Results in Engineering 4 (2019) 100036

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.editorialmanager.com/rineng/Default.aspx

Can a computer see what an ice expert sees? Multilabel ice objects classification with convolutional neural networks☆

Ekaterina Kim [a,b,*], Gurvinder Singh Dahiya [c], Sveinung Løset [b], Roger Skjetne [a,b]

[a] Centre for Autonomous Marine Operations and Systems (AMOS), Norway
[b] Centre for Sustainable Arctic Marine and Coastal Technology (SAMCoT), Norwegian University of Science and Technology, Trondheim, Norway
[c] DAHIYA ENK., Trondheim, Norway

ARTICLE INFO

Keywords:
Ice navigation
Ice conditions
Arctic

ABSTRACT

Computer-aided scene analysis has drawn much attention, especially in autonomous navigation and advanced navigation assistance systems for surface vessels. In ice-infested waters, multilabel ice object classification and segmentation form the core of these systems, which are required for path-planning and collision avoidance algorithms. This study focuses on the interpretation of ice conditions from close-range optical imagery. It presents a model for multilabel ice object classification that builds on state-of-the-art open source libraries and deep learning platforms. This work explores the generalization ability of open source models to differentiate between nine categories of surface ice features: level ice, deformed ice, broken ice, icebergs, floebergs, floebits, ice floes, pancake ice, and brash ice. The results demonstrate the ability of the models to classify these nine categories from optical close-range images, which were gathered online and during a research cruise to the Fram Strait on the RV Lance in 2012. We tested a variety of classification algorithms on the collected ice imagery and compared the results against randomly selected test cases representing different ice features with different degrees of local texture distortion. In doing so, we can evaluate the effectiveness of the classification of different classes and compare different levels of information presented for the classification. In addition, we provide a model implementation: a GitHub repository, ICEXPERT, that is suitable for ice object classification from close-range ice imagery.

Introduction

Currently, all navigation of surface vessels in ice-infested waters is done largely as a manual task that requires much training and experience from sailing in icy waters. The latter includes consideration of ice among other factors such as topography, currents, metrology, etc., that are described in [1] and schematically shown in Fig. 1. To determine the safest route, the captain must consider ice types and overall ice conditions.

The World Meteorological Organization [2] developed well-established nomenclature for classifying sea ice. A trained ice navigator is required to identify various shapes and forms of ice, to reliably recognize ice and, when possible, to avoid the most dangerous forms of sea ice (e.g., massive ice features such as ice ridges). Distinguishing between first-year, second-year, and multi-year ice may be challenging in good weather [3]. In fog, snow, and darkness, this task becomes even more difficult. The age of the ice may be determined by using its color, thickness, freeboard, floe shape, size, ponding/drainage, hummocking, etc., and the details can be found in the information that is reported in [3]. New ice is grey, and water can be seen through the ice. Light blue colored ice indicates stronger ice (multi-year ice); and contact with this ice should be avoided. It is also recommended to avoid ice that is covered by snow since it causes additional friction on the hull. Flat and even ice cover is preferable over hummocks.

The number of vessels in ice-covered waters grows faster than the number of trained professionals that can safely and efficiently navigate through the ice. Computer-aided understanding of the ice conditions in front of a surface vessel becomes important for pilot assistance systems,

☆ Note that we have only set the random seed for data split between training, validation and testing, and the random seed for the initialization of the network layers has not been set (refer to the ICEXPERT code). Thus, small variations in the reported model performances are anticipated; however, the main trends that are reported in this study will not change.

* Corresponding author. Norwegian University of Science and Technology (NTNU), Trondheim, Norway.
E-mail address: ekaterina.kim@ntnu.no (E. Kim).

https://doi.org/10.1016/j.rineng.2019.100036
Received 1 August 2019; Received in revised form 15 September 2019; Accepted 16 September 2019

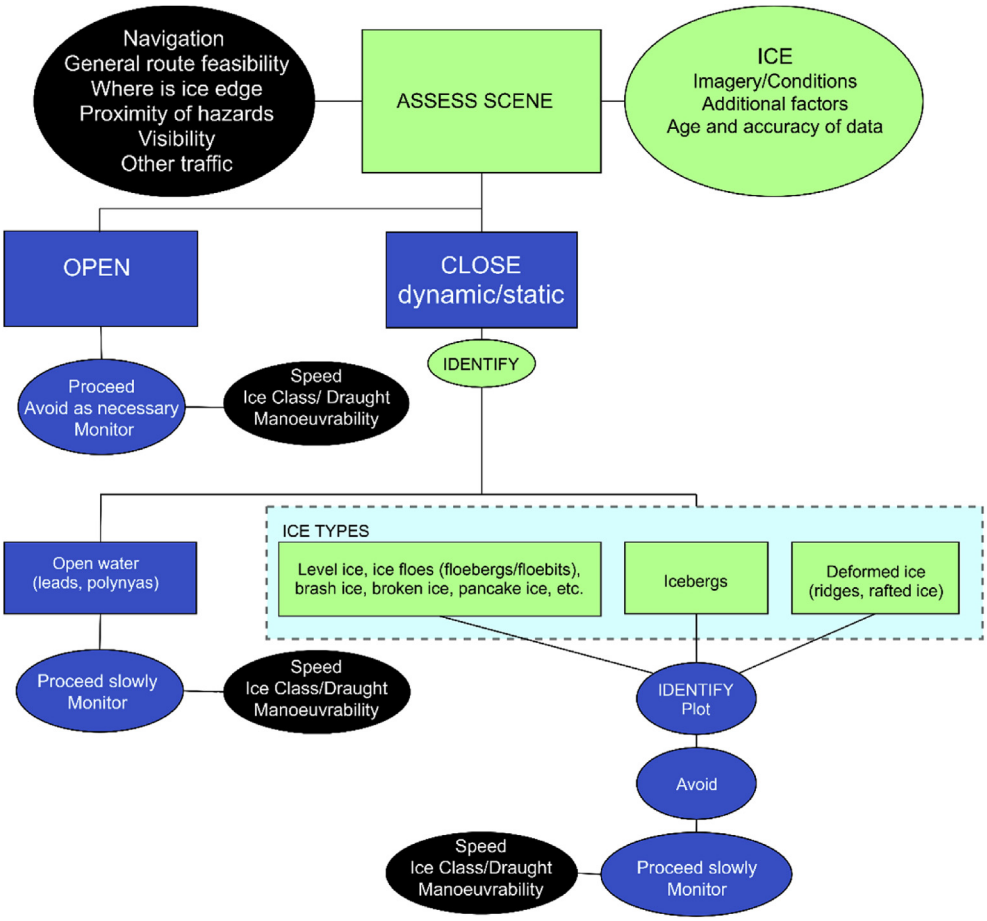
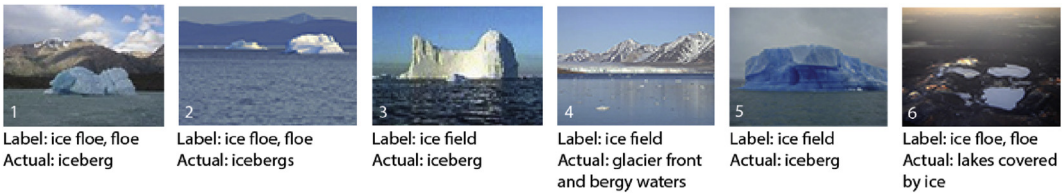


Fig. 1. An example of a human decision process when transiting in ice, adapted from Snider [1]; p. 88–89 with some modifications to highlight (in green) the importance of ice identification when transiting in ice.



Thumbnail ID	URLs of images
1	http://farm1.static.flickr.com/104/309452352_e224fb93ea.jpg
2	http://farm1.static.flickr.com/48/130208643_007a87e87b.jpg
3	http://www.masmar.net/var/masmar/storage/images/media/images/iceberg2001/12675-1-esl-ES/iceberg2001.jpg
4	http://www.digitalnature.org/landschappen/gletsjer%203%20kl.jpg
5	http://farm1.static.flickr.com/220/488207718_ecd382c79e.jpg
6	http://farm1.static.flickr.com/211/474340420_8c983c9b5a.jpg

Fig. 2. Examples of typical labeling inconsistencies from ImageNet [26]; what shown is thumbnails with corresponding URLs of original images.

as the understanding of severity of ice conditions is essential for a successful ice navigation. The surface ice features need to be correctly detected, classified, and accurately segmented, and the path needs to be carefully planned to avoid excessive ice loads on the ship hull or the ship besetting in ice.

Although considerable research has been devoted to autonomous shipping technology for aspects such as navigation support, equipment monitoring, and safety improvements, less attention has been paid to computer-aided scene assessment from ships that are travelling in ice. Examples of relevant works are given in Muramoto et al. [4]; Hall et al. [5]; Lu and Li [6]; Ji et al. [7]; Zhang and Skjetne [8]; Lu et al. [9]; Heyn et al. [10]; and Kjerstad et al. [11]. These papers describe techniques and sensor systems for analysis of ice concentrations, floe-size distributions, and drift speeds using shipborne images from optical cameras and/or marine radars. Despite recent progress in machine learning, the methods and standards for processing and analyzing sea ice imagery remain underdeveloped. There is a need for robust and efficient methods enabling the automated classification of sea ice imagery to aid in the derivation of useful characteristics of sea ice cover, and efforts have been made to address this. While most of the methods use low-to high-resolution satellite imagery or airborne imagery [8,12–16], none of these techniques capitalize on deep learning architecture for optical imagery classification

that is essential for navigation in ice. To the best of our knowledge, none of the currently available models can discriminate between ice types using optical images from a surface vessel. This issue must be addressed to enable automated image processing and the use of close-range imagery to support ice navigation tasks (i.e., interpretation of ice conditions and finding weak ice).

Motivated by the progress in deep-learning-based algorithms, the aim of this work is to study whether the state-of-the-art machine learning models can correctly identify ice surface features from close-range imagery (i.e., optical images predominantly taken from onboard the vessels).

This paper lays foundation for the automated identification of ice for surface vessels, and it provides a model for multilabel ice object classification that builds on state-of-the-art open source libraries and machine learning platforms. It then demonstrates the ability of this model to classify ice surface features from close-range optical imagery gathered online and during a research cruise to the Fram Strait on the RV Lance in March 2012. Different algorithms were tested on the collected ice imagery. The results were compared against randomly selected test cases representing different ice classes with varying degrees of distortion. In doing so, the effectiveness of the classification of different classes is evaluated and different levels of information presented for the classification are compared.

Background

Floating ice covers are complex and a very mixed medium. An increasing number of studies reports similarities in the way large neural networks and a human visual system process the objects Geirhos et al. [17]. provides an overview on this. This suggests that large neural networks may as well model human visual recognition of ice conditions. Currently, no universally accepted procedure exists for the automated interpretation of ice conditions using close-range ice imagery. Convolutional neural networks (CNNs), introduced by LeCun et al. [18]; have demonstrated an excellent performance in tasks such as image analysis, speech recognition, document analysis, and spam detection; see, e.g., Krizhevsky et al. (2012a,b [19,20]; Abdel-Hamid et al. [21]; and Ciresan et al. [22]. In recent years, several papers have shown that CNNs can also achieve an impressive performance on more challenging tasks, such as in the colorization of black and white images [23], remote sensing [24], and medical diagnostics (e.g., refer to an overview in Ravi et al. [25].

Despite this growing number of successful applications, the classification of ice objects from close-range imagery remains challenging. Available datasets from academic and Kaggle communities are limited to a few ice categories (e.g., icebergs, growlers, ice floes, and ice fields) and have many incorrectly labeled images, especially in the categories of ice fields, ice floes, and growlers (see Fig. 2 for a few examples).

In this paper, we classify ice surface features into nine categories: *icebergs*, *floebergs*, *floebits*, *floes*, *broken ice*, *level ice*, *brash ice*, *pancake ice*, and *deformed ice* (see Table 1). However, the presented model is suitable for classifying any number of categories and may be adopted for imagery from other sources, provided that the availability of close-range ice imagery continues to increase.

The following section introduces the deep machine learning method used in this study.

Method

The model builds on the *fastai* [27] version 1.0.57 and *PyTorch* [28] version 1.1.0 libraries. Image processing tasks are high-dimensional problems that require many matrix operations. Almost all of the computations in this study were performed on an NVIDIA TITAN X GPU (12GB) located at UNINETT Sigma2 AS in Norway.

It has been experimentally shown [29,30] that models trained on large datasets produce representations that are transferable to other

Table 1
Ice object categories (the numbers in brackets are the number of objects in each category).

Ice Object (ImageNet/this study)	ImageNet	WMO Sea Ice Nomenclature	This study
Growler (129/none)	A small iceberg or ice floe just large enough to be hazardous for shipping	Piece of ice [glacier origin] floating less than 1 m above the sea surface. A growler generally appears white but is sometimes transparent or blue-green in color. Extending less than 1 m above the sea surface and normally occupying an area of approximately 20 m^2, growlers are difficult to distinguish when surrounded by sea ice or in high sea states.	–
Iceberg, berg (1050/51)	A large mass of ice floating at sea, usually broken-off of a polar glacier	A massive piece of ice of greatly varying shape, protruding more than 5 m above sea level, that has broken away from a glacier and may be afloat or aground.	A piece of ice of glacier origin, floating at sea
Ice field (75/none)	A large flat mass of ice (larger than an ice floe) floating at sea	Area of floating ice, which is greater than 10 km across, consisting of floes of any size.	–
Ice floe, floe (655/15)	Flat mass of ice (smaller than an ice field) floating at sea	Any contiguous piece of sea ice.	From the WMO
Broken ice (none/135)	–	–	Predominantly flat ice cover broken by gravity waves or due to melting decay
Pancake ice (none/28)	–	Predominantly circular pieces of ice from 30 cm to 3 m in diameter, and up to approximately 10 cm in thickness, with raised rims due to the pieces striking against one another.	From the WMO
Brash ice (none/100)	–	Accumulations of floating ice made up of fragments not more than 2 m across, the wreckage of other forms of ice.	From the WMO
Floebit (none/26)	–	A relatively small piece of sea ice, normally not more than 10 m across composed of a hummock (or more than one hummock) or part of a ridge (or more than one ridge) frozen together and separated from any surroundings. It typically protrudes up to 2 m above sea level.	From the WMO
Floeberg (none/24)	–	A large piece of sea ice composed of a hummock, or a group of hummocks frozen together, and separated from any ice surroundings. It typically protrudes up to 5 m above sea level.	From the WMO
Deformed ice (ice ridges, rubble, hummocks) (none/153)	–	A general term for ice that has been squeezed together and, in places, forced upwards (and downwards). Subdivisions are rafted ice, ridged ice and hummocked ice.	From the WMO
Level ice (none/86)	–	Sea ice that has not been affected by deformation.	From the WMO

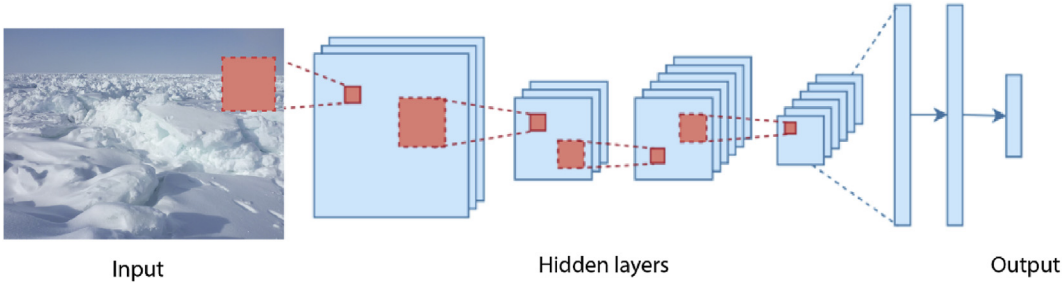


Fig. 3. Example of a CNN model (adopted from a figure from www.towardsdatascience.com).

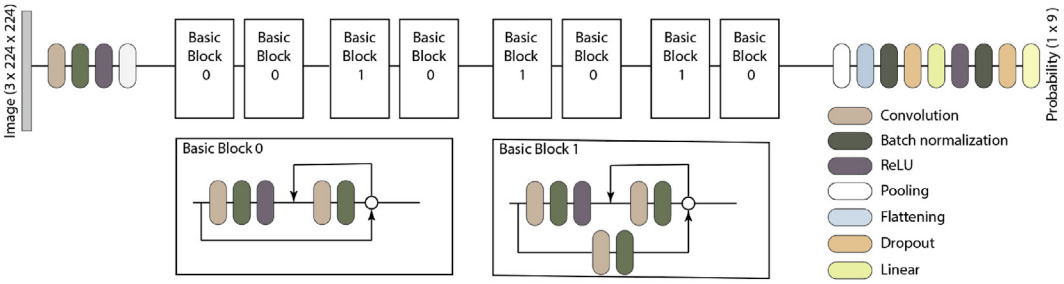


Fig. 4. Architecture of the ResNet18 model.

Fig. 5. Typical ice images in the dataset with labels.

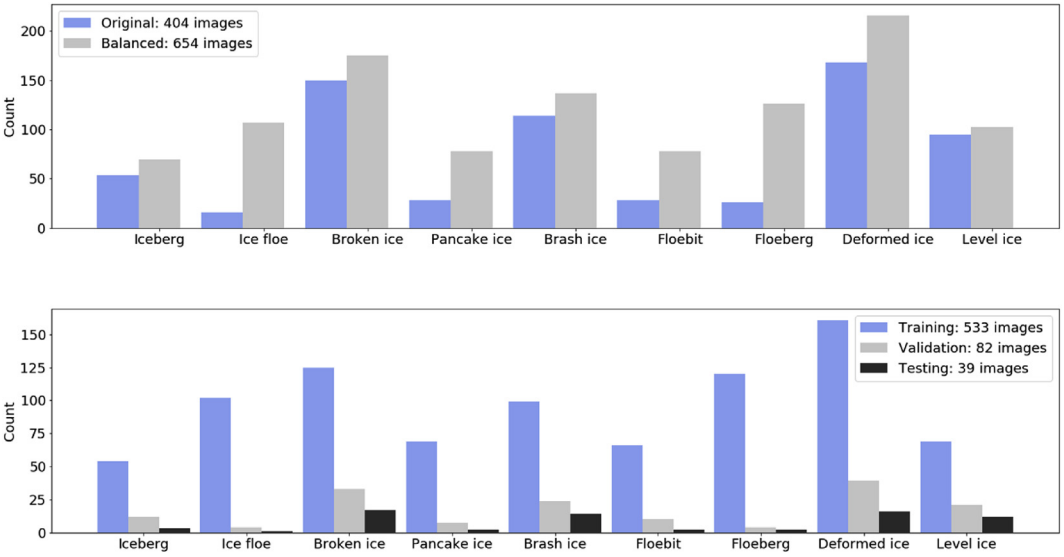


Fig. 6. Object distribution per class.

Table 2
Overview of the model hyperparameters.

Hyperparameter	Value
Weight decay	0.2 (top layer), 0.1 (whole model)
Number of epoch	12 (top layer), 8.0 (whole model)
Dropout rate	0.1, 0.2 and 0.5
Batch size	32
Learning rate	10^{-2} (top layer), [10^{-6} 10^{-3}] (whole model)
Image size	224 and 299

datasets. We start with a pertained residual neural network model (ResNet18) that is able to recognize thousands of categories of objects. This model has already been trained by analyzing approximately 1.5 million images (ImageNet), including various images containing ice; thus, this ResNet18 model 'knows' what ice looks like and can work with the little data we had available for training. The model takes ice images as input and predicts the probability of each of the nine categories listed in Table 1.

Basics

A typical CNN model has input-, hidden-, and output layers (Fig. 3). The *input layer* (in our case, the input data are optical images of ice conditions) provides information from the outside world to the network. The *hidden layers* perform computations and transfer information from the input layer to the output layers. The *output layers* are part of a densely connected network that takes input from the CNN network, which is a learned representation of the input images, and produces probabilities of the abovementioned classes.

The information in the network can be fed in the forward direction (from the input, through the hidden layers, to output) or there can be feedback loops between the layers (Fig. 4). The latter is used in our case. Once the forward pass is completed, the loss is calculated based on the cross-entropy function, and the model weights are updated to make the prediction better for the next pass. This forward/-backward pass is performed over many epochs to find the minimum loss point.

Given a set of features in **x**, and a desired output **y**, the model can learn the relationship between the feature and the target by minimizing the error between the model prediction and the desired output.

Model architecture

The architecture of the ResNet18 model is adopted from the PyTorch library and schematically shown in Fig. 4. The residual neural network (or ResNet) architecture was originally proposed by He et al. [31] to address the training accuracy issues that arose upon increasing the depth of deep neural networks. Since then, ResNets have been one of the most stable methods for training large CNNs.

There are three basic components in the architecture: (1) convolution layers, (2) pooling layers, and (3) fully connected layers. The following paragraphs briefly summarize these concepts. For more details refer to Goodfellow et al. [32].

In the first layer, the convolution is applied to the input data (**x**) using a convolution filter/kernel (**w**) to produce a feature map (***s***). For each pixel location (*i*, *j*), new pixel values are determined according to the following formula:

$$s_{i,j} = \sum_{k=1}^{m} \sum_{l=1}^{m} w_{k,l} x_{i+k-1,j+l-1}, \quad (1)$$

where *m* is the kernel width and height.

Typically, a convolution layer consists of several kernels, and multiple convolutions on the input are performed (for details on ResNets, see He et al. [31]). Each convolution operation uses a different kernel and results in a distinct feature map. The obtained feature maps are then stacked

Table 3
Accuracy (%) and the F-beta score in the validation dataset.

Model architecture	Accuracy %	F-beta	Misclassified	Reference
ResNet18	90	0.70	12 out of 82	He et al. [31]
ResNet34	91	0.72	6 out of 82	He et al. [31]
ResNet50	92	0.78	8 out of 82	He et al. [31]
SE_ResNet50	91	0.68	9 out of 82	Hu et al. [41]
Xception-Cadene	91	0.71	9 out of 82	Chollet [42]
Inception-v4	90	0.68	11 out of 82	Szegedy et al. [43]
Inception-ResNet-v2	84	0.71	17 out of 82	Szegedy et al. [43]

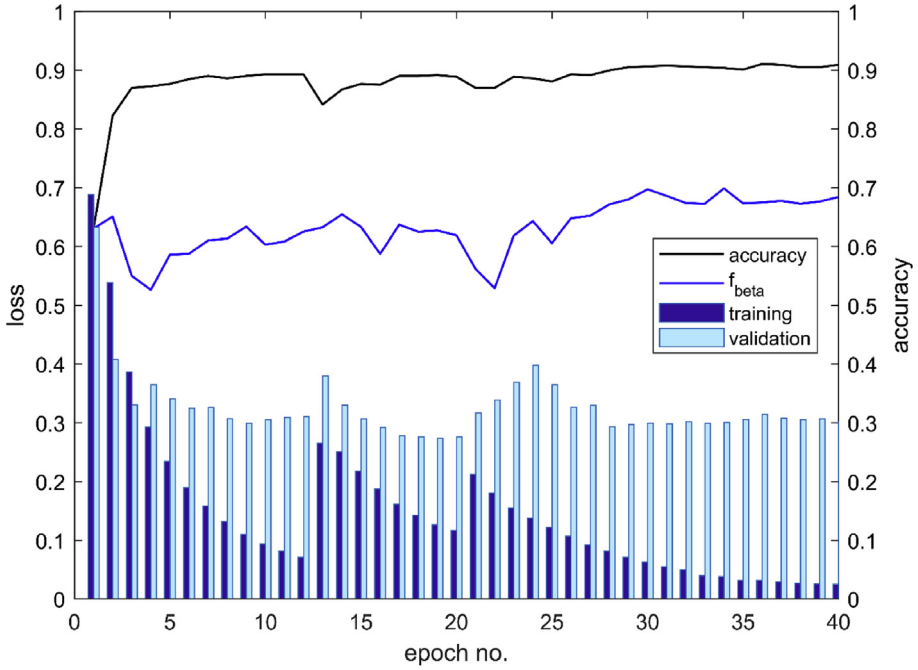


Fig. 7. ResNet18 model performance on the validation and training datasets.

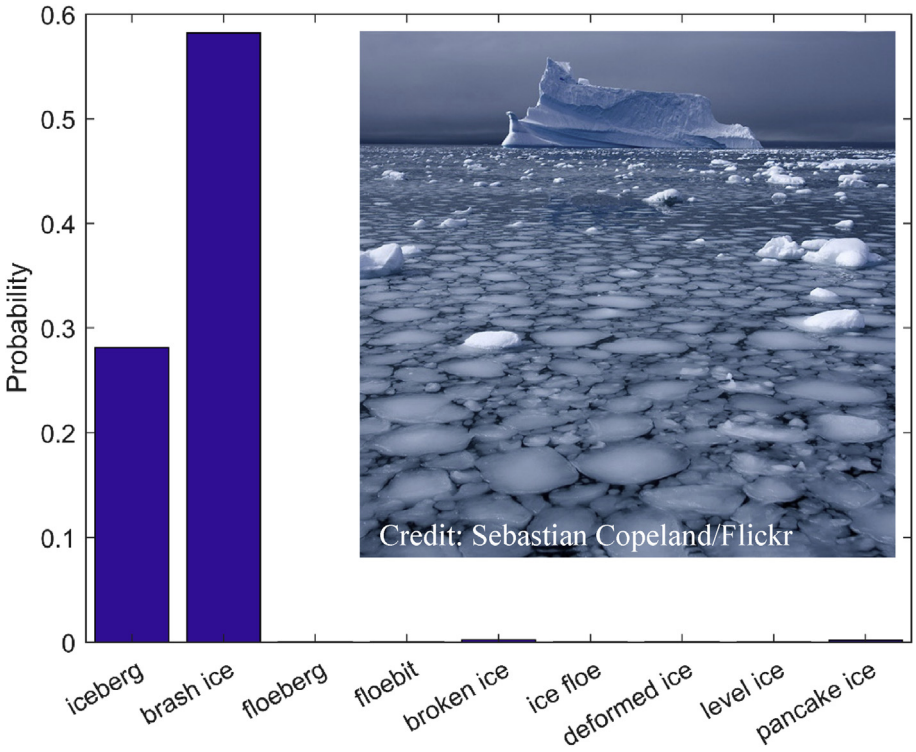


Fig. 8. Validation results (probability output), image source https://www.flickr.com/photos/globalgreenusa/2613534477.

Table 4
Confusion matrix for ResNet50: discrimination threshold 0.5 (TN – true negatives, FP – false positives, FN – false negatives, TP – true positives).

TN	FP
FN	TP

Example

69	1
6	6

Iceberg

56	2
6	18

Brash ice

78	0
3	1

Floeberg

72	0
7	3

Floebit

45	4
3	30

Broken ice

78	0
2	2

Ice floe

39	4
4	35

Deformed ice

57	4
8	13

Level ice

74	1
3	4

Pancake ice

Table 5
Confusion matrix for ResNet50: discrimination threshold 0.9 (TN – true negatives, FP – false positives, FN – false negatives, TP – true positives).

TN	FP
FN	TP

Example

70	0
7	5

Iceberg

58	0
14	10

Brash ice

78	0
4	0

Floeberg

72	0
9	1

Floebit

48	1
11	22

Broken ice

78	0
3	1

Ice floe

41	2
11	28

Deformed ice

60	1
14	7

Level ice

75	0
4	3

Pancake ice

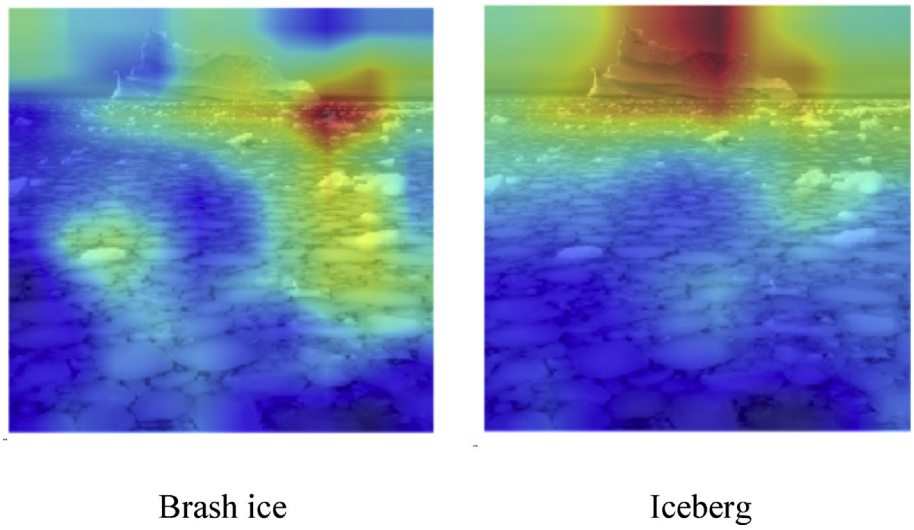


Fig. 9. Original image and the visual explanations generated by Grad-CAM on a given image. The heatmaps highlight the important regions of the image for predicting different ice objects: from the most important (red) to the least important (blue). These explanations are for decisions made by the ResNet50 architecture.

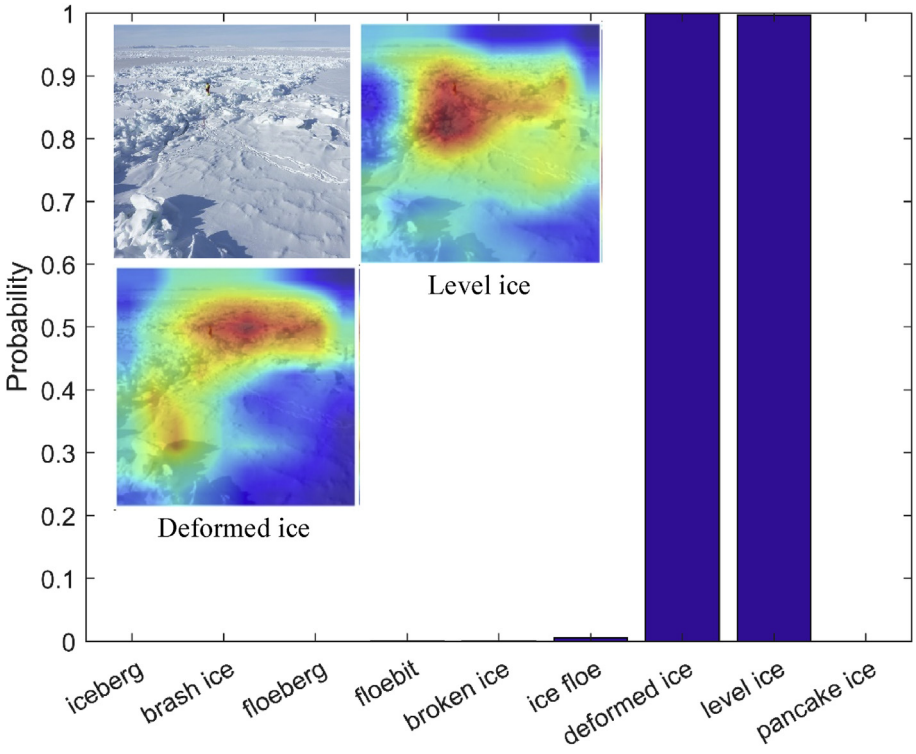


Fig. 10. Original image, the probability output and the visual explanation for the decisions. The heatmaps highlight the important regions of the image for predicting different ice objects: from the most important (red) to the least important (blue). These explanations are for decisions made by the ResNet50 architecture.

together to become the final output of the convolution layer. With more convolutional layers added, the model can learn features in a hierarchical manner (gradients, lines, edges, basic shapes, complex shapes, etc.) as shown in Zeiler and Fergus [33].

Another operation that is typically performed is a pooling operation, which maps a subregion to its maximum value (maximum pooling) or to its average value (average pooling). This operation makes the convolution output less sensitive to variance in the image (e.g., in object translations). Equation (2) is a mathematical expression of the maximum pooling operation:

$$s_{i,j} = max\left(x_{i+k-1,j+l-1}\right) \text{ for } 1 \le k \le m \text{ and } 1 \le l \le m. \quad (2)$$

Yet another important operation is using an activation function, such as the rectified linear unit (ReLU). ReLU adds nonlinearity to deep learning models. It takes an input value and replaces negative values with zeros. Mathematically, ReLU is expressed as $y = \max(0, x)$. By not applying this function, the image classification problem will be treated as linear. This layer helps reduce the number of parameters in the system.

The flatten layer converts a matrix into a column vector. To increase the computational efficiency and stabilize the network during training,

Table 6
Confusion matrix for ResNet50 with discrimination threshold of 0.9 (test set).

TN	FP
FN	TP

Example

36	0
3	0

Iceberg

25	0
5	9

Brash ice

37	0
2	0

Floeberg

37	0
2	0

Floebit

21	1
3	14

Broken ice

38	0
1	0

Ice floe

22	1
4	12

Deformed ice

26	1
9	3

Level ice

37	0
2	0

Pancake ice

Predicted: broken ice

Actual: deformed ice, level ice

Predicted: broken ice

Actual: level ice

Fig. 11. Examples of misclassified images (original image source, from left to right: https://www.flickr.com/photos/polarnix/5641749102/in/album-72157626422325235/, http://research.iarc.uaf.edu/~jenny/ShipObs2010/pobedy.php.

the inputs are normalized (see the batch normalization layer in Fig. 3).

The dropout operation proposed by Srivastava et al. [34] has been implemented in this study to achieve a better performance and generalization of the model. This operation enables models to avoid overfitting and to generalize the results better to images that were not used during training. The overfitting issue is discussed in Section 6.

In a very simplistic form, a residual neural network model can be mathematically expressed as

$$\mathbf{y} = f\left(h(\mathbf{x}) + g_p(\mathbf{x})\right) \tag{3}$$

where ($\mathbf{x}$, $\mathbf{y}$) is the input-output data, p are the tunable parameters of the function g_p, and f and h are fixed functions. The input data in our case are an image represented as a three-dimensional array containing the image's pixels and color channels. The output is the probability that this image contains ice objects belonging to a certain category listed in Table 1. The procedure for finding unknown parameters of the functions in Eq. (3) is an optimization process. In this work, a stochastic gradient descent-based method (AdamW) with deployed backpropagation is used. Details of this method can be found in the work by Loshchilov and Hutter [35]. The original parameters of the network along and the parameters of the batch normalization layer, are learned during training. The objective of the method is to minimize the cross-entropy loss function over multiclass classifications.

We implement the model using the standard *PyTorch* and *fastai* libraries, and reference is made to the source code files of ICEXPERT. Next, we train our model using the ice object dataset and use the accuracy of the model to evaluate the performance. The dataset contains close-range photographs of ice cover, where the objective is to recognize ice objects in these images.

Data

The training, validation, and testing datasets, 404 unique images in total, consist of the following:

a) Ice imagery from Internet collected by querying the Google, Yandex, Baidu search engine in different languages.
b) Imagery gathered during the research cruise to the Fram Strait on the RV Lance in 2012.

All the images were labeled manually in accordance with the definitions provided in Table 1. We then asked an ice expert to verify whether each image contains ice objects in the label list. Fig. 5 shows examples of images in the dataset. Several ice images have multiple labels (a varying number per sample).

The object distribution per class in the training, validation, and testing sets are shown in Fig. 6. The distribution of the number of categories in each image is as follows: 171 images have only one category, 193 images have two categories, 39 images have three categories, and one image has four categories.

The original dataset (404 images) is imbalanced, as there is an uneven distribution of ice classes within the dataset (Fig. 6, top). The ratio of "ice floe" to "deformed ice" is approximately 1:10, and thus the class "ice floe" is under-represented as well as the "floebit", "floeberg", "pancake ice", and "iceberg" classes. To improve this uneven class distribution, we used random minority oversampling method proposed by Buda et al. [36]. We replicated randomly-selected samples from the minority classes and added those to the original dataset resulting in a total of 654 images. It has been shown [37,38] that this method is effective, but can lead to overfitting.

Training details

The classifier was trained on the balanced dataset comprising 533 images (see Section Data), which were labeled in accordance with the object definitions provided in Table 1.

Our aim was to train a CNN model to correctly identify ice objects in a given image. Our strategy was to use transfer learning during which we initially kept all the convolution layers with their weights, which were pretrained on ImageNet, and define only custom set of fully connected linear layers on top with weights randomly initialized using the Kaiming initialization method [39].

We trained the model in two phases. First, we trained only the linear layers on the top for our ice data, whereas the rest of the model had no change in the weights (as we froze the weights of the bottom layers). Then, we unfroze the remaining layers of the model and fine-tuned the whole model using differential learning rates [40]. The best weights from the first phase were chosen as the starting point for the training in the second phase. The optimum learning rate was determined by finding the value in which the learning rate was highest, and the loss was still descending. For the upper layer, it was set to 10^{-2}, whereas for the whole model it ranged between 10^{-6} and 10^{-3}. As the bottom layers have their pre-learned weights, we wanted to fine-tune only them in our task, and, hence the learning rate for the lower layers was a few magnitudes smaller than the learning rate for the upper layers.

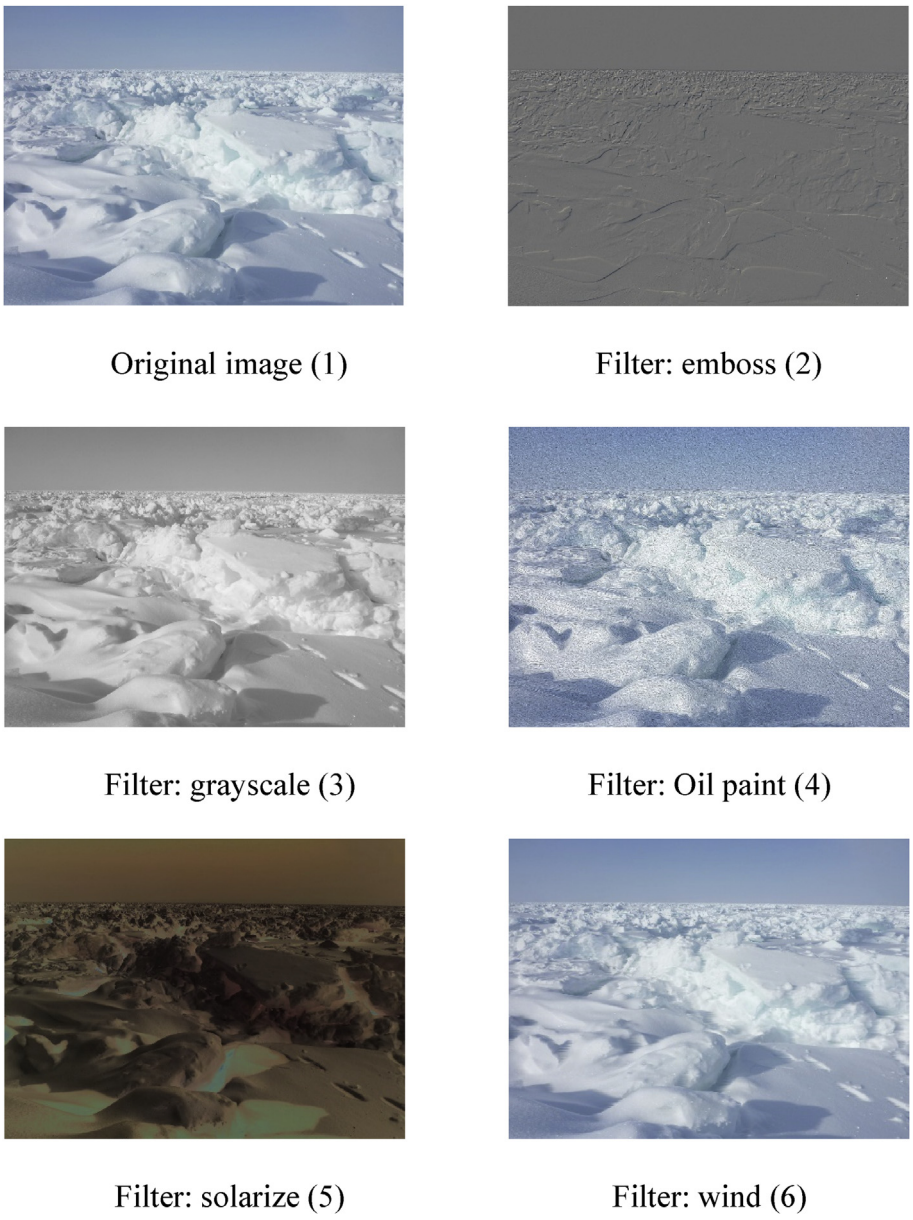


Fig. 12. Visualization of the synthetically modified test data by applying Photoshop filters to an original test image of deformed ice. After applying the filters, the global shape tends to be retained.

To handle a small amount of training samples, we exposed the model to more aspects of the data. More training data were generated from the available training samples using data augmentation. For each image in the dataset, random geometric and color augmentation were performed, including zooming, cropping, horizontal flipping, rotation, lightening and contrast changes, and symmetric warping. This data augmentation increases the breadth of information the model can analyze to learn a given dataset. The dataset was better suited to recognize target objects in images of varied contrast, size, angles, and so on.

In addition, we adopted an adaptive synthetic sampling approach for imbalanced learning by artificially generating additional data for categories with few images, such as pancake ice, ice floe, floebergs, floebits, and icebergs. We also exposed the model to different image sizes, which enabled the model to generalize better and handle the small dataset size.

We stopped training after 40 epochs on two different image sizes (20 epoch per size), which took approximately 30 min on a single NVIDIA GPU (12 GB). The learning rate was manually annealed throughout the training when the validation error plateaued. The weight decay and dropout rates were adjusted to improve the model performance and generalization. Table 2 presents a summary of the model hyperparameters.

To evaluate the performance of the model and to compare the quality of predictions with different models, two metrics were used: the F-beta score and the accuracy. F-beta is the harmonic mean of the precision and recall weighted towards the recall ($\beta = 2$, library default). F-beta is defined as follows:

$$F_\beta = \left(1+\beta^2\right) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}\ , \tag{4}$$

where *precision* is the fraction of relevant instances (true positives) among the retrieved instances (true positives + false positives) and *recall* is the fraction of relevant instances (true positives) that are retrieved over the total number of relevant instances (the number of true positives in the data). β is the weight given to the recall over the precision, where $\beta > 1$ favors the recall over the precision. The F-beta score lies in the range

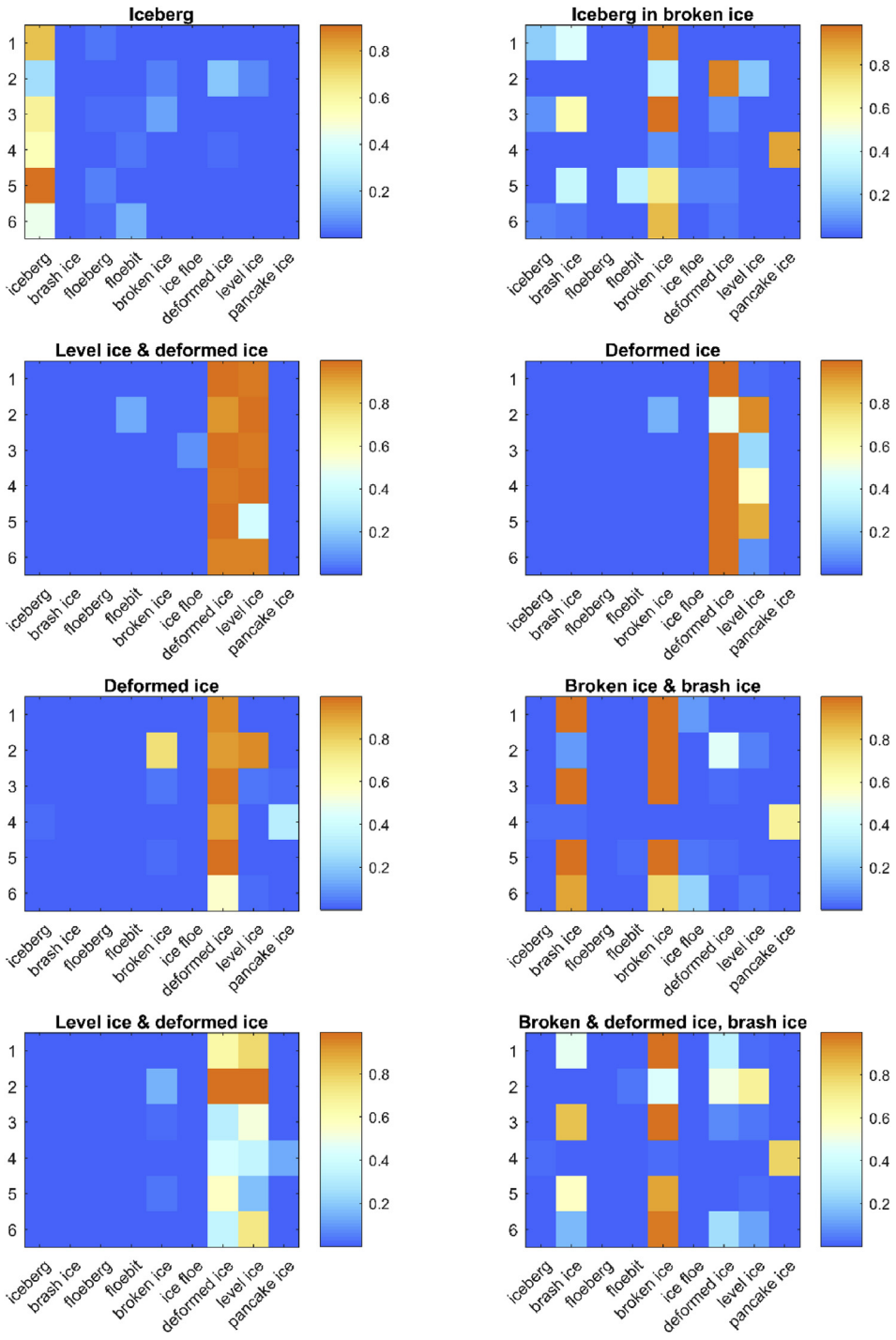


Fig. 13. Effect of object distortion on eight test images for ResNet50.

[0,1], with 1.0 being ideal (when both the precision and recall are equal to 1.0) and 0.0 being unsatisfactory.

The accuracy is the percentage of ice objects identified correctly. Both the accuracy and the F-beta score are used together with a classification threshold of 0.5 (library default). The latter means that if the probability that an image contains an ice object belonging to a certain category is greater than 0.5, we make the decision that this image contains this ice object.

Experiments and discussion

First, we analyze the performance of the classifier during training and validation. The training and validation loss as well as the accuracy for each iteration (epoch) were calculated and plotted (Fig. 6). We present the accuracy of the classification, as defined by the accuracy threshold criterion [27]. The F-beta score is calculated as in Eq. (4). Fig. 6 shows the results obtained by training with the ResNet18 model.

The goal of a machine learning model is to generalize from the training data to any seen or unseen data from the problem domain, which will allow predictions on data that the model never has seen. The behavior of the loss diagrams for the training and validation datasets (Fig. 7) indicates some degree of overfitting with the model, which means that ResNet18 models the training data too well by learning the detail (and noise) in the training data to the extent that it negatively impacts the performance of the model on the validation dataset.

Furthermore, the plots of the accuracy and F-beta scores in Fig. 7 show the importance of looking at the images at different resolutions. A slight additional increase in the F-beta score is obtained when using higher resolution images, which is seen in the scores after epoch 25.

Effect of architectural changes

Furthermore, the effect of the architectural changes on the performance of the model is analyzed. Images were classified using the following architectures: ResNet18, ResNet34, ResNet50, SE-ResNet50, Xception-Cadene, Inception-v4, and Inception-ResNet-v2. The experimental setup is the same as that described in Section Training details, with the difference being only the CNN architecture used. Table 3 presents the results (the accuracy, the F-beta score, and the number of misclassifications).

Comparing the numbers in Table 3, it is evident that the ResNets models performed slightly better than the other models in classifying ice images.

An analysis of the misclassified images showed that most of the models in Table 3 have some difficulties discriminating pancake ice and brash ice. The latter could be due to a small size of the dataset. This could also indicate that the high-order feature interactions, which are specific to ice, were not entirely captured with the studied models. Most of the models failed to discriminate between an iceberg, a floebit, and a floeberg. On a few occasions, broken and deformed ice was classified only as the broken ice.

The decision was to work further with the ResNet50 giving the highest performance metrics. Examples of detections and precision-recall matrices (also known as confusion matrices) for each object class are shown in Figs. 8–10 (detections) and Tables 4 and 5 (precision-recall matrices for discrimination thresholds of 0.5 and 0.9). Each row of the matrix represents the instances in the actual class (ice class, non-ice class) while each column represents the instances in the predicted class (predicted ice class, predicted non-ice class). The rows and columns in Tables 4 and 5 report the number of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN). For example, the number of real "brash ice" instances in the validation dataset is 24, while the rest of the images (58) do not contain any brash ice. The model was able to recognize brash ice in 18 instances (TP = 18) with 6 misclassifications (FN = 6) and negatively predicted brash ice with correct rejection in 56 instances (TN = 56) and 2 false detections (FP = 2).

In addition, Figs. 9 and 10 present additional results on the visual explanations for typical images containing brash ice, level ice, deformed ice, and an iceberg. The heatmaps in Figs. 9 and 10 were produced using the Grad-CAM technique [44] and show which parts of an input image that were looked at by the CNN for assigning a label for the predicted categories. Note that this technique does not necessarily capture the entire object but serves as a visual explanation of the CNN predictions.

The analysis of the predictions in Figs. 8 and 9, and the scores in Tables 4 and 5 indicate that the ResNet50 model performs poorer in predicting icebergs than broken ice, or deformed ice. Despite a seemingly correct activation (Fig. 9, iceberg instance), the predicted probability is below 50%. One possible explanation for this model behavior is the small size of the training dataset for the 'iceberg' class (54), compared with broken ice (125) and deformed ice (161).

Overall, the numbers in Tables 4 and 5 indicate a correlation between the model performance and the number of images in each ice class. ResNet50 performed better in classifying broken ice, deformed ice, and brash ice, whereas the performance for minority classes (floebit, floeberg, iceberg, ice floe, and pancake ice) is lower. We expect that the classification ability of the model will improve when supplied with more training data.

Testing

Herein we present further experimental results to confirm the effectiveness of the ResNet50 model. The performance of the model is estimated on the randomly selected test set described in Section 4. The achieved accuracy is 92% with 6 misclassified samples out of 39 images in the test dataset. Among the misclassified samples was an image containing a part of a foreign object (a vessel bow). The F-beta score was 0.76, and the confusion matrix for each object class is shown in Table 6.

Examples of misclassified images are presented in Fig. 11. The model was not able to correctly distinguish between the level ice and the broken ice when melting ponds were present on the former (see the example image in Fig. 11). This result is not surprising since the dataset was limited to a few images of decayed level ice. A careful examination of the image suggests uncertainty that the ice cover is a continuous ice sheet (level ice). If a pond melts through the entire thickness of the ice, the color of the pond turns dark. The color of the melt ponds at the horizon is slightly darker than that of the melt ponds in the front part of the image; therefore, there is no clear visual confirmation that this image is of a continuous ice sheet.

Similar to the performance in the validation set, the model experienced difficulties in detecting pancake and brash ice. Furthermore, in few instances, level ice and deformed ice was classified as the broken ice and vice versa.

Effect of object distortion

Recent experiments by Geirhos et al. [45] showed that object recognition with ResNets trained on ImageNet can be biased towards recognizing textures rather than shapes. To test the robustness of the developed algorithm for ice objects (presumably one 'ice' texture), we created an additional set of test data. Eight representative images from the original test dataset were converted in Photoshop (version 2017.1.1) using the following filters: grayscale, oil paint, solarize, wind, and emboss (resulting images samples are shown in Fig. 12). After applying the filters, the local textures and/or color were removed while maintaining the global shape of the ice objects.

The classification results with the trained ResNet50 are shown in Fig. 13. The vertical axis is the filter number, where 1 refers to the original image. The horizontal axis is the ice object categories. The prediction probabilities over the ice classes are color-coded (red represents 1.0, and blue represents 0.0). Despite being trained on the color images, the model is capable of correctly classifying ice objects from grayscale images (filter 3); however, predictions over the classes are nosier.

The numerical results indicate that the trained ResNet50 model is biased towards texture. The sensitivity maps in Fig. 13 show an adverse effect on the model predictions, specifically with filters numbers 2, 4 and occasionally 5 or 6. These filters introduce severe stylization of the image removing the local ice texture.

Conclusions and remarks

Conclusions

This work lays the foundation for the automated identification of ice objects for surface vessels using convolutional neural networks. In this study, we have explored the generalization ability of deep learning models to differentiate between nine categories of surface ice features: level ice, deformed ice, broken ice, icebergs, floebergs, floebits, ice floes, pancake ice, and brash ice. Open source frameworks have been used to develop our models, and we used pretrained models from ImageNet to overcome the lack of a large dataset. The conclusions are as follows:

- The paper presents a model for multilabel ice object classification that builds on state-of-the-art open source libraries and machine learning platforms.
- Using numerical experiments, we showed that the ResNet model does not require much data to achieve good performance metrics (an accuracy of approximately 90% and an F-beta score of 0.7); however,

the performance of the model for the minority classes is lower than that for majority classes and limited to clear photographs.
- For the images with distorted ice textures, the performance is also lower than that for the other images. Overall, the performance is expected to improve when supplied with more training data and a more balanced dataset.

Optical ice images collected at night, during heavy snow, or under poor visibility conditions (i.e., images with naturally distorted textures) are scarce, and future efforts should be aimed at collecting such data and analyzing it.

Remarks

This work presents a model for multilabel ice object classification that builds on state-of-the-art open source libraries and machine learning platforms. Similar methods have been used earlier on other image types, but never for the multilabel classification of sea ice objects from close-range optical images. The paper demonstrates the ability of the model to classify ice surface features from close-range optical imagery gathered online, and during the research cruise to the Fram Strait in 2012. A variety of CNN model architectures has been tested on the collected ice imagery. The results for the best-performing model were compared against the test cases representing different ice classes with varying degrees of distortion. We evaluated the effectiveness of the classification of different classes and compared different levels of information presented for the classification.

In addition to reporting the classification results, we presented detailed experiments that provide the following new insights:

- ResNet50 outperformed the other tested models.
- Despite little data (404 uniqie images for trainig, validation, and testing), the trained CNN model demonstrated a good performance on a randomly selected test dataset (39 images), achieving a 92% accuracy and an F-beta of 0.76 with ResNet50*.
- The most challenging tasks were to detect and/or to distinguish between pancake ice and brash ice, as well as to distinguish between broken ice and partially melted level ice.
- The model is capable of classifying ice objects from greyscale images with poorer performance on significantly distorted images with a small degrees of local ice texture.

In the future, one may test the model performance on low-visibility images (fog, darkness, during snow, etc.), to check whether the model performance will converge as more training samples become available, as well as to enhance the model to extract measurable parameters (ice/ridge concentration, degree of ice decay, etc.) from the ice images.

We hope that the presented findings can stimulate and support the development of ice navigation support systems. Moreover, the proposed model can be used on geotagged optical ice images to automatically map, e.g., icebergs and other ice features from surface vessels. This information can later be used as a ground truth for satellite data. The model is also useful for large-scale automated image processing, such as automatic annotation of images collected during research expeditions, as well as for education, for helping people to learn about different ice features beyond the icebergs and ice floes.

Conflict of interest

The author declare that there is no conflict of interest.

Acknowledgments

The authors would like to thank Professor Theoharis for the valuable discussions during this study. We would also like to acknowledge UNINETT Sigma2 AS, the national infrastructure for computational science in Norway, for granting access to their data storage and processing resources.

References

[1] D. Snider, Polar Ship Operations, The Nautical Institute, 2012, 136p.
[2] WMO Sea Ice Nomenclature, World Meteorological Organization, 2014.
[3] M.E. Johnston, G.W. Timco, Understanding and Identifying Old Ice in Summer, Canadian Hydraulics Centre, National Research Council, CHC-TR-055, Canada, 2008.
[4] K. Muramoto, K. Matsuura, T. Endoh, Measuring sea-ice concentration and floe-size distribution by image processing, Ann. Glaciol. 18 (33) (1993).
[5] R.J. Hall, N. Hughes, P. Wadhams, A systematic method of obtaining ice concentration measurements from ship-based observations, Cold Reg. Sci. Technol. 34 (2) (2002) 97–102.
[6] P. Lu, Z. Li, A method of obtaining ice concentration and floe size from shipboard oblique sea ice images, IEEE Trans. Geosci. Remote Sens. 48 (2010) 2771–2780.
[7] S. Ji, H. Li, A. Wang, Q. Yue, Digital Image Techniques of Sea Ice Field Observation in the Bohai Sea, Proceedings of POAC11-077, 2011.
[8] Q. Zhang, R. Skjetne, Image processing for identification of sea-ice floes and the floe size distributions, IEEE Trans. Geosci. Remote Sens. 53 (5) (2015) 2913–2924.
[9] W. Lu, Q. Zhang, R. Lubbad, S. Løset, R. Skjetne, A shipborne measurement system to acquire sea ice thickness and concentration at engineering scale, in: Proceedings of Offshore Technology Conference, 2016, https://doi.org/10.4043/27361-MS.
[10] H.-M. Heyn, M. Knoche, Q. Zhang, R. Skjetne, A system for automated vision-based sea-ice concentration detection and floe-size distribution indication from an icebreaker, in: Proceedings of the International Conference on Ocean, Offshore, and Arctic Engineering, 2017.
[11] Ø.K. Kjerstad, S. Løset, R. Skjetne, R.A. Skarbø, An ice-drift estimation algorithm using radar and ship motion measurements, IEEE Trans. Geosci. Remote Sens. 56 (6) (2018) 3007–3019.
[12] R. Kwok, Declassified high-resolution visible imagery for Arctic sea ice investigations: an overview, Remote Sens. Environ. 142 (2014) 44–56.
[13] X. Miao, H. Xie, S.F. Ackley, S. Zheng, Object-based arctic sea ice ridge detection from high-spatial-resolution imagery, IEEE Geosci. Remote Sens. Lett. 13 (2016) 787–791.
[14] N. Zakhvatkina, A. Korosov, S. Muckenhuber, S. Sandven, M. Babiker, Operational algorithm for ice–water classification on dual-polarized RADARSAT-2 images, The Cryosphere 11 (2017) 33–46.
[15] N.C. Wright, C.M. Polashenski, Open-source algorithm for detecting sea ice surface features in high-resolution optical imagery, The Cryosphere 12 (2018) 1307–1329.
[16] K. Duncan, S. Farrell, L. Connor, J. Richter-Menge, J. Hutchings, R. Dominguez, High-resolution airborne observations of sea-ice pressure ridge sail height, Ann. Glaciol. 59 (2018) 137–147, 76pt2.
[17] R. Geirhos, D.H.J. Janssen, H.H. Schütt, J. Rauber, M. Bethge, F.A. Wichmann, Comparing Deep Neural Networks against Humans: Object Recognition when the Signal Gets Weaker, 2017.
[18] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[19] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[20] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, 2012, pp. 1106–1114.
[21] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, Li Deng, G. Penn, D. Yu, Convolutional neural networks for speech recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 22 (10) (2014) 1533–1545.
[22] D.C. Ciresan, U. Meier, L.M. Gambardella, J. Schmidhuber, Convolutional neural network committees for handwritten character classification, in: Proceedings of the 2011 International Conference on Document Analysis and Recognition, IEEE Computer Society, Washington, USA, 2011, pp. 1135–1139.
[23] Y. Chen, Y. Luo, Y. Ding, B. Yu, Automatic colorization of images from Chinese black and white films based on CNN, in: International Conference on Audio, Language and Image Processing, Shanghai, 2018, pp. 97–102.
[24] Y. Zhong, F. Fei, L. Zhang, Large patch convolutional neural networks for the scene classification of high spatial resolution imagery, J. Appl. Remote Sens. 10 (2) (2016) 025006.
[25] D. Ravi, C.F. Wong, M. Deligianni, J. Berthelot, Andreu-Perez, B. Lo, G.-Z. Yang, Deep learning for health informatics, IEEE Journal of Biomedical Health Informatics 21 (2017) 4–21.
[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, 2009. CVPR09.
[27] J. Howard, R. Thomas, S. Gugger, Fastai, GitHub, 2018.
[28] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic Differentiation in PyTorch, 2017.
[29] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, IEEE CVPRW (2014) 512–519.
[30] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, A. Oliva, Learning deep features for scene recognition using places database, NIPS (News Physiol. Sci.) (2014) 487–495.
[31] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, 2016, pp. 770–778. https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.
[32] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[33] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, 2014, pp. 818–833.
[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[35] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, ICLR, 2019.
[36] M. Buda, A. Maki, M.A. Mazurowski, A systematic study of the class imbalance problem in convolutional neural networks, Neural Netw. 106 (2018) 249–259.
[37] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-smote: a new oversampling method in imbalanced data sets learning, Advances in Intelligent Computing (2005) 878–887.
[38] K.-J. Wang, B. Makond, K.-H. Chen, K.-M. Wang, A hybrid classifier combining smote with pso to estimate 5-year survivability of breast cancer patients, Appl. Soft Comput. 20 (2014) 15–24.
[39] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: surpassing human-level performance on ImageNet classification, in: ICCV, 2015, p. 2015.
[40] L.N. Smith, A Disciplined Approach to Neural Network Hyper-Parameters: Part 1 – Learning Rate, Batch Size, Momentum, and Weight Decay, US Naval Research Laboratory, 2018. Technical Report 5510-026.
[41] J. Hu, Li Shen, G. Sun, IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[42] F. Chollet, Xception: deep learning with depth wise separable convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 1800–1807.
[43] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, inception-resnet and the impact of residual connections on learning, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 4278–4284.
[44] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: visual explanations from deep networks via gradient-based localization, IEEE International Conference on Computer Vision (2017) 618–626.
[45] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F.A. Wichmann, W. Brendel, ImageNet-trained CNNs Are Biased towards Texture; Increasing Shape Bias Improves Accuracy and Robustness International Conference on Learning Representations, 2019.