Iceberg Detection In Satellite Images

Index

Iceberg Detection In Satellite Images

1. INTRODUCTION

1.1 Overview

Project entitled 'Iceberg Detection In Satellite Images', was given with dataset of train and test with Iceberg and Ship.
From the given, we need to build a CNN model, which later then used for detecting icebergs in the video.
Model got good accuracy of 80's, which is not overfirtted the given dataset, instead used for general purpose.
Later 'Flask' web application will be built by using model generated and work demonstrated using a typical iceberg video , which renders output on video.

1.2 Purpose

We all know how 'Titanic Ship' ended its journey, by brutally invading the people inside to death.
Tragedic incident like the above can be easily pre-detected by using the technology(real time satellite image and cnn) by giving correct message to captain of the ship, so many life can be protected.

Iceberg Detection In Satellite Images

2. Literature Survey

2.1 Exisiting probelm:

Less ships with the technology enabled, which can be implemented with very less cost using AI.

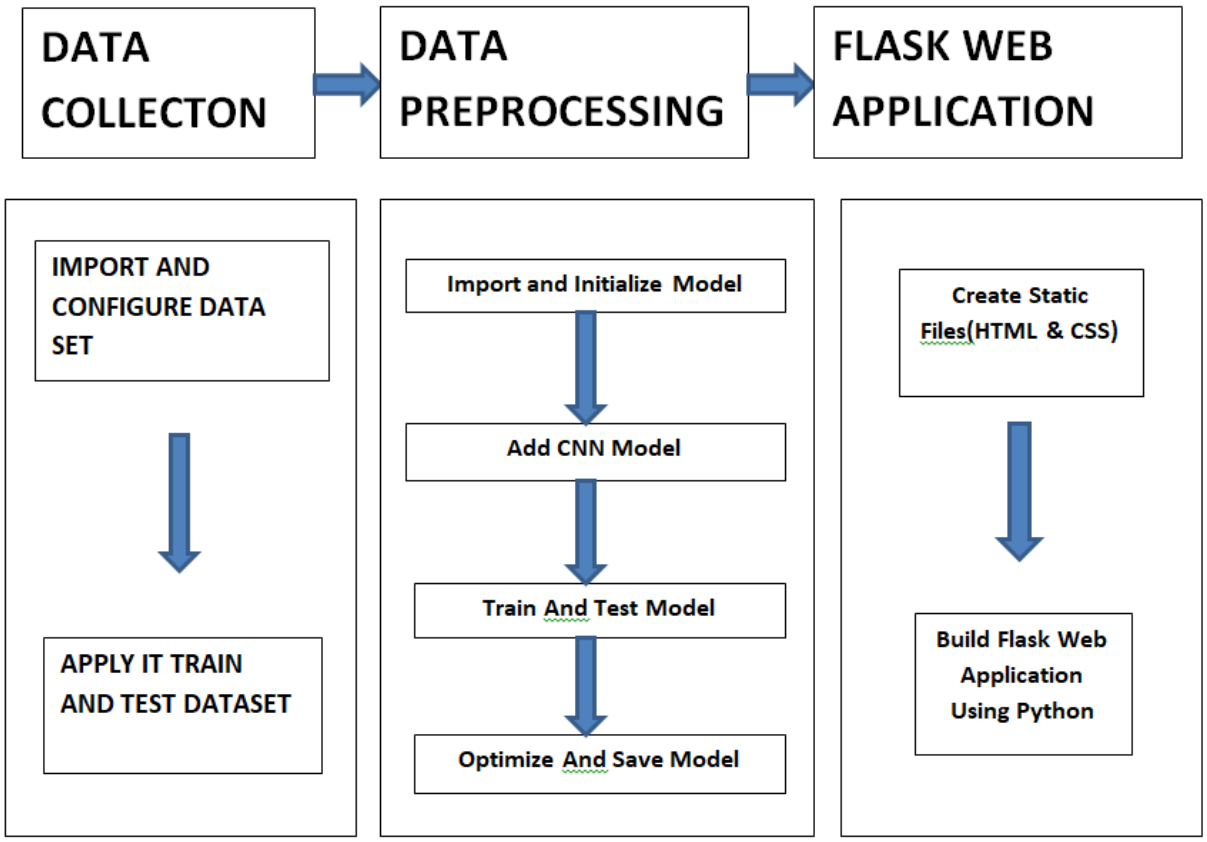
2.2 Proposed solution

By using CNN model built, we can detect whether there is iceberg or not while travelling through sea route, so the accident which can arrive in future can be prevented, hence saving lot of resorces along with lives.

3. Theoritical Analysis

3.1 Block Diagram

Iceberg Detection In Satellite Images



3.2 Hardware / Software Designing

1. Strategy: Using CNN to check whether iceberg or ship will be found in given image/video .
2. Dataset Creation: Data Collection.
3. Data Preprocessing:
    a. Importing Dataset
    b. Using joint plot for checking distribution of various images in given dataset.

4. Buiding Model:
    a. Import required libraries for building model.

b. Initilaizing model.
c. Loading preprocessed data.
d. Adding Conv2D, MaxPooling and Dropout layers.
e. Adding flatten layer.
f. Addign Dense layer.
g. Configure learning process.
h. Train and Test model.
h. Optimize and save model.

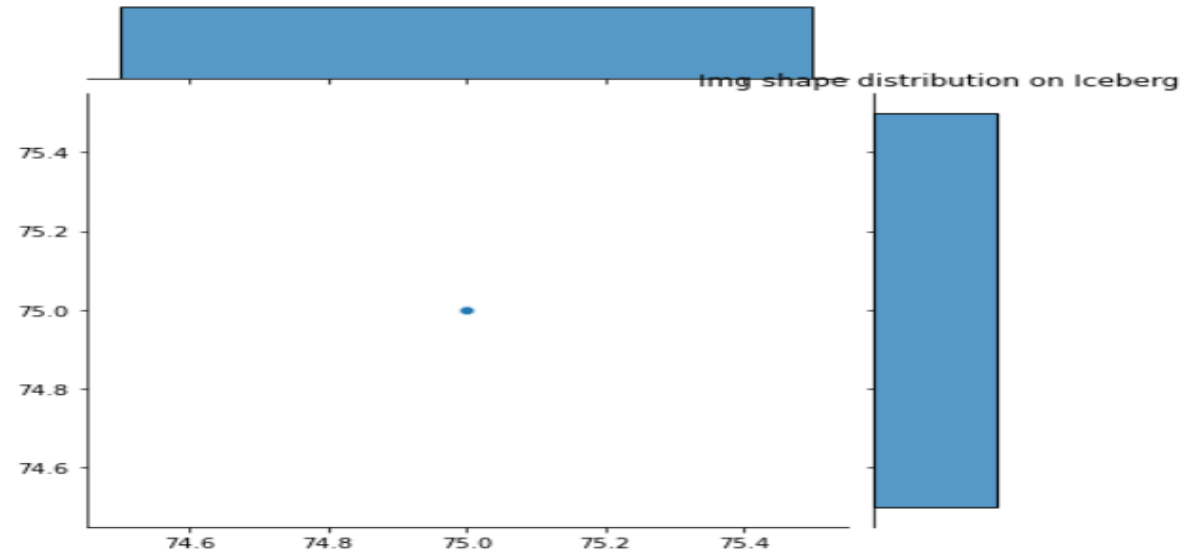
5. Flask application.
    a. Create template using HTML and include static files like CSS and JS(if required).
    b. Build python code.

4. Experimental Investigations

The given dataset is available in image format, with the labels as seperate folders.
Training data will be imported using 'ImageDataGenerator' with FlowFromDirectory, so that the resultant training data makes model robust, by introducing variations in image for ex: horizonatl flips, vertical flips, shear, zoom etc.
Usually in real world, images which are generated will show different distribution, so before moving on to buid model, it's one responsibility to know "How images vary in shape?". Which Is demonstrated using jointplot in seaborn.

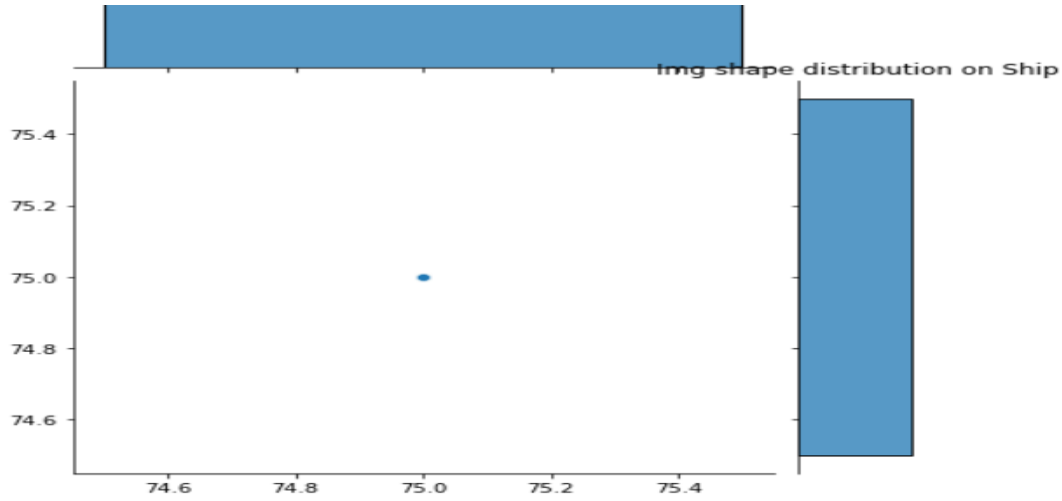
Iceberg Detection In Satellite Images

Iceberg:




```
Img Max: 1.0
Img Min: 0.0
dim1 mean: 75.0
dim2 mean: 75.0
Last Image's shape: (75, 75)
```

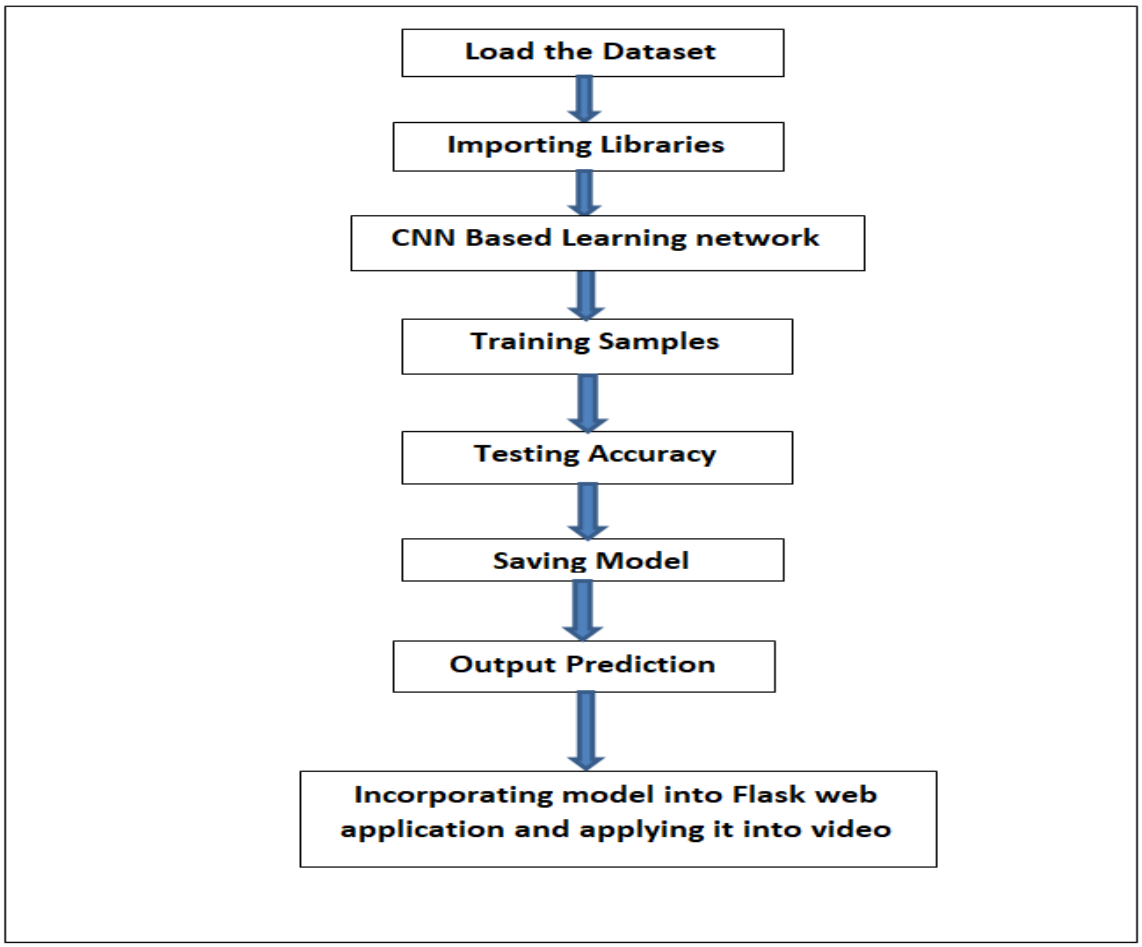
Ship:




```
Img Max: 1.0
Img Min: 0.0
dim1 mean: 75.0
dim2 mean: 75.0
Last Image's shape: (75, 75)
```

From the above distribution graph, we can see that all images with dimension 75 x 75 px.Now model can be built without introducing images into new shape or resize.Since images are loaded directly from directory using ImageDataGenerator through flowfromdirectory there is no explicit need for label, since taken implictly.Sequential model is used instead Functional, hence by saving model after training and optimizing.Later saved model will be tested, later incorported to build flask web application. At last built model is applied on video and results are generated on video.

5. FLOWCHART



Iceberg Detection In Satellite Images

6. Result

Based on the images in training dataset, model built and produced accuracy of 88%, which tells about capacity of model to predict there is iceberg or ship in given image or video . So this can be applied to real world scenarios.

7. Advantage

1. Prevents investing more on more technology instead AI.
2. By implementing it in real time satellite information broadcast, lot of lives can be saved.

Disadvantage

1. Model might have still more accurate.

8. Application

1. Real-time satellite information broadcast, for changing

route while travelling .

2. Implementing a visual sensor with this model in ships and boat (night vision enabled).

9. Conclusion

This work presents a technical application of recent progress of CNN based processing.

It will predict the presence of iceberg in water or not, so that ships can travel and reach destination without any problem in between, and that was final aim of this work.

10. Future Scope

1. The scope of Machine Learning is not limited to the investment sector. Rather, it is expanding across all fields such as banking and finance, information technology, media & entertainment, gaming, and the automotive industry.

2. Machine learning is merely based on predictions made based on experience. It enables machines to make data-driven decisions, which is more efficient than explicitly programming to carry out certain tasks.

11. BIBLIOGRAPHY AND APPENDIX

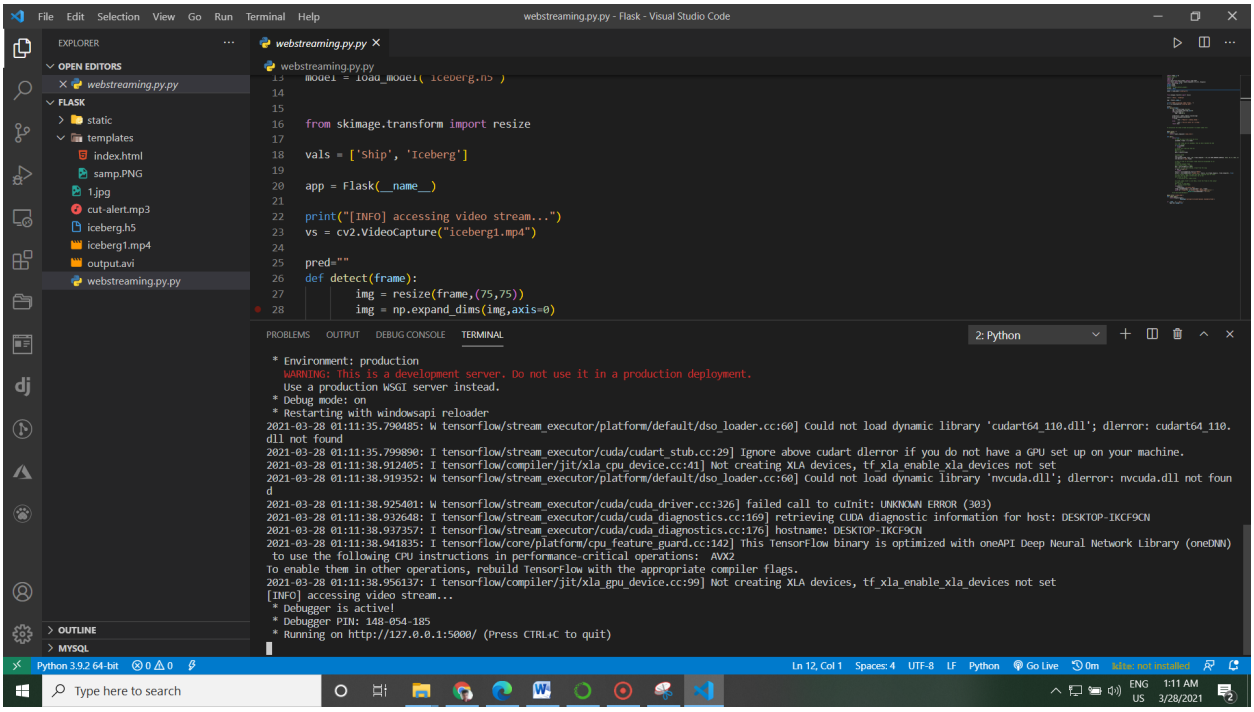
Iceberg Detection In Satellite Images

Model Building

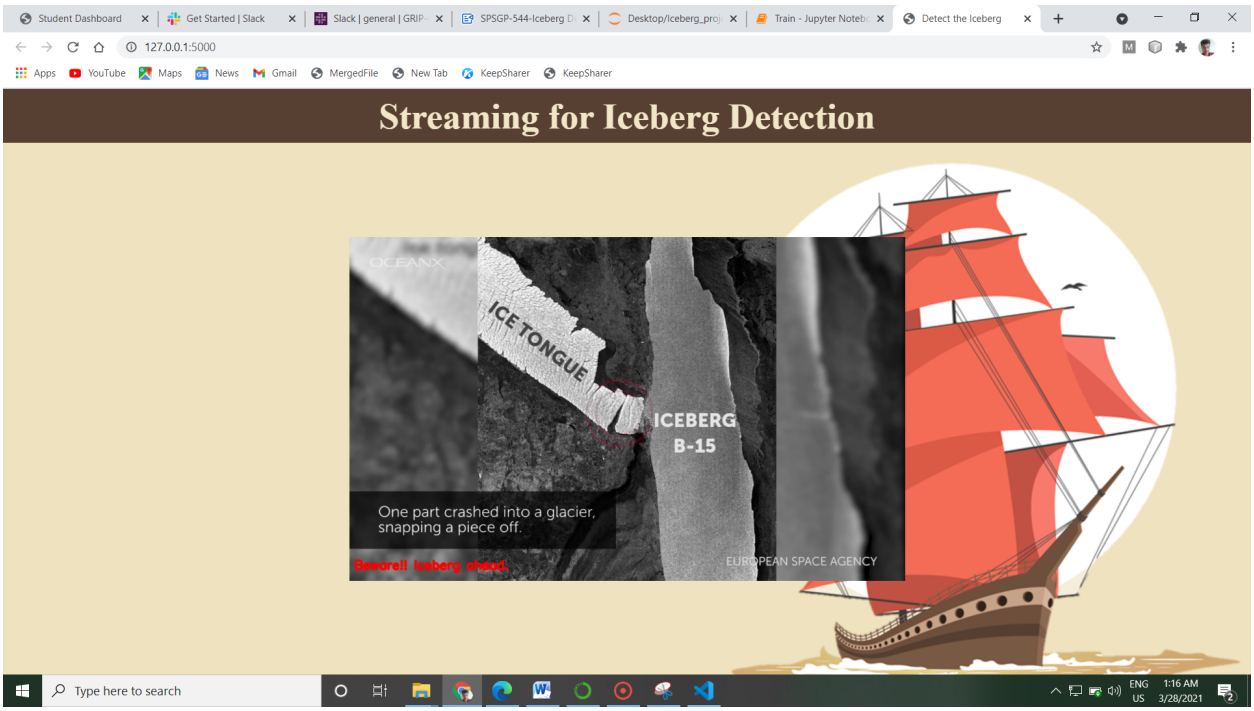
1. Dataset
2. Jupiter Notebook

Application Builiding

1. HTML file
2. CSS file
3. Flask
4. VS Code



Iceberg Detection In Satellite Images



A Source Code:

Iceberg Detection In Satellite Images

HTML CODE

```
<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;
```

Iceberg Detection In Satellite Images

```
                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-positio
n: 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>
```

Iceberg Detection In Satellite Images

</html>

Python Flask

```
import numpy as np

import imutils

import cv2

import os

from keras.models import load_model

from flask import Flask, render_template, url_for, Response

import tensorflow as tf

global graph

global writer

#graph = tf.get_default_graph()

writer = None


model = load_model('iceberg.h5')



from skimage.transform import resize


vals = ['Ship', 'Iceberg']
```

Iceberg Detection In Satellite Images

```
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


    prediction = model.predict_classes(img)
    pred=vals[prediction[0][0]]
    if pred:
        text = "Beware!! Iceberg ahead."
    else:
        text = "You are safe! It's a Ship."
    return text



# initialize the video stream and pointer to output video file



@app.route('/')
```

Iceberg Detection In Satellite Images

```
def index():
    return render_template('index.html')


def gen():
        while True:
            # read the next frame from the file
            (grabbed, frame) = vs.read()


            # if the frame was not grabbed, then we have reached the end
            # of the stream
            if not grabbed:
                break
            # resize the frame and then de
            #print(ix)
            #for x in vals:
            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)


            # check to see if the output frame should be displayed to our
            # screen
```

```
        # show the output frame
        #cv2.imshow("Frame", 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 an output video file path has been supplied and the video
        # writer has not been initialized, do so now
        #if writer is None:
            # initialize our video writer


        # if the video writer is not None, write the frame to the output
        # video file
        #if writer is not None:
        #    writer.write(frame)
        if(pred==1):
            playsound(r'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')
    #cv2.destroyAllWindows()
```

Iceberg Detection In Satellite Images

```
@app.route('/video_feed')
def video_feed():
    return Response(gen(),
                    mimetype='multipart/x-mixed-replace; boundary=frame')


if __name__ == '__main__':
    app.run( debug=True)
```