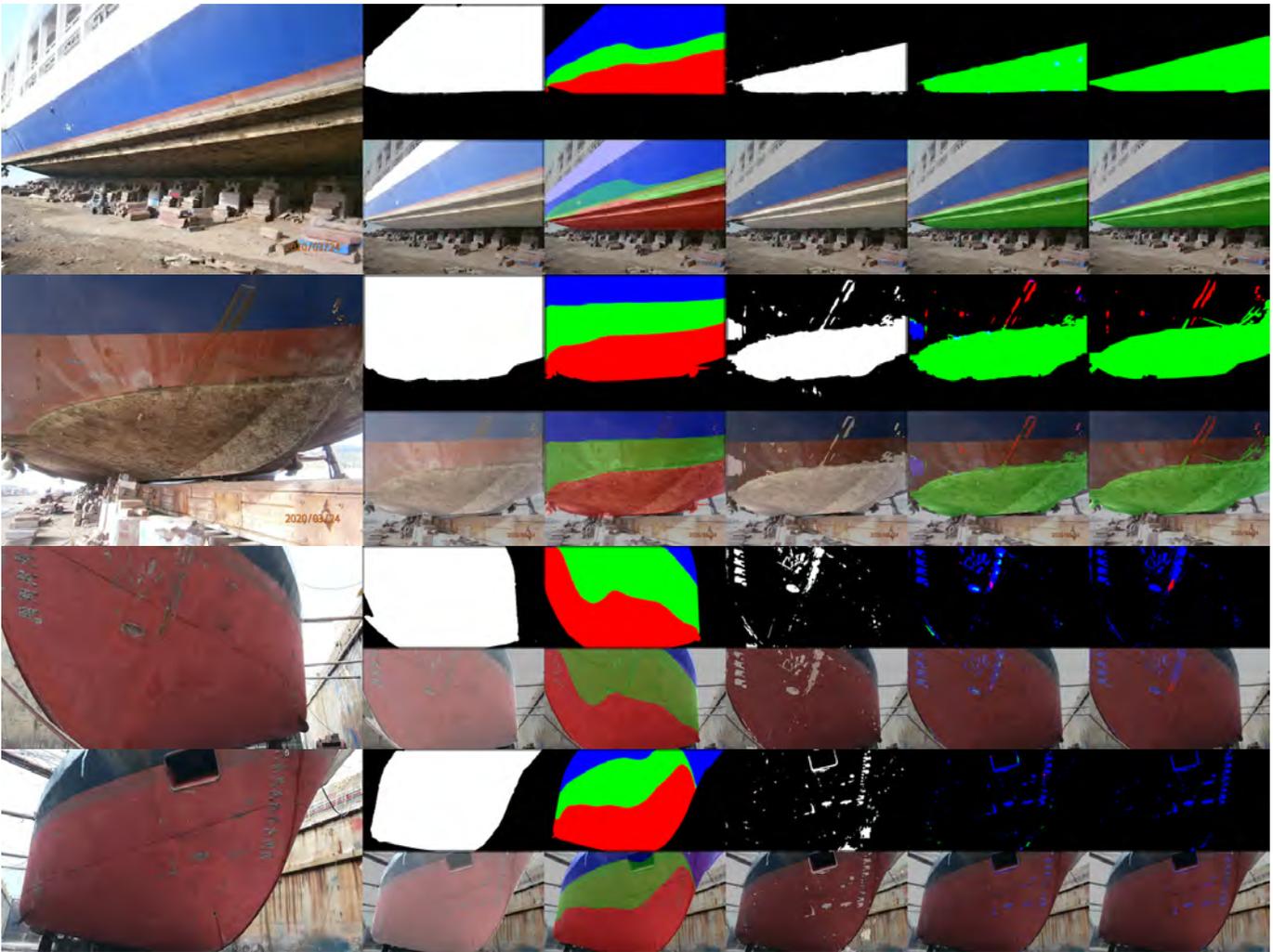
Fig. 11: Four examples that the algorithm is doing the same as the provided labeled data.

REFERENCES

[1] P. Navarro, A. Iborra, C. Fernández, P. Sánchez, and J. Suardíaz, "A sensor system for detection of hull surface defects," *Sensors*, vol. 10, no. 8, pp. 7067–7081, 2010.

[2] A. Aijazi, L. Malaterre, M. Tazir, L. Trassoudaine, and P. Checchin, "Detecting and analyzing corrosion spots on the hull of large marine vessels using colored 3d lidar point clouds." *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 3, no. 3, 2016.

[3] A. Jalalian, W. F. Lu, F. S. Wong, S. M. Ahmed, and C.-M. Chew, "An automatic visual inspection method based on statistical approach for defect detection of ship hull surfaces," in *Proceedings of the IEEE International Conference on Automation Science and Engineering*. IEEE, 2018, pp. 445–450.

[4] S. Susan and M. Sharma, "Automatic texture defect detection using Gaussian mixture entropy modeling," *Neurocomputing*, vol. 239, pp. 232–237, 2017.

[5] C. Fernández-Isla, P. J. Navarro, and P. M. Alcover, "Automated visual inspection of ship hull surfaces using the wavelet transform," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[6] F. Bonnín-Pascual and A. Ortiz, "Detection of cracks and corrosion for automated vessels visual inspection," in *Proceedings of the 13th International Conference of theCatalan Association for Artificial Intelligence*, 2010, pp. 111–120.

[7] T. Yamaguchi and S. Hashimoto, "Fast crack detection method for large-size concrete surface images using percolation-based image processing," *Machine Vision and Applications*, vol. 21, no. 5, pp. 797–809, 2010.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[9] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.

[10] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[12] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Pro-*

ceedings of the Fourth International Conference on 3D Vision. IEEE, 2016, pp. 565–571.

[13] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "UNet++: A nested U-Net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 3–11.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[16] L.-C. Chen, A. Hermans, G. Papandreou, F. Schroff, P. Wang, and H. Adam, "Masklab: Instance segmentation by refining object detection with semantic and direction features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4013–4022.

[17] Y. Lee and J. Park, "CenterMask: Real-time anchor-free instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13906–13915.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[19] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[20] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[21] X. Wei, S. Jiang, Y. Li, C. Li, L. Jia, and Y. Li, "Defect detection of pantograph slide based on deep learning and image processing technology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 947–958, 2019.

[22] R. Fan and M. Liu, "Road damage detection based on unsupervised disparity map segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[23] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[25] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015, pp. 2017–2025. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/33ceb07bf4eeb3da587e268d663aba1a-Paper.pdf

[26] C. Sun, C.-W. Hsiao, M. Sun, and H.-T. Chen, "HorizonNet: Learning room layout with 1d representation and pano stretch data augmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1047–1056.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[29] C. Rother, V. Kolmogorov, and A. Blake, """ grabcut" interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[30] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1635–1643.

[31] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp. 2261–2269.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[33] H. H. Tan and K. H. Lim, "Vanishing gradient mitigation with deep learning neural network optimization," in *Proceedings of the International Conference on Smart Computing Communications (ICSCC)*, 2019, pp. 1–4.

[34] K. Metwaly and V. Monga, "Attention-mask dense merger (attendense) deep hdr for ghost removal," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2623–2627.

[35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[36] T. Pellegrini and L. Cances, "Cosine-similarity penalty to discriminate sound classes in weakly-supervised sound event detection," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[37] V. Monga, Ed., *Handbook of Convex Optimization Methods in Imaging Science*. Cham: Springer International Publishing, 2018. [Online]. Available: http://link.springer.com/10.1007/978-3-319-61609-4

[38] J. Gu and V. Tresp, "Improving the robustness of capsule networks to image affine transformations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.
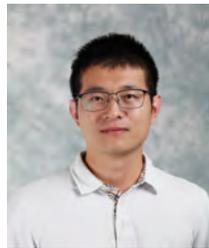
Li Yu received his MS degree in Informatics from the School of Information Sciences and Technology at The Pennsylvania State University. He received his bachelor's degree with double major in chemistry and mathematics from Peking University and the MS degree in chemistry from Emory University. His research interests lie in the field of computer vision and image processing.

Kareem Metwaly is a PhD candidate at the School of Electrical Engineering and Computer Science, The Pennsylvania State University. He obtained both his bachelor's degree in Electrical Engineering and MS degree in Engineering Mathematics from Alexandria University, Egypt. His research interests lie in the intersection of machine learning, specifically deep learning, with computer vision and image processing. He worked previously in Brightskies Technologies, Alexandria, Egypt for two years, where he worked on implementing mathematically and computationally high-performance algorithms. He is currently doing an internship at Toyota Motor North America, where he is using machine learning strategies and approaches to solve some computer vision tasks for vehicles. More information about him is at http://personal.psu.edu/kmm1122

James Z. Wang is a Professor of Information Sciences and Technology at The Pennsylvania State University. He received the bachelor's degree in mathematics *summa cum laude* from the University of Minnesota, and the MS degree in mathematics, the MS degree in computer science, and the PhD degree in medical information sciences, all from Stanford University. His research interests include image analysis, image modeling, image retrieval, and their applications. He was a visiting professor at the Robotics Institute at Carnegie Mellon University (2007-2008), a lead special section guest editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence (2008), and a program manager at the Office of the Director of the National Science Foundation (2011-2012). He was a recipient of a National Science Foundation Career award (2004) and Amazon Research Awards (2018, 2019).

Vishal Monga received his Ph.D. degree in electrical engineering from the University of Texas at Austin. Currently, he is a professor in the School of Electrical Engineering and is a member of the electrical engineering and computer science faculty at Pennsylvania State University, University Park, Pennsylvania, 16802, USA. He is an elected member of the IEEE Image Video and Multidimensional Signal Processing Technical Committee and a senior area editor of IEEE Signal Processing Letters, and he has served on the editorial boards of IEEE Transactions on Image Processing, IEEE Signal Processing Letters, and IEEE Transactions on Circuits and Systems for Video Technology. He is a recipient of the U.S. National Science Foundation CAREER Award and a 2016 Joel and Ruth Spira Teaching Excellence Award. His research interests include optimization-based methods with applications in signal and image processing, learning, and computer vision. He is a Senior Member of IEEE.