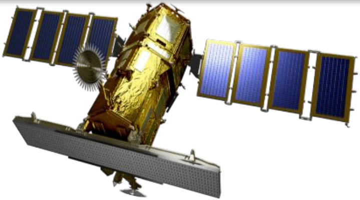
# DETECTING PRESENCE OF WATER USING SAR IMAGES



**08 JUNE 2021**

Department of CSE
Jyothi Engineering College Thrissur

# GROUP MEMBERS

**Presented By**

| | |
|---|---|
| Nair Anjali | [JEC17CS072] |
| Shibana | [JEC17CS092] |
| Najeeb VK | [JEC17CS068] |
| San Jose | [JEC17CS089] |

**Guided By**

Mr. Aneesh Chandran

# DEPARTMENT VISION AND MISSION

**VISION:**

Creating eminent and ethical leaders in the domain of computational sciences through quality professional education with a focus on holistic learning and excellence.

**MISSION:**

1. To create technically competent and ethically conscious graduates in the field of Computer Science and Engineering by encouraging holistic learning and excellence.
2. To prepare students for careers in Industry, Academia and the Government.
3. To instill Entrepreneurial orientation and research motivation among the students of the department.
4. To emerge as a leader in education in the region by encouraging teaching, learning, industry and societal connect.

# CONTENTS

# ABSTRACT

- Satellite imagery offers  a rich source of information
- There are two types of satellite imageries  Optical  & radar
- Recently research on SAR has been started
- Novel method for detecting presence of water
- Supervised deep learning is applied to map water regions

# INTRODUCTION

- Water - most important natural resources on the earth
- Observation of surface water- ecological and hydrological processes
- Surface water bodies are dynamic in nature
- The retrieved information used for water management tasks
- Overcome shortcoming of traditional ground based surveillance

# INTRODUCTION

- Traditionally optical data is used
- But no proper night vision is obtained
- SAR measurements are taken regardless of weather conditions and lighting conditions
- Active Sensors, successive pulse of MW are transmitted & echo of each pulse is recorded
- Bright areas- high backscatter , Darker areas –low backscatter

# INTRODUCTION

- **Motivation**

Implementing the technology of deep learning to efficiently detect the existence of surface water, to extract its extent, to quantify its volume, and to monitor its dynamics

- **Objective**

Introduce an autonomous water detection system using deep learning.

- **Problem Statement**

Surface water bodies are dynamic in nature ,Variations in this impact other natural resources, human assets & environment. Change usually causes serious consequences. In extreme cases, rapid increase of surface water can result in flooding & Landslides

# 1. Detecting, Extracting, and Monitoring Surface Water From Space Using Optical Sensors

## Features

- ➢ Reviews the current status of water
- ➢ Techniques - pixel unmixing, reconstruction, and spatio-temporal fusion

## Advantages

- ➢ Provides information for hydrological studies in ungauged areas.
- ➢ Increasing demand of monitoring global water dynamics

## Disadvantages

- ➢ Obscuration by clouds and vegetation.

# 2. Radar Satellite Imagery and Automatic Detection of Water Bodies

## Features

- Detection of water bodies based on histogram analysis and thresholding
- Radar signal wavelength and active satellite sensor is used

## Advantages

- Water bodies on the image are seen as dark regions.
- The process is fully automatic

## Disadvantages

- There is time delay

# 3. Automatic detection of surface-water bodies from Satellite images for effective mosquito larvae control

## Features

- Regular mapping of surface-water bodies in rice fields and wetlands
- Four methods were adapted and developed for automated application

## Advantages

- Uninterrupted data supply under any weather conditions
- Provide water detection at any time globally

## Disadvantages

- The accuracy is lower at rice-growing season

## 4. Water Detection using Satellite Images Obtained through Remote Sensing

### Features

- ➢ Single-band and multiband methods used for image processing
- ➢ Standard supervised maximum likelihood classification was used for the images.

### Advantages

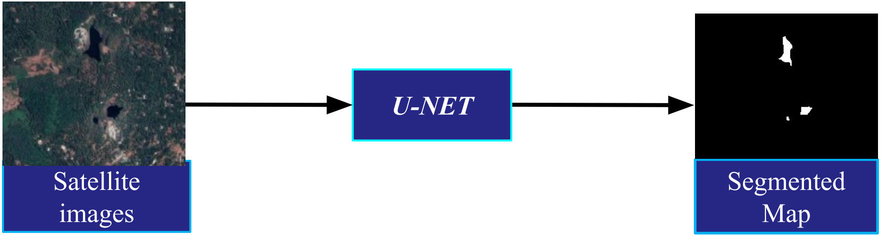
- ➢ Monitor the global water supply
- ➢ help in analyzing the changes in the earth areas in more rapid and easy manner.

### Disadvantages

- ➢ low accuracy rate

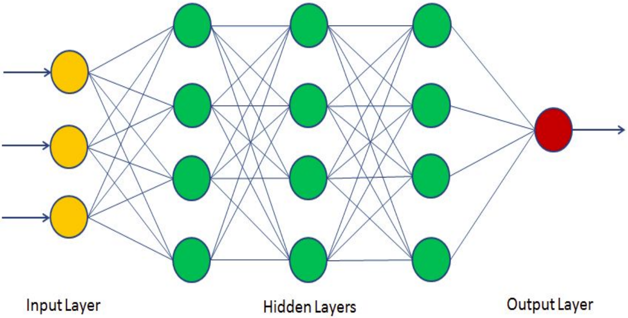
# PROPOSED SYSTEM

- Satellite Images are used to map water areas(Input)
- U-NET Architecture is used for semantic segmentation
- Corresponding masked image(output)



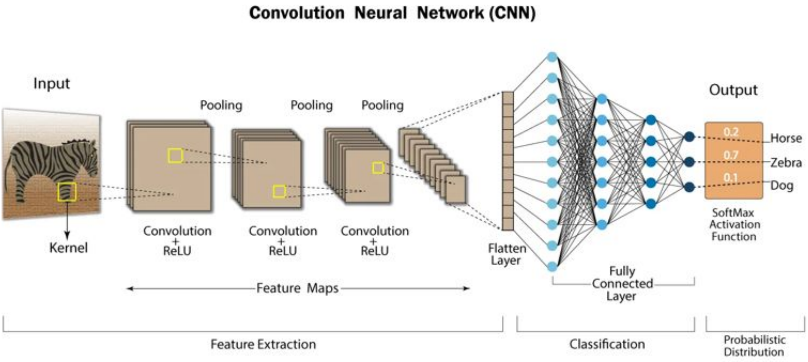
# Deep Learning

Deep learning an AI function, mimics the workings of the human brain



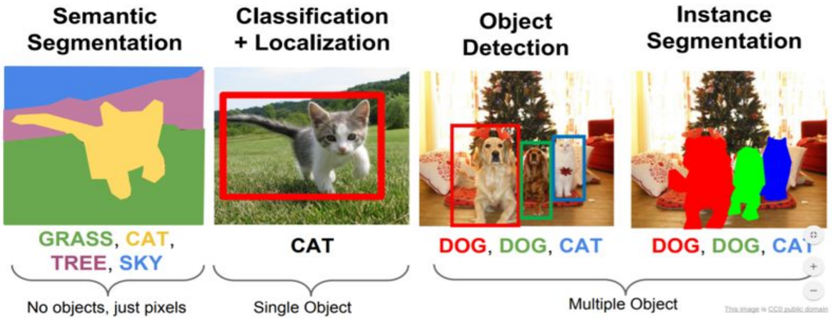
# Convolutional Neural Network

- **CNN** is a class of deep neural networks, most commonly applied to analyze visual imagery

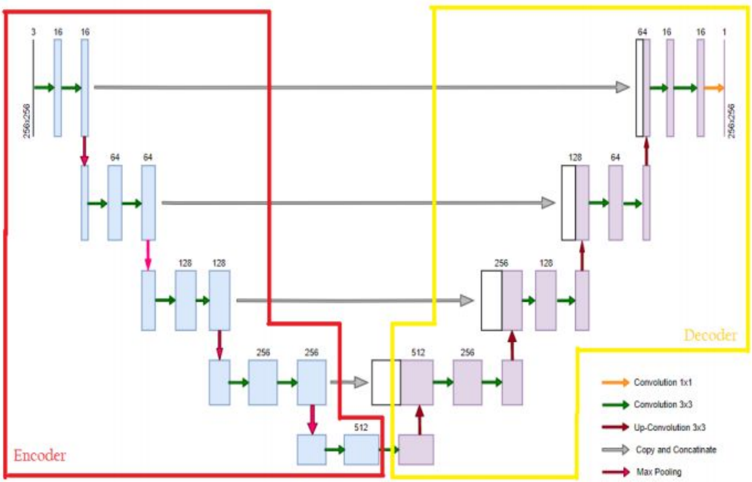


## Segmentation

Segmentation is the process of separating your data into distinct groups.

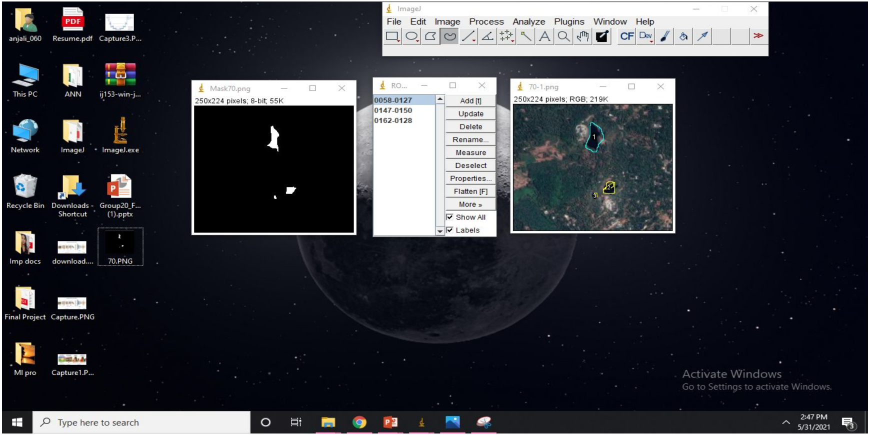


# U-Net


3
16
16
256x256
64
64
128
128
256
256
512
64
16
16
1
256x256
128
64
256
128
512
256
Decoder
Encoder
Convolution 1x1
Convolution 3x3
Up-Convolution 3x3
Copy and Concatinate
Max Pooling

# DATASET

**Source: Copernicus Open Access Hub at EU**



**Training DataSet:**

1. 66 training images(along with corresponding mask image)
   from this- 90% training
   10% validation

**Testing DataSet:**

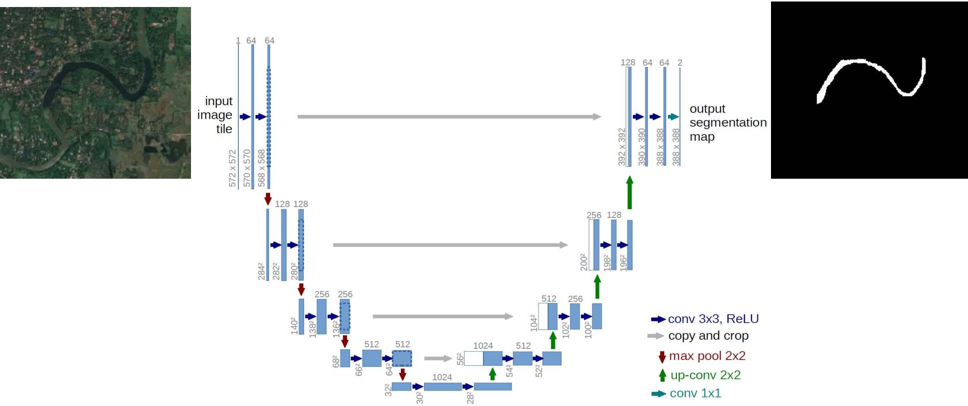
5 test images

**Libraries used:**

tensorflow, tqdm, numpy, scikit-learn, matplotlib

# IMPLEMENTATION

## Architectural Diagram



# IMPLEMENTATION

## ❖ **Steps Of Implementation/Code Overview:**

1. Import Modules
2. Declared image dimension: 128*128*3
3. Load & Store data
4. Resize every image to predefined value(both train & test)
5. Perform random check
6. Built the model

❖ **Steps Of Implementation/Code Overview:**

7. U-NET

❖ **Contraction**

1.convolution layer
2.dropout
3.2nd conv layer
4.Max pooling

❖ **Decoder**

1.conv Transpose
2. concatenate
3.Conv2D
4.Dropout
5.conv2d

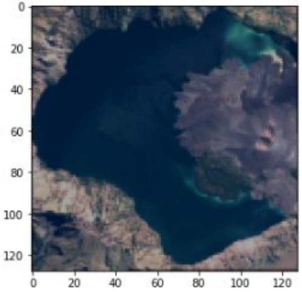
# IMPLEMENTATION

- ## <u>**Steps Of Implementation/Code Overview:**</u>

8. Create Output Layer
9. Model Summary
10. Save best model
11. Sanitary Check
12. Prediction
13. Obtain result

❖ **Steps Of Implementation/ code Overview :**

# Verify the images feeded:




# ❖ Testing & Results:

## Model Summary:

```
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            [(None, 128, 128, 3) 0
__________________________________________________________________________________________________
lambda (Lambda)                 (None, 128, 128, 3)  0           input_1[0][0]
__________________________________________________________________________________________________
conv2d_19 (Conv2D)              (None, 128, 128, 16) 448         lambda[0][0]
__________________________________________________________________________________________________
dropout_9 (Dropout)             (None, 128, 128, 16) 0           conv2d_19[0][0]
__________________________________________________________________________________________________
conv2d_20 (Conv2D)              (None, 128, 128, 16) 2320        dropout_9[0][0]
__________________________________________________________________________________________________
max_pooling2d_4 (MaxPooling2D)  (None, 64, 64, 16)   0           conv2d_20[0][0]
__________________________________________________________________________________________________
conv2d_21 (Conv2D)              (None, 64, 64, 32)   4640        max_pooling2d_4[0][0]
__________________________________________________________________________________________________
dropout_10 (Dropout)            (None, 64, 64, 32)   0           conv2d_21[0][0]
```

## ❖ Testing & Results:

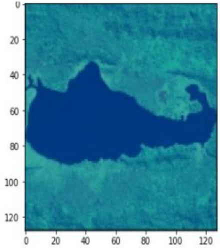
### Training:

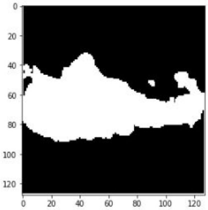
```
Epoch 1/25
4/4 [==============================] - 99s 21s/step - loss: 0.7059 - accuracy: 0.5112 - val_loss: 0.6911 - val_accuracy: 0.5385
Epoch 2/25
4/4 [==============================] - 45s 11s/step - loss: 0.6904 - accuracy: 0.5547 - val_loss: 0.6908 - val_accuracy: 0.6072
Epoch 3/25
4/4 [==============================] - 44s 11s/step - loss: 0.6916 - accuracy: 0.5504 - val_loss: 0.6909 - val_accuracy: 0.5845
Epoch 4/25
4/4 [==============================] - 44s 11s/step - loss: 0.6896 - accuracy: 0.5693 - val_loss: 0.6896 - val_accuracy: 0.5523
Epoch 5/25
4/4 [==============================] - 44s 11s/step - loss: 0.6868 - accuracy: 0.5783 - val_loss: 0.6871 - val_accuracy: 0.4983
Epoch 6/25
4/4 [==============================] - 45s 11s/step - loss: 0.6824 - accuracy: 0.5769 - val_loss: 0.6844 - val_accuracy: 0.4861
Epoch 7/25
4/4 [==============================] - 46s 11s/step - loss: 0.6750 - accuracy: 0.5743 - val_loss: 0.6824 - val_accuracy: 0.4855
Epoch 8/25
4/4 [==============================] - 52s 13s/step - loss: 0.6647 - accuracy: 0.5719 - val_loss: 0.6677 - val_accuracy: 0.4862
Epoch 9/25
4/4 [==============================] - 52s 13s/step - loss: 0.6443 - accuracy: 0.5855 - val_loss: 0.6132 - val_accuracy: 0.6091
Epoch 10/25
4/4 [==============================] - 56s 14s/step - loss: 0.6168 - accuracy: 0.6026 - val_loss: 0.5632 - val_accuracy: 0.7412
Epoch 11/25
4/4 [==============================] - 64s 16s/step - loss: 0.5768 - accuracy: 0.6603 - val_loss: 0.5078 - val_accuracy: 0.7916
Epoch 12/25
4/4 [==============================] - 54s 13s/step - loss: 0.5459 - accuracy: 0.6864 - val_loss: 0.4735 - val_accuracy: 0.8001
Epoch 13/25
4/4 [==============================] - 49s 12s/step - loss: 0.5693 - accuracy: 0.6664 - val_loss: 0.5487 - val_accuracy: 0.7221
Epoch 14/25
4/4 [==============================] - 51s 13s/step - loss: 0.5522 - accuracy: 0.6807 - val_loss: 0.4816 - val_accuracy: 0.7842
```

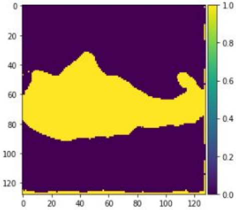
# IMPLEMENTATION

❖ **Testing & Results:**

**Random test of Validation:**

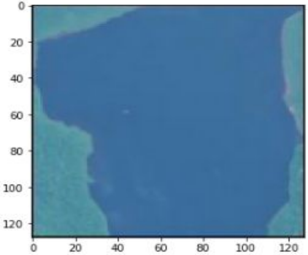


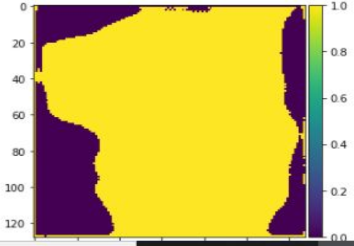



## ❖ Testing & Results:

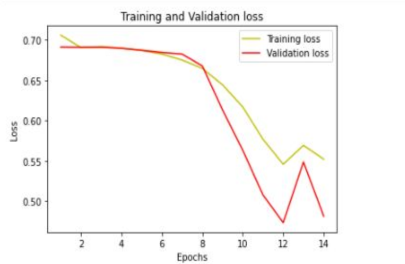
**Prediction:**

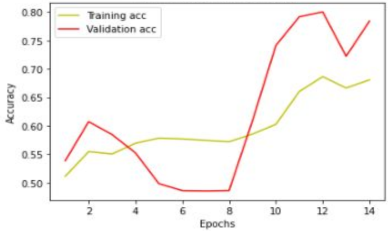


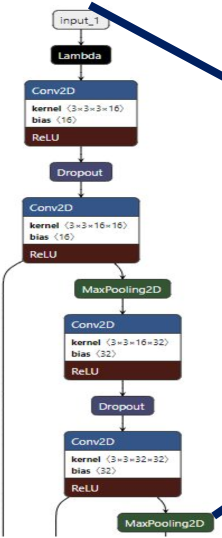
## ❖ Testing & Results:

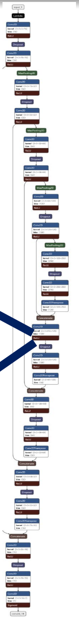
## Loss & Accuracy Graph:



❖ **Testing & Results: Saved Model:**



## Conclusion

- Presented a model for detecting the presence of water
- U-net architecture is used for  segmentation
- The purpose is to make fast algorithms for water detection using Radar satellite images.

## Future Enhancement

- detect flood/water inundation using the obtained dynamic of water areas.
- detect various model like military vehicles etc

# REFERENCES


1. Huang, Chang, et al. "Detecting, extracting, and monitoring surface water from space using optical sensors: A review." *Reviews of Geophysics* 56.2 (2018): 333-360.
2. Čotar, Klemen, Krištof Oštir, and Žiga Kokalj. "Radar satellite imagery and automatic detection of water bodies." *Geodetski glasnik* 50.47 (2016): 5-15.
3. Rana, Himanshu, and Nirvair Neeru. "Water detection using satellite images obtained through remote sensing." *Adv. Comput. Sci. Technol* 10 (2017): 1923-1940.
4. Ovakoglou, Georgios, et al. "Automatic detection of surface-water bodies from Sentinel-1 images for effective mosquito larvae control." *Journal of Applied Remote Sensing* 15.1 (2021): 014507.
5. Kreiser, Zachary, Brian Killough, and Syed R. Rizvi. "Water across synthetic aperture radar data (wasard): Sar water body classification for the open data cube." *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018.
6. Bartsch, A., et al. "Detection of open water dynamics with ENVISAT ASAR in support of land surface modelling at high latitudes." *Biogeosciences* 9.2 (2012): 703-714.

# THANKYOU