Classifying icebergs and ships from Sentinel-I SAR imagery using convolutional neural networks

Aleksi Hämäläinen[1] and Mikko Kuusisto[2]

Abstract— **To decrease costs of maintaining safe working conditions on maritime operations, Statoil and C-CORE published a Kaggle challenge to developing a high-performance classification model that automatically identifies ships and icebergs from each other in the radar imagery. In total the data includes 1604 labeled 75x75 dual channel images captured by Sentinel-I - remote sensing system and its Synthetic Aperture Radar (SAR). The imagery includes both HH and HV polarization channels which makes it possible to more efficiently recognize objects with different bounceback polarization distributions. The provided data includes plenty of background noise as expected from a remote sensing system. Thus, several pre-processing methods and convolution deep learning models were compared with different hyperparameters. As the classification problem requires the model to detect small details from the image, small kernel size is required. Our results showed that deep models achieve better results with small kernel size. In addition, we found that with a small batch size, overfitting can be decreased. With data augmentation, we were able to increase labeled data set from 1600 to 1800. he validation loss decreased from 0.2614160 to 0.20011 with data augmentation.**

I. Introduction

In this article we will explain our approach in creating a model for the machine learning problem described on the online platform Kaggle which is meant for predictive modeling and analytics competitions. The challenge addresses the problem of classifying drifting icebergs from ships in arctic seas from radar data collected by Earth orbiting Sentinel satellites. Automatically identifying drifting icebergs from the satellite radar imagery could significantly decrease the threats that icebergs cause for the maritime traffic. Furthermore, the satellite monitoring data is the only viable solution to acquire reliable and constant data for remote areas with difficult weather conditions. The Kaggle challenge has been initiated by Statoil and C-CORE to developed more efficient algorithms that automatically identifies ships and icebergs from each others in the radar imagery. The ultimate goal is to decrease costs of maintaining safe working conditions on maritime operations. [1]

The available datasets include 75x75 images with two bands. There is a labeled training set and unlabeled test dataset to be used to evaluate the accuracy of different algorithms submitted for the competition. We focused entirely on the labeled training set and divided it into training and testing datasets to be used in cross-validation. The labels in the training set are provided by human experts and indicates whether the object in the picture is an iceberg or a ship. [1] Therefore, the goal is to create a binary classifier to predict the label of each object. We selected Convolutional Neural Networks as the basis of our model due to its good reputation in image classifying problems. Convolutional Neural Networks have been successfully applied to analyzing visual imagery based on their ability of recognizing the underlying patterns of an object through the convolutional layers. [2]

However, before training the model, several pre-processing methods were used. We standardized each image data point to have a zero-mean and unit variance as well as used batch standardization. These are common techniques to improve the training performance of the model by making it easier for the model to learn the relevant patterns in the data. [2], [3]

Furthermore, one of the challenges related to training the model was the small amount of labeled data (1604 unique two channel images with labels). We ended up using data augmentation to increase

[1] Aleksi Hämäläinen, 425287, Department of Computer Science, Aalto University Schoolf of Science

[2] Mikko Kuusisto, 345370, Department of Industrial Engineering and Management, Aalto University School of Science

the amount of data available for training. In this case, data augmentation means rotating, flipping and zooming the images in order to generate a larger dataset which enables us to train more complex models without over-fitting.

In the end, we ended up using three different network structures. In addition, we optimized the hyperparameters such as batch size, learning rate and amount of epochs to improve and fine-tune the model.

II. Related work

Convolutional neural networks are nowadays widely used especially for image recognition or object classification from images. In practice, the convolutional neural networks (CNN) means a family of feed-forward artificial neural networks that are partly inspired by the physiology of animal visual cortex. The CNN consists of several convolutional and pooling layers that create feature maps of the input data. Due to this structure CNN is efficient in recognizing patterns and objects in an image. [2], [3] The current solutions can achieve remarkably high accuracies and some networks are even beating humans in object recognition. [4]. However, a downside with CNN is that it needs plenty of training data in order to attain the best results. Thus, especially in the beginning many of the applications included objects of which it was easy to attain labeled image samples. [2]

The convolutional neural networks have also been used to classify SAR images. Xu and Scott [5] studied CNN based models in classifying sea ice and open water in similar dual-polarization SAR data as the Kaggle competition provided. However, as mentioned earlier, the problem has previously been in low amount of properly labeled satellite imagery data. However, another important challenge in interpreting the radar images is the high amount of various factors that impact the interactions between the signal, sea ice, water and other objects or substances in the ocean as well as air. Xu and Scott implemented a transfer learning method with convolutional neural networks using the same three dimensions as we have available in our data: HH-, HV-polarization data and incidence angle. They applied a softmax classifier and extracted features from AlexNet [6], a CNN based model which competed in ImageNet 2012 competition. Xu and Scott achieved with this model an overall accuracy of 92.36% for classifying between sea ice and open ocean [5]. This indicates that our approach of using CNN based classifiers should yield good results in binary object recognition.

III. Method

To gain experience with the newest technologies in the machine learning field, we chose to use Pytorch to implement our deep learning models. Pytorch is a relatively new deep learning framework that is becoming popular among researchers [7].

The goal is to find parameters of a model that minimize the Negative Log Loss Likelihood (NLLL) function:

$$\sum_{n}^{B} L(y,p) = \sum_{n}^{B} -(y_n \log(p(x_n)) + (1-y_n) log(1-p(x_n)), \tag{1}$$

where B represents a batch size, $y \in 0, 1$ represents class of an image where 0 means a ship and 1 means an iceberg. The used model produced an output p that represents estimated probability that the image is an iceberg. NLLL determines a direction in which to update parameters of the model.

The probability of an iceberg occurring in the image is achieved with the Softmax function:

$$P(x) = \frac{e^{f(x)_2}}{e^{f(x)_1} + e^{f(x)_2}}, \tag{2}$$

where x represents an input image and f represents the model, that returns a two dimensional vector. When the first value of the vector is higher than the second value, the model predicts that the image represents a ship. If the second value is higher than the first value, the model predicts an iceberg.

To build a model that achieves good results for our problem, we compared performances of three different CNN based deep learning models. In addition, three different learning rates and four different batch sizes were compared. The results were compared and cross-validated by splitting the labeled data into separate training and validation datasets.

Furthermore, we created more training data by using data augmentation. Modifications to be used in the augmentation can be for example image rotation, horizontal and vertical translation, and translation along positive diagonal. These modifications generate altered versions of the same data and, thus, generate more training samples for the model.

IV. Data

We acquired the data from Kaggle, a platform for predictive modeling and analytics competitions. The competition and data are provided by the Norwegian multinational oil and gas company Statoil in collaboration with Canadian research and development company C-Core. [1] The different fields in the dataset are displayed in the Table I below. In total the data includes 1604 labeled images. Furthermore, the Kaggle competition provides unlabeled test data which is used in creating predictions that are tested through the platform to create the leaderboard of all the teams attending the competition. However, in this article we will be only focusing on the labeled training data as it is split and used for both training and validation.

TABLE I: The format of the data fields and description

Variables	Description
id	The id of the image.
band_1	Both bands contain a flattened 75x75 pixel image meaning the list has 5625 elements. The pixel values are float numbers indicating the signal strength in dB. The band_1 is the HH imagery data.
band_2	Band_2 has the same format but includes the HV imagery.
inc_angle	The incidence angle of which the image was taken. This is the angle between the ocean surface and the bounced off signal. This feature has 133 missing values.
is_iceberg	The human experts have determined the label in the training data by hand indicating whether the image has an iceberg (value 1) or ship (value 0).

A. Overview of the Sentinel-I -remote sensing system

The Sentinel-I remote sensing system consists of two identical satellites orbiting at an altitude of 680 km on a polar orbit 180 apart from each other. Sentinel-I has a C-band Synthetic Aperture Radar (SAR) meaning that it can penetrate virtually any weather conditions and is not dependent on sunlight to capture imagery. Satellite radar works basically in the same way as any echo based remote sensor. It emits a signal and captures the signal echo that has bounced of from the object. Therefore, solid objects are recorded as bright spots as they reflect more of the signal back than the background. [8]

The radar polarization is also important aspect of the imagery. Sentinel-I can transmit and record signals on two polarization planes. In this machine learning challenge, the data includes two channels: HH and HV. The HH means that the channel is acquired by both transmitting and receiving the signal in the horizontal polarization plane while HV indicates that the signal is transmitted in horizontal but received in vertical plane.

B. Initial data analysis and features

Before we started to create the model, we familiarize ourselves with the main characteristics and features of the data. First, we analyzed the dataset for missing data points and noticed that the field inc_angle had 133 non-existing values but otherwise the length of the image vectors were correct and there was not any other missing data. After this we examined a couple of random samples representing objects from both classes to better understand the structure of the data. The visualization shows that, there is clearly a visual difference in the two different channels. Especially the HV channel (i.e. band_2) shows a different reflection when compared to HH as seen below in Figures 1 and 2. However, the amount of background noise and its high variation between the pictures is already imminent from these samples. This is because various factors affect the backscatter of the ocean. For example, high winds will cause larges waves which will make the background more varying causing also a brighter backscatter. Low winds and calm seas have an opposite effect. [1], [8]

In addition to the samples, we also examined the differences in pixel value distributions between the two different classes. The initial analysis shows

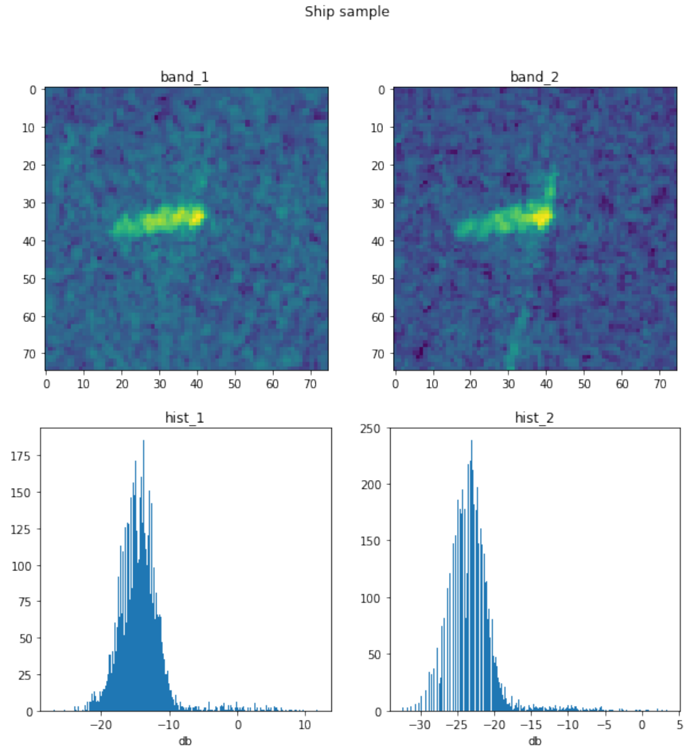


Fig. 1: A sample data vector representing a ship.

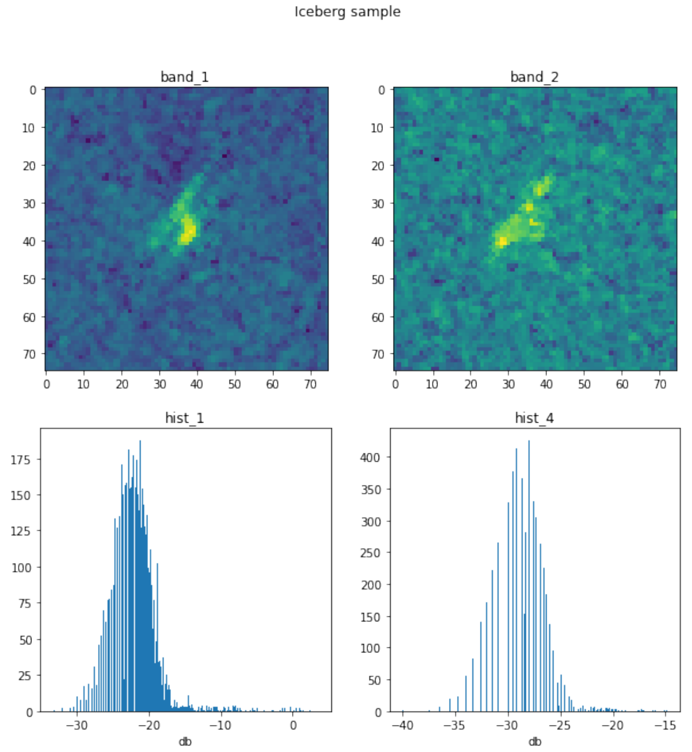


Fig. 2: A sample data vector representing an iceberg.

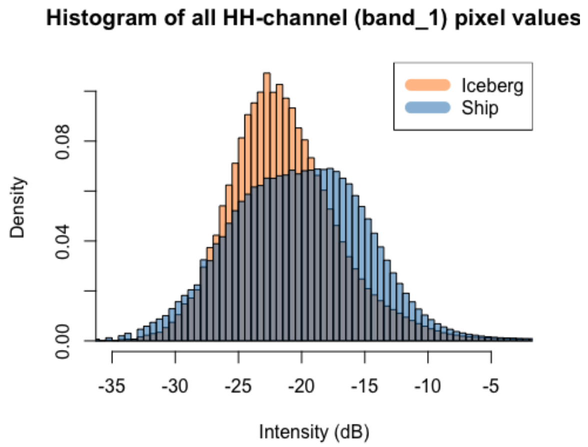


Fig. 3: The distribution density of all the pixel values in band_1 fields.

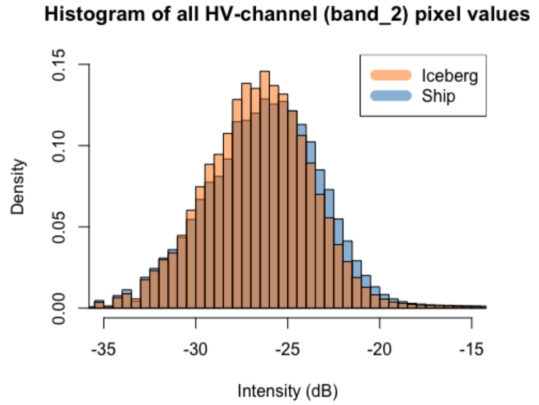


Fig. 4: The distribution density of all the pixel values in band_2 fields.

that the band_1 might include more variance between the classes as can be seen in the Figures 3 and 4. However, as the object classification relies strongly on pattern recognition (i.e. pixel position data) it is premature to draw any conclusions solely based on the distribution analysis.

Finally, we also analyzed the importance of the incidence angle data. Several articles and data sources implied that the incidence angle could be used to reduce the backscatter noise. This is mostly due to the fact that the incidence angle affects the amount of interference the signal experiences in the atmosphere but also impacts the strength as well as polarization of the signal that is bouncing back. [9]–[11] We also hypothesized ourselves whether the incidence angle could carry any information of the latitude or longitude, thus, implying the probability of the existence of icebergs in that area. However, analyzing the distribution of the incidence angle revealed a possible data leakage in the provided data set. The Figure 5 shows there is clearly three different incidence angle ranges which includes only data labeled as icebergs. Furthermore, the ranges are extremely narrow and it looks like that the icebergs with incidence angles in these ranges, could be classified by solely using the inc_angle -field. The same topic is widely discussed on the Kaggle competition discussion boards and it is believed that the data leakage has happened during the labeling process. [1] In practice, this would mean that the human experts labeling the iceberg samples in these groups have used imagery with almost exactly the same incidence angle (e.g. splitting a large image from the arctic sea including no ships into smaller ones to increase the amount of iceberg samples). All in all, after these findings we decided to abandon the incidence angle in our models and concentrate only on creating an image classifier based on the radar imagery data. We decided that figuring out the full extent and nature of the incidence angle in this case would highly exceed the scope of this study. In other words, we wanted to keep focus on studying the use of CNN as an object classifier and not risk that the incidence angle would explain the majority of the labels.

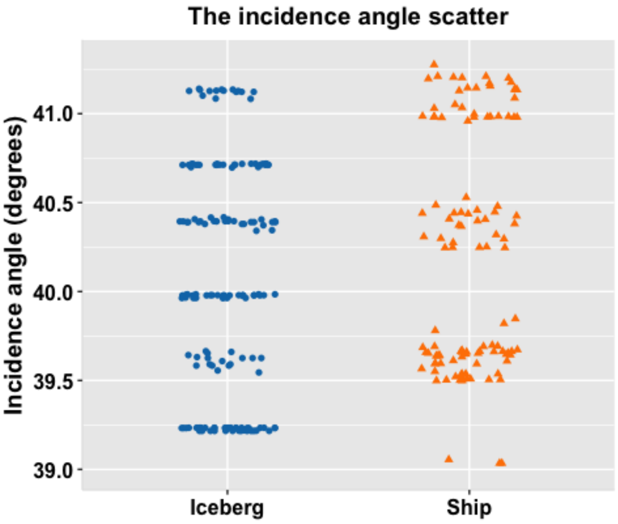


Fig. 5: Incidence angle scatter in the range of 39.0 to 41.3 degrees.

C. Pre-processing

The original data received from the the Sentinel-I satellites were pre-processed at least by splitting and cropping the large radar imagery files to 75x75 pixel images that include only one object at a time at its center. However, the competition or data download page do not imply any other pre-processing of the images except labeling them [1]. Due to the high amount of background noise in the pictures we decided ourselves to standardize the datasets. We standardized each image data point to have a zero-mean and unit variance as well as used batch standardization. These are common techniques to improve the training performance of the model by making it easier for the model to learn the relevant patterns in the data. This was done by subtracting the mean of each images pixel values from each pixel and dividing each pixel value by the standard deviation of the image. Especially convolutional neural networks have been noticed to benefit from standardization as it will help the network to converge towards the same minimum more efficiently. [2], [3]

V. Experiments

We divided the labeled data into a training set of 1200 images and a validation set of 404 images. The goal of the experiment is to optimize a CNN based deep learning model to minimize validation loss of our classification problem.

First models were cross-validated by a learning rate of 0.001 and a batch size of 80. Performances of $[0.001, 0.003, 0.006]$ learning rates were compared with the best CNN structure. Then $[40, 80, 120, 160]$ batch sizes were compared by using the best model structure with the optimal learning rate. Finally, performance of the best model with optimal hyperparameters was studied with the test data in Kaggle.

With a small kernel size, a CNN model can capture finer details of an image [12]. As Figures and 2 shows, sizes of icebergs and ships in the samples were relatively small. To classify satellite images, small details should be noticed. Hence we chose to use the kernel size of 3 for the all our models. In addition, we used stride of 1 and padding of 1.

The Adam optimizer algorithm was utilized to update parameters after every training batch. Unlike in Stochastic Gradient Descent where a single learning rate α is maintained for all weight updates and the learning does not change during training, individual adaptive learning rates for parameters are defined by first and second moments of the gradients in Adam [2]. As no optimization functions are accepted to outperform others, we utilized only Adam, as its adaptive learning rates simplify comparison of learning rates [13].

The first CNN based model consisted of two CNN layers. The first layer has 32 feature maps. The second layer had 64 feature maps. The activation function of the CNN layers was ReLU:

$$f(x) = \max(0, x), \tag{3}$$

where x was the output of the kernel multiplication. The ReLu is widely used with classification models as its gradients do not usually explode [2]. Max Pooling with a size of 2×2 is used as a pooling layer. Three Fully Connected (FC) layers processed activated features of the CNN. The widths of the layer are sequentially $[120, 84, 2]$. ReLu was the activation function of the first and second FC layers. The output of the last layer was fed to the Softmax function.

The second CNN based model consisted of three CNN layers. The number of the feature maps were sequentially $[64, 128, 64]$. The third CNN based model had four CNN layers of which number of feature maps are $[64, 128, 128, 64]$. Otherwise the structures of the second and third models were the same as the first model had.

With limited amount of training data, the built model eventually overfitted after a few epochs. To improve the performance of the model with the discovered optimal hyperparameters, we generated augmented data. We created new training data with seven different augmentation techniques. These resulted in 14 new samples from one sample. After data augmentation, the training set was 18000 in total. Appendix A shows augmented images of a random sample. The structure of the image is very simple. An iceberg or a ship is located near the center of the image. Otherwise the image consists of surface of the sea. Surface of the sea can be considered as a noise that can be normalized. Here our augmentation techniques, f.ex. zooming in or out, does not damage the images. In addition, scale of a position movement, a rotation, a zoom, and a translation along diagonal are randomly drawn from a suitable interval for every sample.

VI. Results

The Figure 6 shows the validation losses of our different CNN based models. Learning rate for the experiment is 0.001 and the batch size is 80. The model with two CNN layers starts overfitting after 80 iterations. The two other models perform almost equally. However, the variance of the learning at the end seems to be higher with the three layer CNN model. The model with four CNN layers achieves a validation loss of 0.27741 and accuracy of 0.87623. Its results are better than those of other models. The model with two CNN layers could not achieve good results because of the high complexity of the problem. In addition, the kernel size of 3 requires depth structure to model very detailed features [6].

The Figure 7 shows the validation losses with different learning rates. Small improvements can be achieved with learning rate tuning. With 0.006 learning rate the most stable learning and the best performance is achieved. With 0.001 learning rate almost as good results are achieved, but its learning is not stable. As the learning rate is small, the gradient might not have been able to move parameters to an optimal direction first.

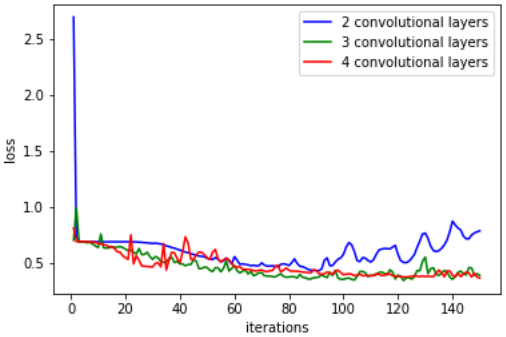


Fig. 6: Validation losses of the different CNN structures.

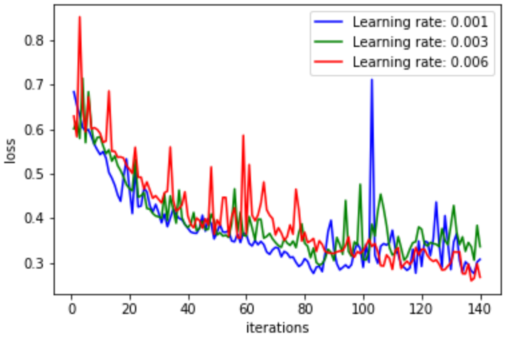


Fig. 7: Validation losses with different learning rates.

Batch size	Min Validation Loss	Max Accuracy
40	0.20686	0.93811
240	0.20011	0.93564

TABLE II: Minimum validation losses and maximum accuracies with data augmentation

The Table II presents results with different batch sizes when the best model structure with the optimal learning rate of 0.006 is used. The best validation loss is achieved with a batch of 40. This seems to be reasonable as with smaller batch sizes overfitting can be prevented because variance of image features in a batch is higher [14]. However, no remarkable increase of the performance is achieved with tuning learning rates and batch sizes.

The Table III shows the results of the augmented data addition. Better results of both validation accuracy and loss are achieved. The Figure 6 shows that with larger training data, larger batch size can achieve equally good results with less gradient updates.

Batch size	Min Validation Loss	Max Accuracy
40	0.2614160	0.8935
80	0.2757081	0.9009
120	0.2697816	0.9009
160	0.3020098	0.8960

TABLE III: Minimum validation losses and maximum accuracies with batch sizes α

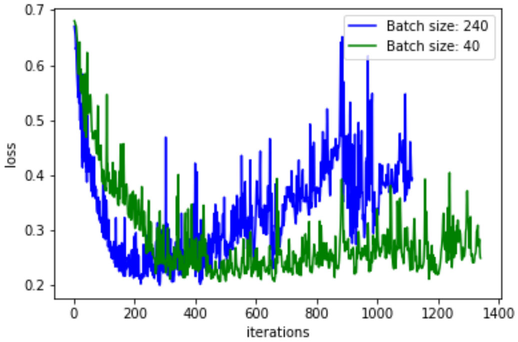


Fig. 8: Validation losses with augmented data.

VII. Discussion

As discussed in the section II, the convolutional neural networks are widely used in image classification. For us, the best model achieved the accuracy of 0.20011 with four layers, batch size of 240 and learning rate 0.006 when the model was trained with the augmented data. This compares well with the XU and Scott research of classifying sea ice and ocean from similar features with the accuracy of 92.36%. [5]

There is also plenty of comparable results on the Kaggle competition page as well. The winning submission received an validation log loss of 0.0822 which is considerably better than ours. [15] However, the winning team revealed in a Kaggle discussion post that they had exploited the data leakage related to the incidence angle. [16] Therefore, their results can not be directly compared with our model as we decided to ignore the inc_angle field due to uncertainties in the data leakage. Teams that did not use the incidence angle but trained a CNN based classifier received similar results to ours hovering around the log loss of 0.2 and accuracy of 90%. However, the implementations not using the data augmentation received slightly weaker results than we did with augmented data. [12]

The data augmentation had clear a significant impact on the accuracy as we expected. This was due to relatively low amount of data we had and generating more training data by utilizing augmentation prevented the models of overfitting as easily. Overall it proved to be an efficient method in this classifying problem as discussed more below.

VIII. Conclusions

As the model needs to be detect small details from the image, a small kernel size is required. Our results showed that the model achieved better results with a deeper CNN structure. When each feature maps detects small features from the image, deeper networks can better generalize the problem.

Optimizing learning rate did not provide significant improvements. As the Adam algorithm adjust its learning rate parameters, initial learning rate have smaller influence on a learning convergence. As the data size is limited, with small batch better overfitting can be prevented. With small batch size, large variance occurs within a batch set.

Data augmentation proved to have significant influence on the learning improvements. We were able to increase training data from 1200 samples to 18000 with different augmentation techniques (Appendix A). The validation loss decreased from 0.2614160 to 0.20011 with data augmentation.

IX. Roles of the authors

In the beginning of the project, we divided the responsibilities so that Kuusisto would concentrate more on the data analysis, literature review and planning the used methods. Hämäläinen would concentrate more on the implementation and designing the model structure. The responsibility of writing the report and executing the experiments were evenly divided between the both of us.

References

[1] Statoil, "Statoil/C-CORE Iceberg Classifier Challenge," 2018. [Online]. Available: https://www.kaggle.com/c/statoil-iceberg-classifier-challenge

[2] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning*, 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *ImageNet Classification with Deep Convolutional Neural Networks*, 2012.

[4] D. Mishkin, N. Sergievskiy, and J. Matas, "Systematic evaluation of convolution neural network advances on the Imagenet," *Computer Vision and Image Understanding*, 2017.

[5] Y. Xu and K. A. Scott, "Sea ice and open water classification of sar imagery using cnn-based transfer learning," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017, pp. 3262–3265.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Alexnet," *Advances In Neural Information Processing Systems*, 2012.

[7] PyTorch Community, "Tensors and Dynamic neural networks in Python with strong GPU acceleration," 2016. [Online]. Available: https://github.com/pytorch/pytorch

[8] ESA, "Introducing Sentinel-1," 2017. [Online]. Available: http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Sentinel-1/Introducing_Sentinel-1

[9] D. Sabel, M. Doubková, W. Wagner, P. Snoeij, and E. Attema, "A Global Backscatter Model for C-band SAR," 2010.

[10] C. Nie and D. G. Long, "A C-band wind/rain backscatter model," *IEEE Transactions on Geoscience and Remote Sensing*, 2007.

[11] D. Sabel, Z. Bartalis, W. Wagner, M. Doubkova, and J. P. Klein, "Development of a Global Backscatter Model in support to the Sentinel-1 mission design," *Remote Sensing of Environment*, 2012.

[12] Devesh Maheshwari, "Keras Model for Beginners (0.210 on LB)+EDA+R&amp;D." [Online]. Available: https://www.kaggle.com/devm2024/keras-model-for-beginners-0-210-on-lb-eda-r-d

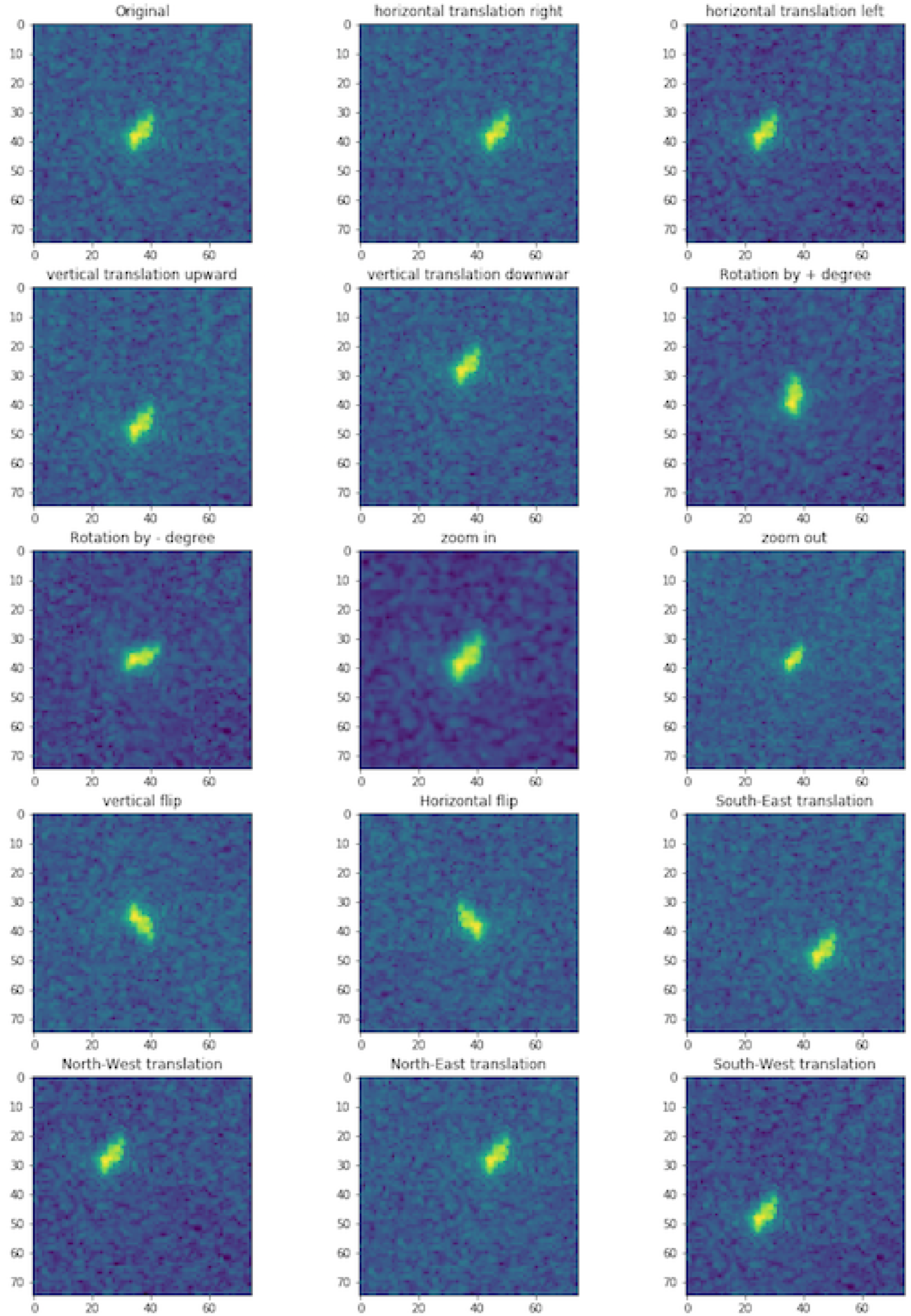
[13] D. P. Kingma and J. L. Ba, "Adam: a Method for Stochastic Optimization," in *Proceedings of International Conference on Learning Representations*, 2015.

[14] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," *CoRR*, vol. abs/1609.04836, 2016. [Online]. Available: http://arxiv.org/abs/1609.04836

[15] Kaggle, "Statoil/C-CORE Iceberg Classifier Challenge - Private Leaderboard," 2018. [Online]. Available: https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/leaderboard

[16] David - Kaggle user, "1st Place Solution overview," 2018. [Online]. Available: https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/discussion/48241

A. Data Augmentation


Original
horizontal translation right
horizontal translation left
vertical translation upward
vertical translation downwar
Rotation by + degree
Rotation by - degree
zoom in
zoom out
vertical flip
Horizontal flip
South-East translation
North-West translation
North-East translation
South-West translation