Iceberg Detection and Analysis Final Report

Sania Bandekar and Victoria Le
Department of Computer Science, San Jose State University, California

I. Introduction

DATA regarding icebergs plays a crucial role in monitoring climate change and understanding its effects on sea levels and ecosystems. Using synthetic aperture radar (SAR) imaging combined with advanced despeckling techniques, we developed a robust methodology to analyze icebergs. This enables predictions on the effects of climate change and provides actionable insights into mitigating potential ecological and environmental impacts.

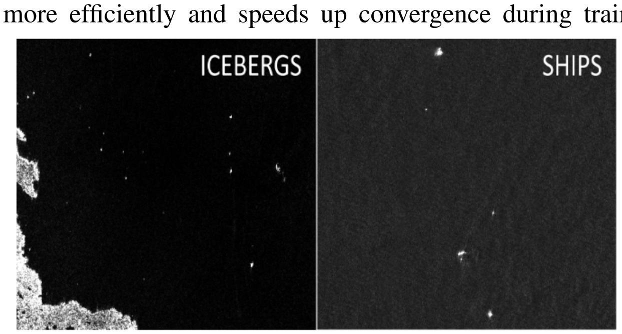
II. Methodology

A. Research and Model Selection

Recent advancements in deep learning, particularly convolutional neural networks (CNNs) and U-Net architectures, have shown remarkable accuracy in processing satellite images. Our approach leverages these techniques to achieve high precision in detecting icebergs and estimating their dimensions. These methods were selected due to their effectiveness in handling large-scale datasets with noise and their ability to segment and classify images [1].

B. Data Preparation

The dataset for this project was obtained from Kaggle due to the limited availability of suitable SAR imaging datasets. The images were preprocessed to enhance important features for iceberg detection. The dataset consists of JSON files containing image data (band 1) and corresponding labels (isiceberg). Each image is a 75x75 pixel grayscale image, and the label indicates whether the image contains an iceberg or not. The images were reshaped and normalized. The JSON files were converted to NumPy arrays to serve as input for training and validation [2] We normalized the images to scale the pixel values between 0 and 1. This step helps the model learn more efficiently and speeds up convergence during training.



C. Convolutional Neural Network Architecture

- Convolutional Layers The implemented CNN architecture consists of three convolutional layers with batch normalization and ReLU activation. In the convolutional layers, we started with 32 filters, increasing to 128 filters in subsequent layers, using 'same' padding to ensure the output has the same spatial dimensions as the input. These layers extract high-level features from the images
- Max Pooling There is max pooling with a 2x2 window to reduce spatial dimensions, while retaining the most critical information.
- Model Regularization To prevent overfitting, we used dropout with a rate of 30 percent. Additionally, we applied L2 regularization to the convolutional and dense layers to penalize large weights and improve generalization.
- Dense Layers We used a dense layer with 128 neurons, followed by a second dense layer with a single output unit. The final layer uses a sigmoid activation function, which is typically used for binary classification tasks like iceberg detection. The output will be a probability between 0 and 1, representing the model's confidence in the prediction.

D. Model Compilation

The binary classification task (distinguishing between icebergs and ships) was optimized using binary cross-entropy loss and the Adam optimizer.

E. Model Training and Evaluation

The model was trained on the preprocessed dataset, with a validation split to assess its performance during training. The final evaluation showed high accuracy and a low loss value, indicating a well-trained model. Training plots, as shown below, illustrate the steady improvement of accuracy and reduction of loss over epochs.

The model was trained for 50 epochs with a batch size of 32. After every 32 samples, the model updates its weights. Validation data was used to monitor the model's performance, detect overfitting and adjust the learning rate when necessary

To improve training stability, we used the ReduceLROnPlateau callback to reduce the learning rate by 50% if the validation loss stopped improving for three consecutive epochs

F. User Interface Development

A user-friendly desktop application was developed using Python, allowing users to run the model and produce charts for Model Accuracy and Model Loss over epochs.

G. Challenges

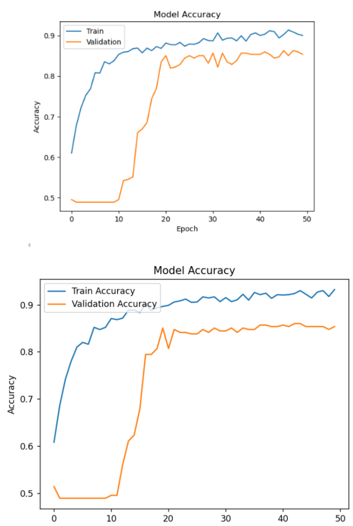
We had previously mentioned that we would include a feature for uploading images and getting predictions, however we ran into complications regarding this due to not being able to obtain images that met all the requirements for the model to work with.

III. Results and Discussion

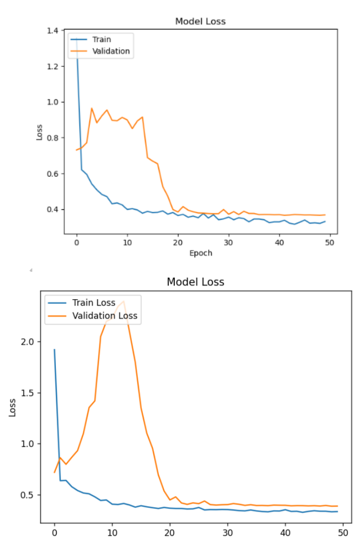
Since the check in, the improvements made to the model yielded slightly better results. For model accuracy, percentages increased from 0.90 to 0.83 while loss decreased from 0.37 to 0.21 in the first set. For the second set, the accuracy seems to be similar but loss was 0.30 which is slightly better results since the project check in where loss was slightly under 0.5

For the figures below, the charts from the progress check in are shown first, followed by the most recent run.

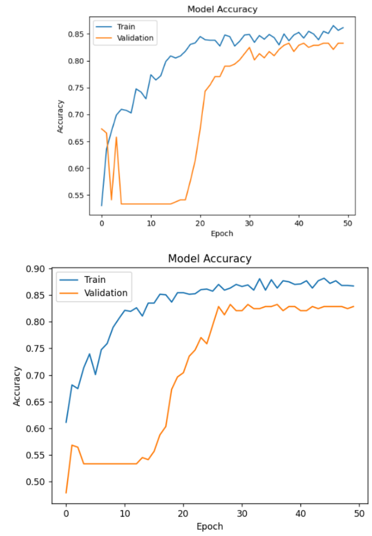
IV. Comparison for Set 1 Model Accuracy



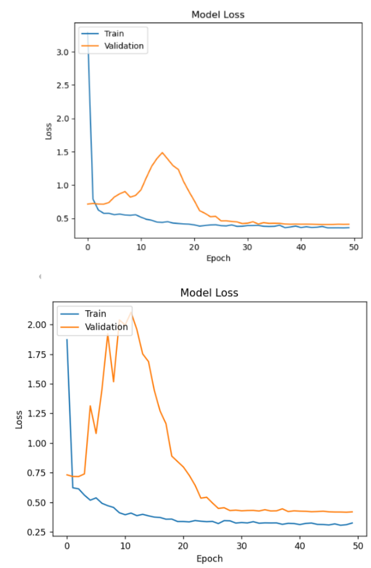
V. Comparison for Set 1 Model Loss



VI. Comparison for Set 2 Model Accuracy



VII. Comparison for Set 2 Model Loss



VIII. Conclusion

This project successfully developed a system for iceberg classification using SAR images and deep learning techniques. The CNN architectures was very effective in identifying icebergs and the user-friendly interface ensures accessibility and ease of use.

References

[1] David Hogg Anne Braakmann-Folgmann, Andrew Shepherd and Ella Redmond. Mapping the extent of giant antarctic icebergs with deep learning, 2023.

[2] Kaggle. Statoil/c-core iceberg classifier challenge, 2017.