ICEBERG DETECTION IN SATELLITE IMAGES USING IBM WATSON STUDIO

MINI PROJECT REPORT

SUBMITTED BY : (BATCH NO :CSE_002)

I.	PAGADALA HEMA	(18UK1A0592)
II.	KASULA MEGHANA	(18UK1A05E2)
III.	VEMULA VAMSHI	(18UK1A05B7)
IV.	MUCHIMPULA AJAY	(18UK1A0589)

INDEX

1. Introduction

Iceberg detection is found to be more critical in the previous researchers. High quality satellite monitoring of dangerous ice formations is critical to navigation safety and economic activity in the regions. The satellite images play a crucial role in the identification of the icebergs. In this manuscript, a convolutional neural network (CNN) model is proposed for the iceberg detection from the satellite images. It is based on the satellite dataset for target classification and target identification. The iceberg detection is based on the statistical criteria for finding the satellite images. This model is used to identify automatically whether it is remote sensed target is iceberg or not. Sometimes the iceberg is wrongly classified as ship. This model is done to make accurate about the changes in the detection. Though navigation in iceberg-infested waters has been recognised from early times to be fraught with special dangers, it is only comparatively recently that attempts have been made to apply scientific methods to the detection of these floating dangers to navigation. Early whalers and explorers in Arctic and Antarctic waters met and surmounted these dangers without such assistance, but the present circumstances of sea travel in waters occasionally subject to invasion by icebergs are so different as to render special precautions necessary. Some of these circumstances are the increasing size and speed of passenger vessels, the replacement of wooden hulls by steel, and the need, in the face of competition from rival steamship companies, to complete the voyages within scheduled times. The small ice 'growler,' floating almost submerged, is not only the most difficult to detect, but is also almost as dangerous an obstruction to modern steel ships as the iceberg of large dimensions, which is likely to be more readily perceived even in a fog.

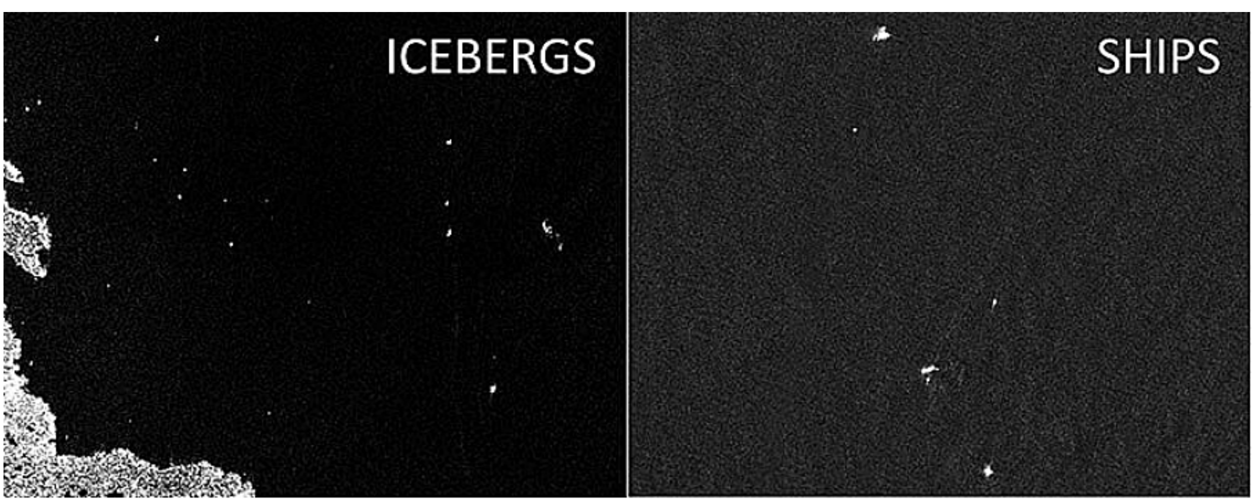


Fig. Satellite images of ships and iceberg

1.1 Overview

Icebergs present serious hazards for ship navigation and offshore installations. To develop methods for detecting icebergs using satellite radar data and high spatial resolution images in the visible spectral range. The methods of satellite monitoring of dangerous ice formations, like icebergs in the Arctic seas represent a threat to the safety of navigation and economic activity on the Arctic shelf. The developed method of iceberg detection is based on statistical criteria for finding gradient zones in the analysis of two-dimensional fields of satellite images. The approaches proposed to detect icebergs from satellite data allow improving the quality and efficiency of service for a wide number of users with ensuring the efficiency and safety of Arctic navigation and activities on the Arctic shelf. Deep learning is a piece of AI systems that is utilized by the layers to draw out the more significant level highlights from the crude info.

So as to get data, remote detecting gave for examining the earth by satellite or high-flying airplane. Recognizing remote detecting pictures gives the better and more noteworthy test for the savvy examine researchers to find the Iceberg acknowledgment in the specific way. Remote detecting frameworks is utilized to distinguish chunks of ice are housed on satellites over the earth. The satellite star grouping is utilized to screen the seas. The satellites catch the pictures at the given area of surface in time at a given moment.

1.2 Purpose

The main aim of this project is to build a model that automatically identifies whether a remotely sensed target is an iceberg or not. Often times an iceberg is wrongly classified as a ship. The algorithm had to be extremely accurate because lives and billions of dollars in energy infrastructure are at stake.

2. Literature Survey

The image analysis technique developed to identify icebergs depicted in synthetic aperture radar images of Antarctica and to determine the outlines of these icebergs. The technique uses a pixel bonding process to delineate the edges of the icebergs. It then separates them from the background water and sea ice by an edge-guided image segmentation process. Characteristics such as centroid position and iceberg area were calculated for each iceberg segment and placed in a file for input to appropriate statistical data analysis software. The technique has been tested on three ERS-1 SAR sub-images in which it succeeded in identifying virtually all segments containing icebergs of size six pixels or larger. The images were first passed through an averaging filter to reduce speckle. This process produced a pixel size of 100m x 100m. As implemented, the technique overestimates iceberg areas by about 20% on average and the detection rate falls off rapidly for icebergs less than six pixels in size. Performance in these areas is expected to improve when additional stages, based on a more detailed analysis of pixel intensity, are implemented.

2.1 Existing Problem

Many businesses and money trade are relying on transport and logistic services. Transportation of heavy goods is only possible with ship transport, but navigation through sea and oceans is not easy. Many ships have hit icebergs and have sunk causing great loss of lives and goods value.

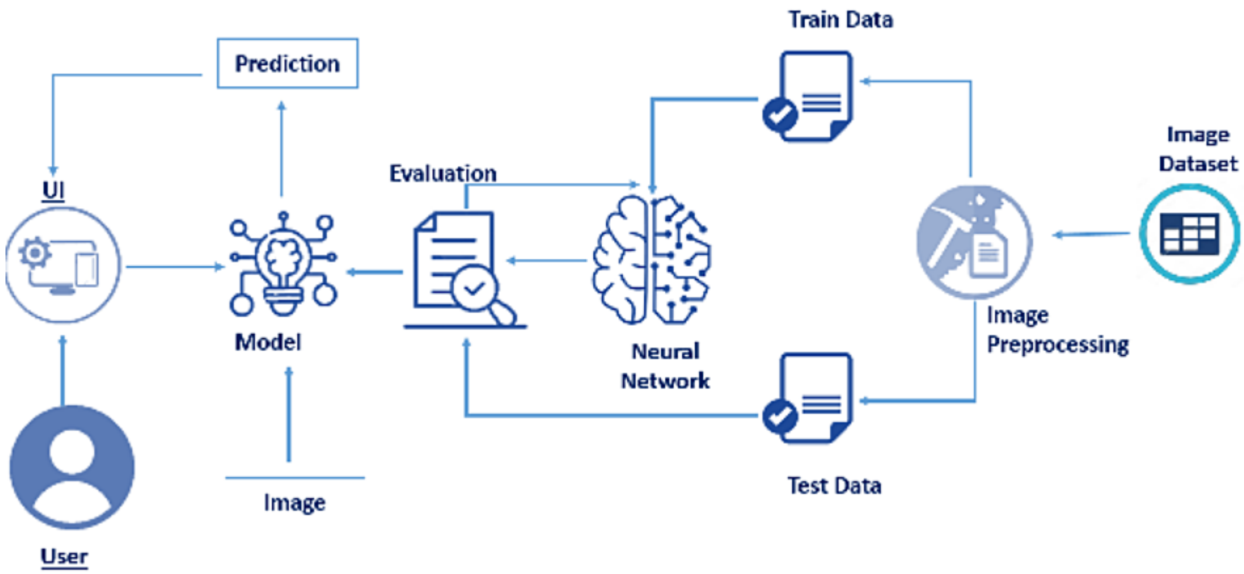
2.2 Proposed Solution

To build an algorithm which automatically identifies whether a remotely sensed target is an iceberg or not. Often times an iceberg is wrongly classified as a ship. The algorithm had to be extremely accurate because lives and billions of dollars in energy infrastructure are at stake. As it is a tedious process to record live radar images, we can create a web application where user can upload available radar. images. This uploaded image is analysed by the model which is built and prediction is showcased on the UI.

3.Theoretical Analysis

It is important to detect Iceberg and ship. Icebergs present serious hazards for ship navigation and offshore installations. Consequently, there is a large interest to localize them timely and over vast areas. Because of their independence of cloud cover and daylight, satellite Synthetic Aperture Radar (SAR) images are among the preferred data sources for operational ice conditions and iceberg occurrences. The methods of satellite monitoring of dangerous ice formations, like icebergs in the Arctic seas represent a threat to the safety of navigation and economic activity on the Arctic shelf. In deep learning, this is generally done by extracting features through a convolutional neural network (CNN) and then classifying using a fully connected network. We have trained a convolutional neural network and obtained a prediction accuracy of up to 83.19%. CNN is a modified variety of deep neural net which depends upon the correlation of neighbouring pixels. It uses randomly defined patches for input at the start, and modifies them in the training process. Once training is done, the network uses these modified patches to predict and validate the result in the testing and validation process. Convolutional neural networks have achieved success in the image classification problem, as the defined nature of CNN matches the data point distribution in the image. As a result, many image processing tasks adapt CNN for automatic feature extraction.

3.1 Block Diagram



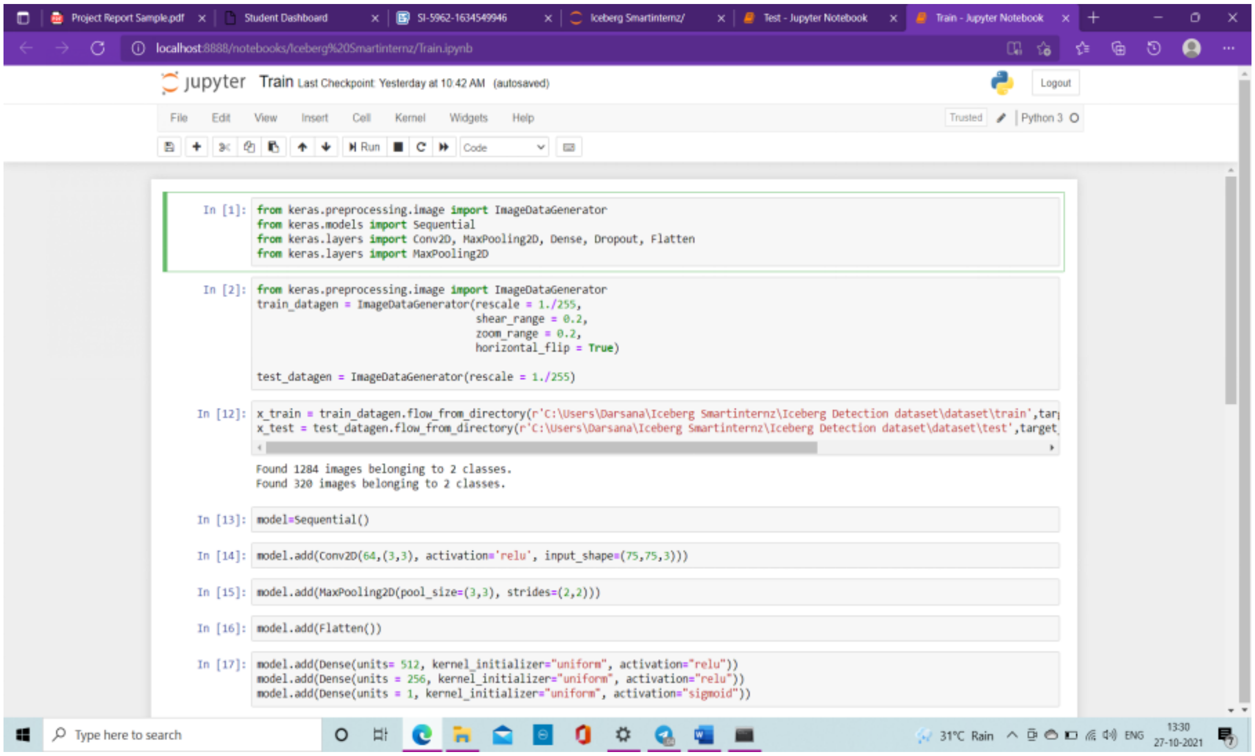
3.2 Software Designing

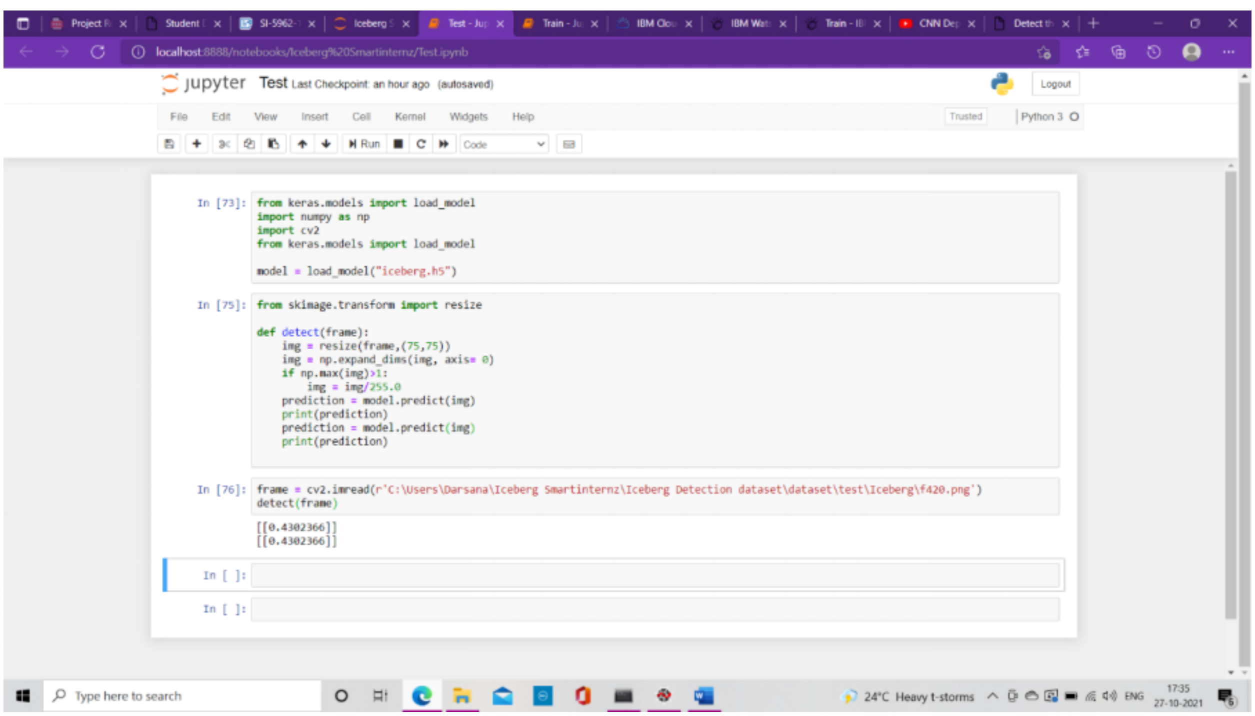
1. Jupyter Notebook Environment
2. Spyder
3. Deep Learning Algorithms
4. Python
5. HTML
6. Flask

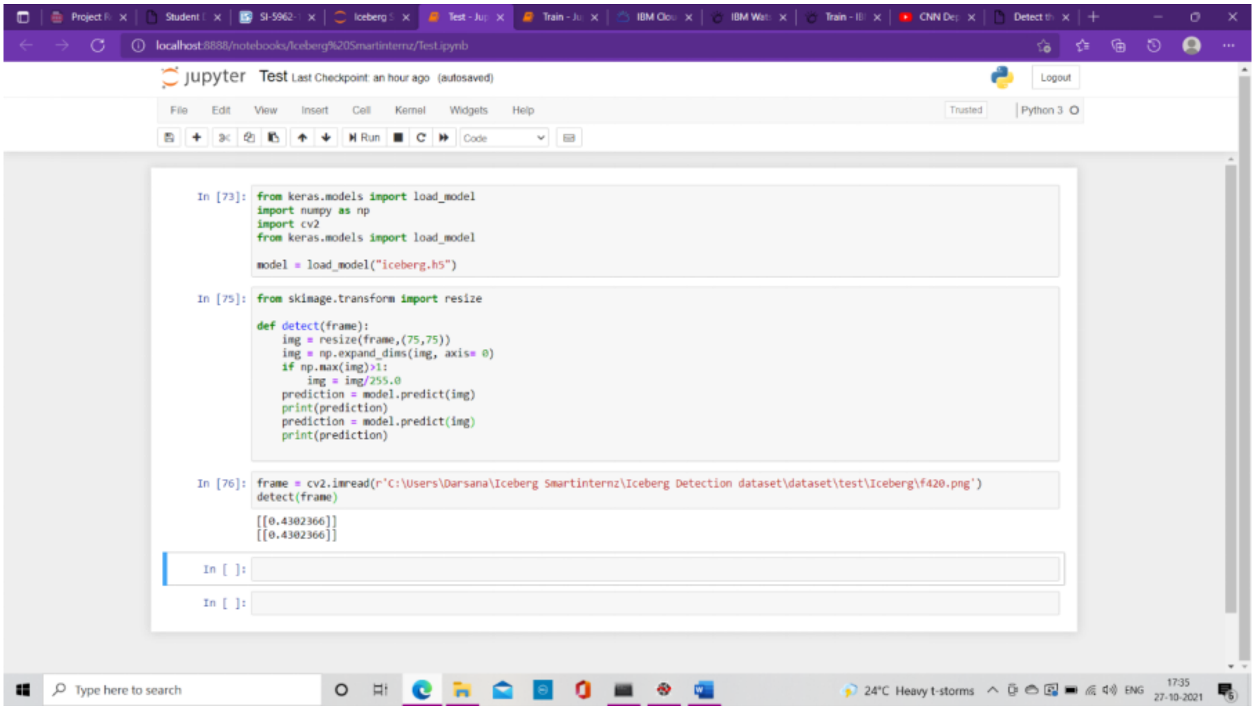
We developed this Iceberg detection by using the Python language, which is a high-level programming language along with Deep Learning Algorithm such as CNN. For coding we used the Jupyter Notebook of Anaconda distributions and Spyder, an integrated scientific programming in python language. Flask is used as a user interface for the prediction. Hypertext Markup Language (*HTML*) is the standard markup language for documents designed to be displayed in a web browser.

4. Experimental Investigation

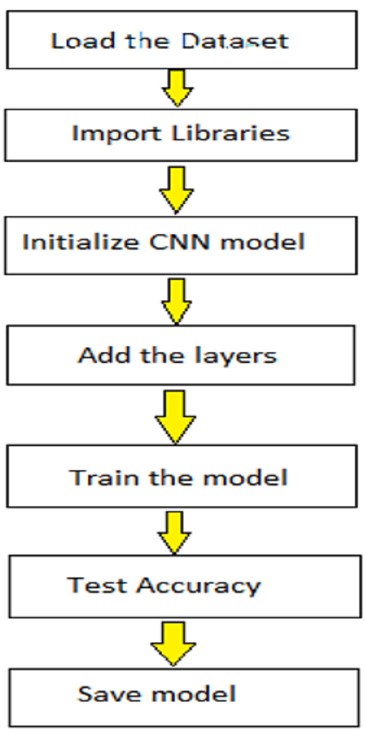
In our project, we have used the **Iceberg detection Dataset**. This dataset contains two folders: test and training. In test folder, we have two categories called **Iceberg** and **Ship**, where, has the images having Iceberg and ships has the images of ships, similarly in the training folder. Having 1284 images belonging to 2 classes in training set and 320 images belonging to 2 classes in test set





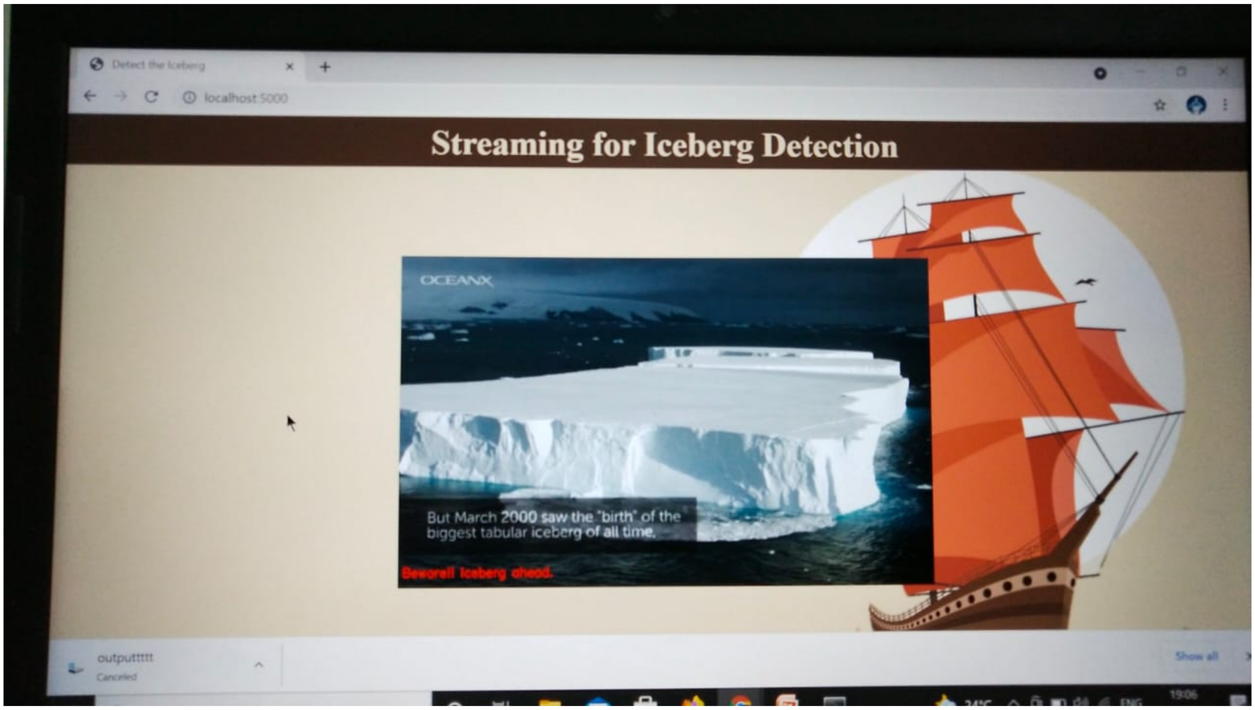

jupyter Test Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
In [73]: from keras.models import load_model
import numpy as np
import cv2
from keras.models import load_model
model = load_model("iceberg.h5")
In [75]: from skimage.transform import resize
def detect(frame):
img = resize(frame,(75,75))
img = np.expand_dims(img, axis= 0)
if np.max(img)>1:
img = img/255.0
prediction = model.predict(img)
print(prediction)
prediction = model.predict(img)
print(prediction)
In [76]: frame = cv2.imread(r'C:\Users\Darsana\Iceberg Smartinternz\Iceberg Detection dataset\dataset\test\Iceberg\f420.png')
detect(frame)
[[0.4302366]]
[[0.4302366]]
In [ ]:
In [ ]:
Type here to search
24°C Heavy t-storms
ENG
27-10-2021

5. Flow Chart


Load the Dataset
Import Libraries
Initialize CNN model
Add the layers
Train the model
Test Accuracy
Save model

6. Result

The CNN model to detect icebergs from the images was developed successfully with an accuracy of 73%. This model is now used to classify the images, as Iceberg or Ship, uploaded in a web page which is integrated with Flask.



7. Advantages and Disadvantages

Merits:

We are able to detect icebergs in an image, so that it would be of great help to the logistics and transportation team.

1. It could bring a whole new dimension of transport for container ships and vessels by tracking icebergs from satellite images in real-time.

1. Iceberg detection is easy to implement and understand.

2.It operates in real-time due to low time complexity.

It is applicable in training and test-time

Deliver invariance with respect to the lesion position, scale and rotation.

1. Remote Sensing is an effective tool to monitoring and track some large tabular iceberg drift in the Southern Ocean. Satellites produce a wide range of instruments and data, are convenient, and provide all year coverage over a vast region of earth.

Demerits:

The time between ground tracking and the repeat cycle. This means that the system is not responsive enough to track small or complex iceberg situations.

A different satellite may pass over the region a few days later its cargo maybe a different sensor, hardware and software that will produce different imagery and can produce different results.

8. Applications

1. "Iceberg Detection using Satellite images" simplifies the management process of Iceberg detection by deploying a web interface to the users.
2. Fast processing and immediate results with high accuracy.
3. Minimizing human effort and cost-efficient databases.
4. Easy and simple interface to understand and help the user.

9. Conclusion

This is one of the great examples where deep learning can be used to solve a challenging real-world problem. If we are able to detect and segment icebergs in an image, it would be of great help to the logistics and transportation team in northern countries. It could bring a whole new dimension of transport for container ships and vessels by tracking icebergs from satellite images and videos in real-time. This implies that CNN can extract better features when presented with SAR images with different number of looks for the same target. The CNN is treated as a black box model, and additional experiments are necessary to understand how the change in image resolution affects the CNN internal activations. The use of additional procedures in the pre-processing step can potentially improve the classification results, and will be the focus of future work.

10. Future Scope

The extraction of the iceberg detection from the satellite imagery can be used in

1. Digital cartography updating: with the increasing availability of the remotely sensed imagery the need for reliable and up-to-date maps is growing. The bottleneck in the production of cartographic data lies in the manual processing applied to the data.

2. Multi-temporal change analysis: Change analysis from remotely sensed data has been studied for more than thirty years by the Digital Image Processing and Pattern Recognition communities.
High resolution imagery raised new difficulties: occlusions, projective distortion, detail profusion or the presence of shadows, create “apparent changes” which do not correspond to real changes of the scene, and therefore make the interpretation difficult.

3. Content-based image indexation: The task of automatic image labelling and classification is becoming crucial with the exponential increase of the number of available images and their variety in terms of acquisition sensor and resolution.

11. Bibliography

Model Building

1. Dataset

2. Jupyter Notebook

Application Building

1. HTML file

2. Static files

3. Flask app

4. Spyder

Appendix

A. Source code

HTML File:

```
<html>
  <head>
    <title>Detect the Iceberg</title>
    <style>
img {
display: block;
margin-left: auto;
margin-right: auto;
width: 45%
.header {
top:0;
margin:0px;
left: 0%;
right: 0px;
position: fixed;
background: #574032;
color: #f1e6c8;
```

```
overflow: hidden;

padding: 0.5%;

font-size: 2.9vw;

width: 100%;

padding-left:0px;

text-align: center;

}

    </style>

  </head>

<body style ="background-image:url({{url_for('static',filename='images/bck.png')}});background-position: center;background-repeat: no-repeat;background-size: cover;">


<div class="header">

    <div style="align:center"><b>Streaming for Iceberg Detection</b></div>

</div>


                        <div style="margin-top:12%">

<img id="bg" class="center" src="{{ url_for('video_feed') }}">

<!img id="bg" class="center" src="samp.png">

</div>



   </body>
```

</html>

Flask program - app.py

```
import numpy as np
import cv2
from keras.models import load_model
from flask import Flask, render_template, Response
import tensorflow as tf
global graph
global writer
from skimage.transform import resize
#graph = tf.get_default_graph()
writer = None

model = load_model('iceberg.h5')

app = Flask(__name__)
 print("[INFO] accessing video stream...")
vs = cv2.VideoCapture("iceberg1.mp4")
 pred=""


def detect(frame)
```

```
img = resize(frame,(75,75))
        img = np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img = img/255.0
        #with graph.as_default():
        prediction = model.predict(img)
        pred = [prediction[0][0]]



        if pred:
                text = "Beware!! Iceberg ahead."
        else:
                text = "You are safe! It's a Ship."
        return text





@app.route('/')
def index():
    return render_template('index.html')




def gen():
        while True:
                (grabbed, frame) = vs.read()
```

```
if not grabbed:
        break

   data = detect(frame)

   #output frame
   text = data
    cv2.putText(frame, text, (10, frame.shape[0] - 25),cv2.FONT_HERSHEY_SIMPLEX, 0.85,   (0,
    0, 255), 3)
   cv2.imwrite("1.jpg",frame)

   key = cv2.waitKey(1) & 0xFF
   # if the `q` key was pressed, break from the loop
   if key == ord("q"):
       break
   fourcc = cv2.VideoWriter_fourcc(*"MJPG")
   writer = cv2.VideoWriter(r"output.avi", fourcc, 25,(frame.shape[1], frame.shape[0]), True)

   if(pred==0):
       playsound('cut_alert.mp3')
   (flag, encodedImage) = cv2.imencode(".jpg", frame)
   yield (b'--frame\r\n' b'Content-Type: image/jpeg\r\n\r\n' +
                   bytearray(encodedImage) + b'\r\n')
```

```python
        #cv2.destroyAllWindows()




@app.route('/video_feed')

def video_feed():

    return Response(gen(),

                    mimetype='multipart/x-mixed-replace; boundary=frame')


if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=False)
```