Computer Vision

Deepak Subramani

Assistant Professor

Dept. of Computational and Data Science

Indian Institute of Science Bengaluru

Deepak Subramani, deepakns@iisc.ac.in

Lecture and Assignment Guide

- This Slide Deck has Material for 6 hours of teaching divided into Parts 1-6
- We will go through
  - Week 01
    - Part 01 - Convolutional and Pooling Layers; AST 01
    - Part 02 - Transfer Learning and Modern CV Design Principle; AST 02
  - Week 02
    - Part 01 - Modern Convolutional Building Blocks for Image Classification; AST 03
    - Part 02 - Object Localization
    - Interpreting what convolutions learn (Advanced topic) – AST 03
  - Week 03
    - Part 01 - Object Detection (YOLO), Image Segmentation – Lec 05
    - Part 02 - Practical CVOps
    - AST04 – Object Detection with YOLO
  - Week 04
    - Revision
    - AST05 – Image Segmentation
- Additional Reading material to go in depth of math with references and code references are provided with the marking of "Additional Material" or "Additional Discussion" etc
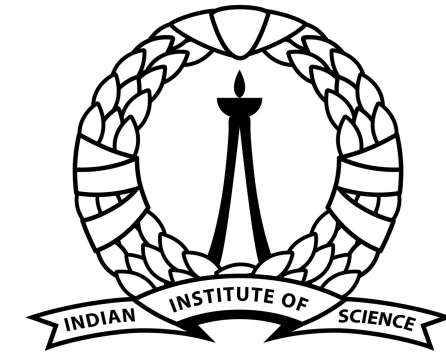
Week 03 – Part 01

Deepak Subramani

Assistant Professor

Dept. of Computational and Data Science

Indian Institute of Science Bengaluru

Object Detection

- Object Detection: The task of simultaneously classifying and localizing multiple objects in an image

- Earlier approaches involved sliding a single object detector CNN over an image and use post processing

- Steps
  - Slide bounding box CNN over the image
  - Find bounding boxes
  - Post process
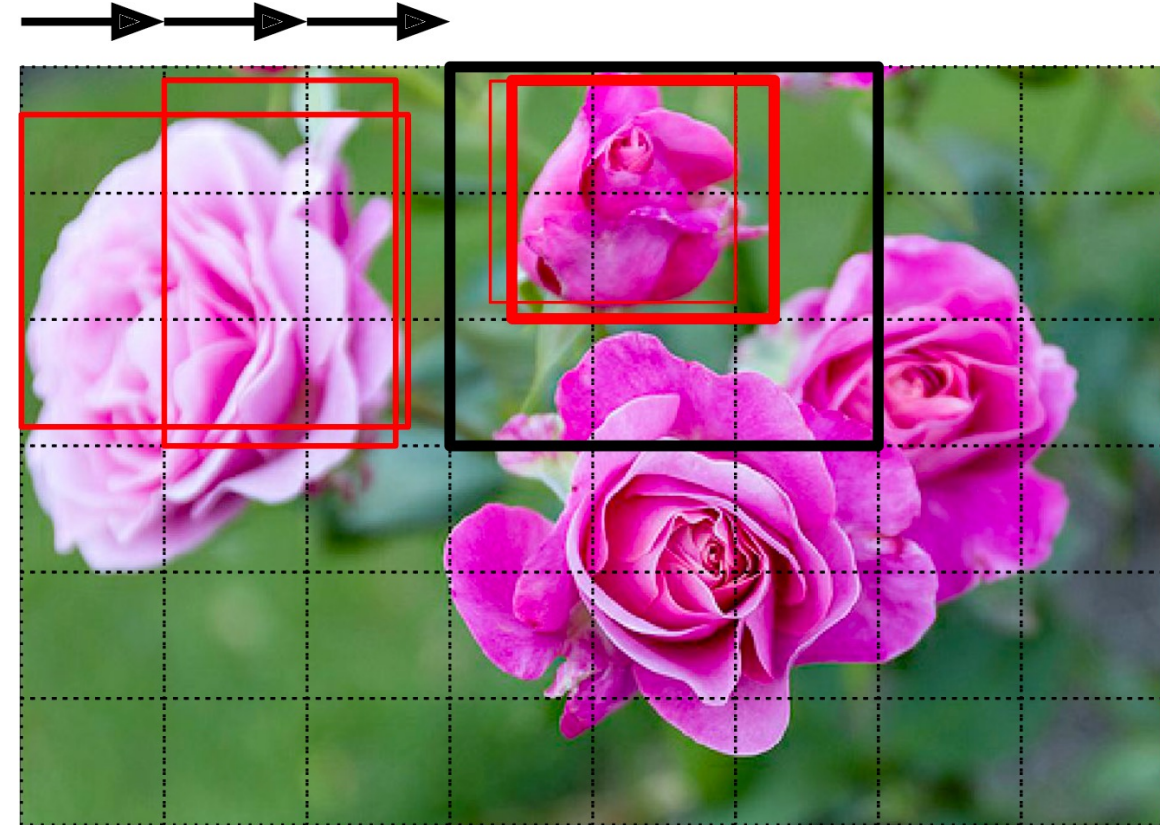
Object Detection

- Steps
  - Divide the image into grids
  - Slide an object detector over each 3x3 region
  - Predict one bounding box per 3x3 region
  - Repeat for 4x4 etc
  - In addition to a bounding box, output "objectness" – the probability of the object being present in the bounding box
    - Sigmoid activation trained with binary CE
  - Postprocess to get rid of overlapping bounding boxes
    - Drop BBox with low objectness
    - Find and drop BBox that overlap with the BBox with highest objectness
    - Repeat until no more boxes can be removed
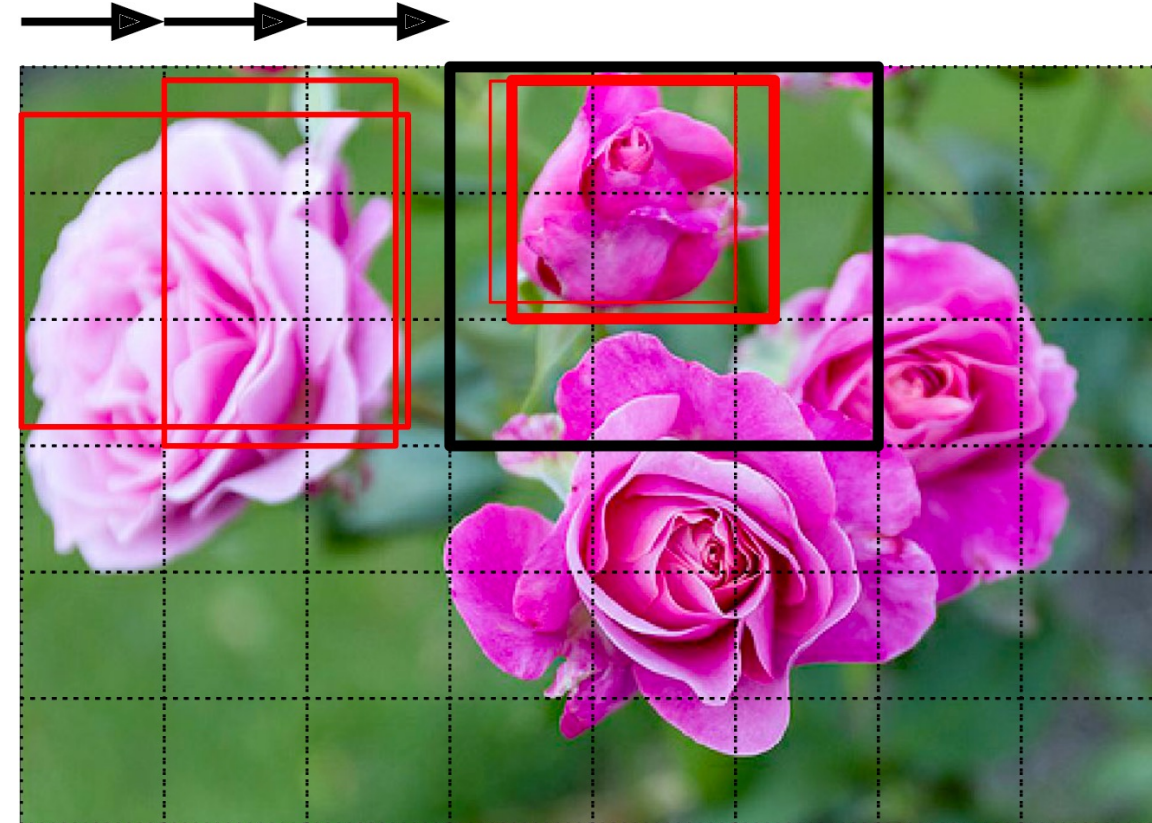
Fully Convolutional Network

- The simple approach in the previous slides are powerful, but slow

- We need to run the CNN many times over the image

- FCN (Fully Convolutional Network) solves this problem

- Key idea: Replace the top dense layers with convolutional layers

- Example – Conv Base C10 followed by D1
  - C10 – 100x7x7 [100 maps each of 7x7]
  - D1 – 200 neurons – Each neuron has a connection to 100x7x7 from C10 + bias
  - Replace D1 with C11 – 200 convolutional filters 7x7+1(V) – Output is 1x1x200
  - This C11 is doing the same job as D1 but using convolutions now
    - V – Valid Padding

FCN: What is the big deal?

- Dense Layer expects a specific input size
  - In this example: It is 100x7x7
  - But a Convolution will work even on 100x14x14 and anything else!
    - It only needs that 100 to be intact

- Let us look at the numbers
  - Trained CNN for flower object detection with input size 224x224
    - 10 outputs on the Dense Layer head – 0-4 (softmax) 1 per class [5 classes], 5 (logistic) objectness score, 6-9 (none) x coordinate, y coordinate, height, width of bounding box
    - Suppose the Conv Base ends at 7x7
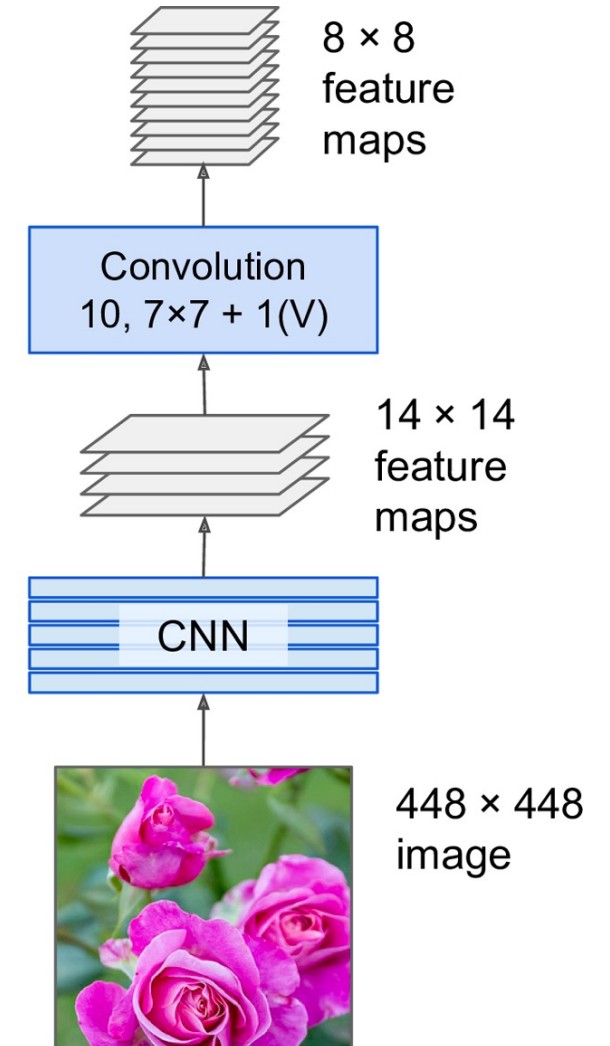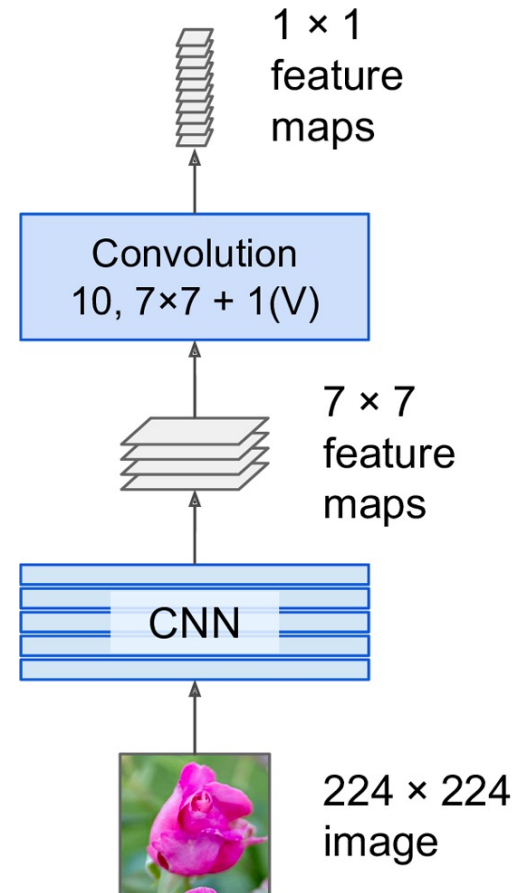   - Replace Dense head with Conv Layer 10 filters of 7x7+1(V)

Poll

1. A Fully Convolutional Network is a more efficient network for object detection
    - True, False

2. If a dense layer is replaced with a FCN after training, then the weights can be copied from Dense to the FCN
    - True, False

3. IOU is the typical loss function used for object detection
    - True, False

FCN Example (Cont)

- What happens if we feed 448x448 images to this FCN?
  - Last conv layer is 14x14, and it will produce a 8x8 map
  - What is this 8x8 map? – It is equivalent to sliding the original CNN across the image
- Now the network has to be run only once
- You Only Look Once (YOLO)

1 × 1 feature maps

Convolution 10, 7×7 + 1(V)

7 × 7 feature maps

CNN

224 × 224 image

8 × 8 feature maps

Convolution 10, 7×7 + 1(V)

14 × 14 feature maps

CNN

448 × 448 image

YOLO: Introduction

- YOLO (and its variants) – Introduced between 2015-18
- State of the art for real time object detection
- Demo of YOLO V3 - https://www.youtube.com/watch?v=MPU2Histivl
- Trained on the PASCAL (Pattern Analysis, Statistical Modeling and Computational Learning) VOC (Visual Object Class) Dataset
- PASCAL VOC challenge is another machine vision challenge for object detection and segmentation
    - Recall: Image net for object classification

YOLO: Basic Idea

- Similar to the algorithm we discussed so far, but with a few engineering modifications – No new concept

- Outputs five bounding boxes (instead of 1) and 20 class probabilities (PASCAL VOC dataset)

- Before training, a K-Means clustering is applied to the training dataset bounding boxes and the centroid of 5 representative bounding boxes are found. This serves as the BBox prior, called as anchor boxes

- The network then predicts the factor by which the anchor box must be resized

- During training, new image dimensions are randomly chosen so that the network learns to deal with different scales

Bounding box: x, y, width, height, confidence, class
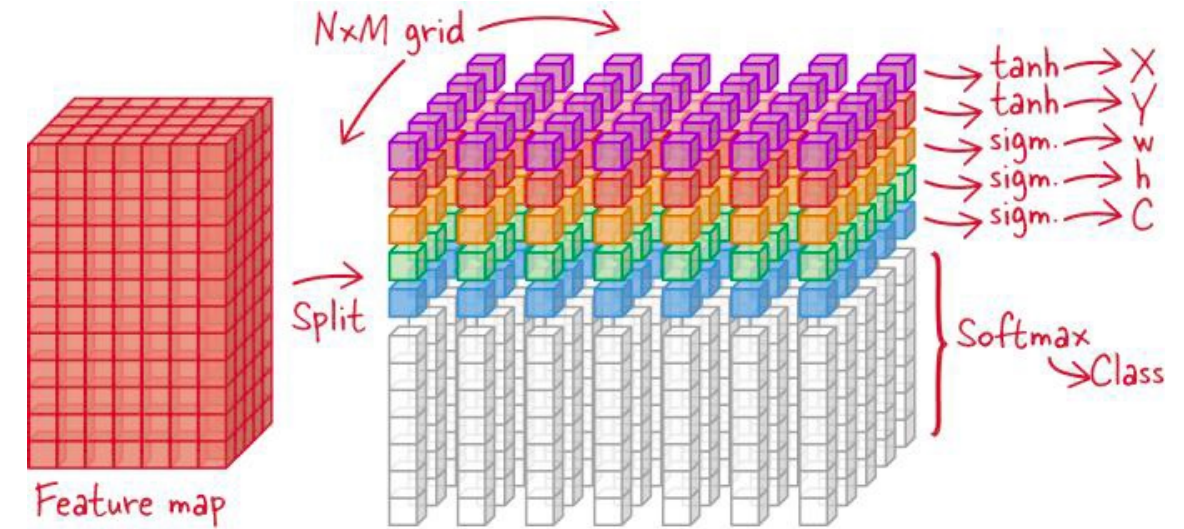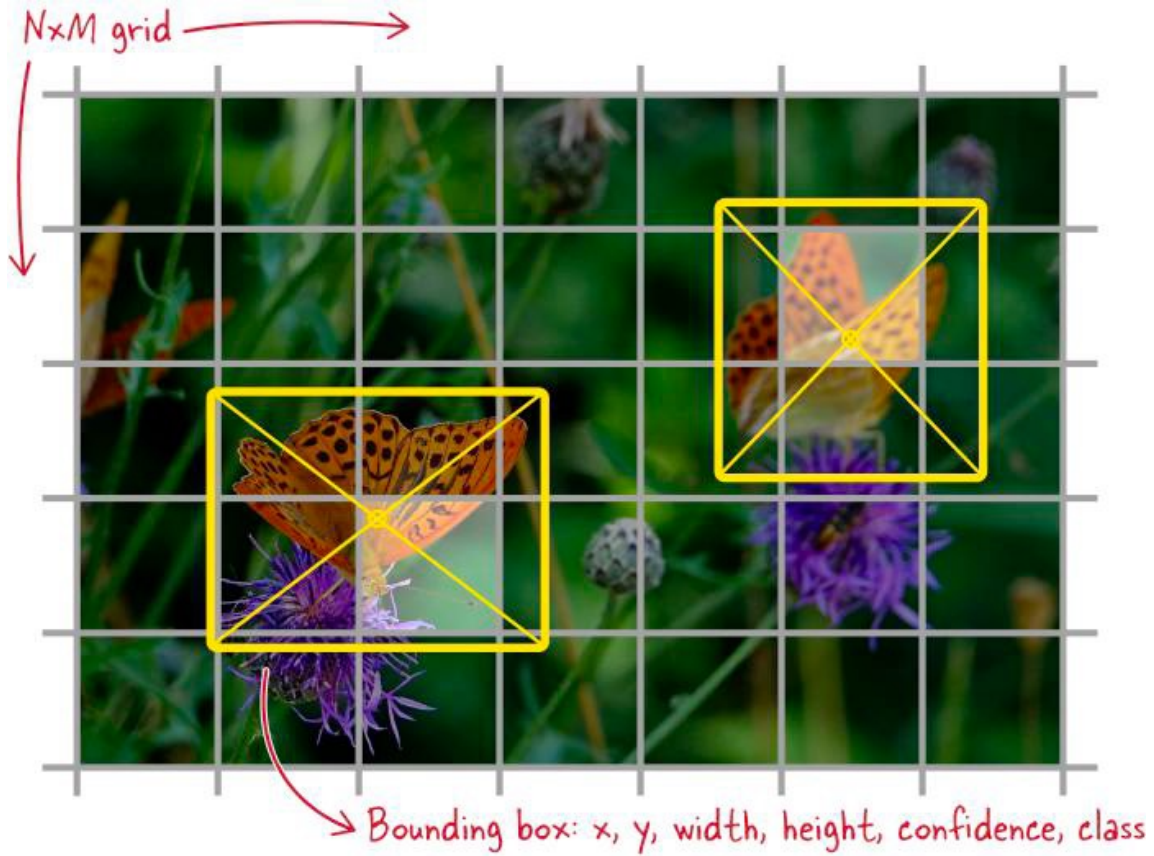
Evaluation Metrics

- Object detection is more complex to evaluate than image classification

- We use the following definitions

  - **True Positive (TP)**: A correct detection. Detection with IOU ≥ *threshold*
  - **False Positive (FP)**: A wrong detection. Detection with IOU < *threshold*
  - **False Negative (FN)**: A ground truth not detected
  - **True Negative (TN)**: Does not apply. It would represent a corrected misdetection. In the object detection task there are many possible bounding boxes that should not be detected within an image. Thus, TN would be all possible bounding boxes that were corrrectly not detected (so many possible boxes within an image). That's why it is not used by the metrics.

  threshold: depending on the metric, it is usually set to 50%, 75% or 95%.
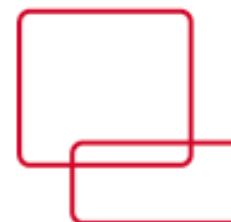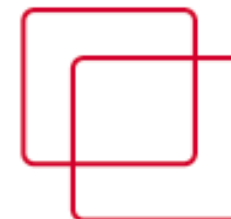
IoU

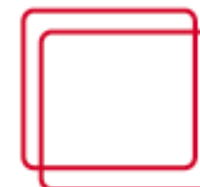$$\text{IOU} = \frac{\text{Area of intersection}}{\text{Area of union}}$$

Examples

IOU = 0.1

IOU = 0.3

IOU = 0.6

Accuracy of Object Detection System: mAP

- Metric for Object Detection Accuracy: Mean Average Precision

- Why mean average?

- Precision-Recall Tradeoff
  - Precision=TP/(TP+FP)=TP/(all detections)
  - Recall=TP/(TP+FN)=TP/(all ground truths)
  - Higher the precision, lower the recall.
  - But sometimes there is no tradeoff, especially at low ends. This leads to mAP

- Sometimes 90% precision at 10% recall and 96% precision at 20% recall. Then we should use the 96,20 operating point

Accuracy of Object Detection System: mAP

- To generalize the situation we should look at the maximum precision with atleast some recall (say 20%)

- We keep changing this recall threshold 0-100 and collect the maximum precision possible. Then report the average. This is AP

- If we have more classes, then we have AP for each class. So we take mean AP

- In object detection there is an added complexity – accuracy depends on the IoU also.

- So we report mAP@IoU. We also take mean(mAP@[IoU range]) – mean mean average precision!

- Example calculation: https://github.com/rafaelpadilla/Object-Detection-Metrics
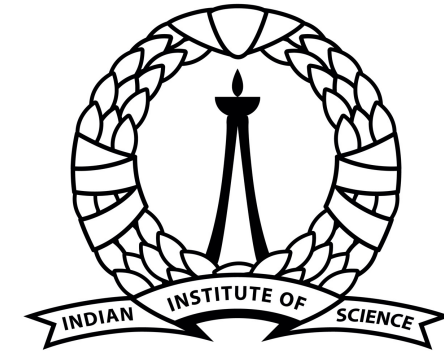
# Week 03 Part 02

Deepak Subramani

Assistant Professor

Dept. of Computational and Data Science

Indian Institute of Science Bengaluru

Segmentation

- Semantic Segmentation: Each pixel is classified and given a class label
- Different objects of the same class are not distinguished
  - Two people standing together will be identified as one big lump of people!
- Instance Segmentation: Each object is distinguished from the other
- The most popular architecture is called Deep Lab (current SOTA)
- U-Net is a popular architecture used in Biomedical Image segmentation and recently for Satellite Image Analysis
- Two concepts to be learnt here
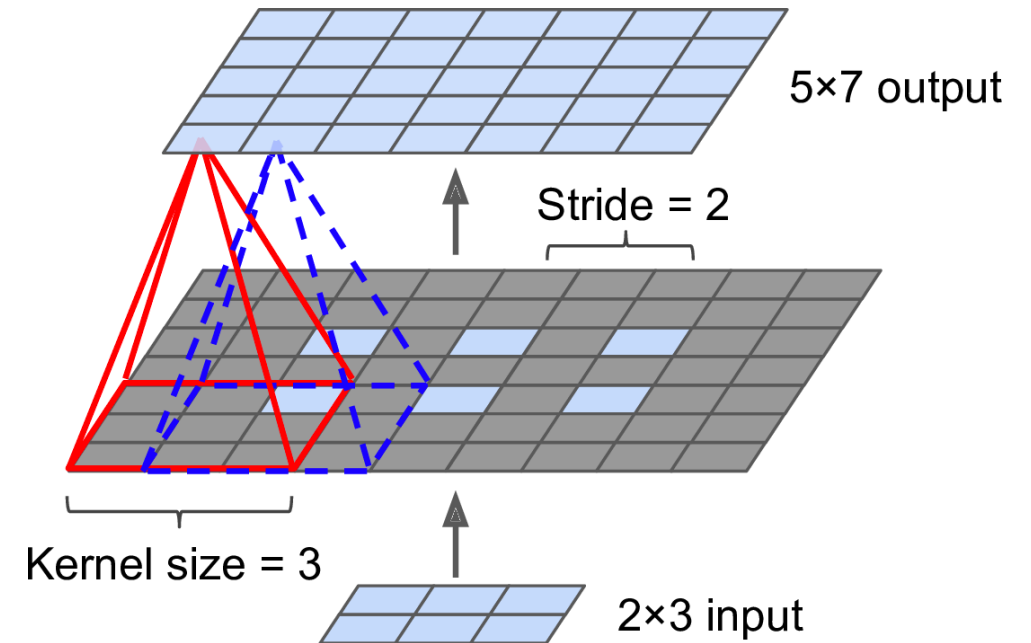 - Transposed Convolution, with skip connection
  - Atrous Convolution
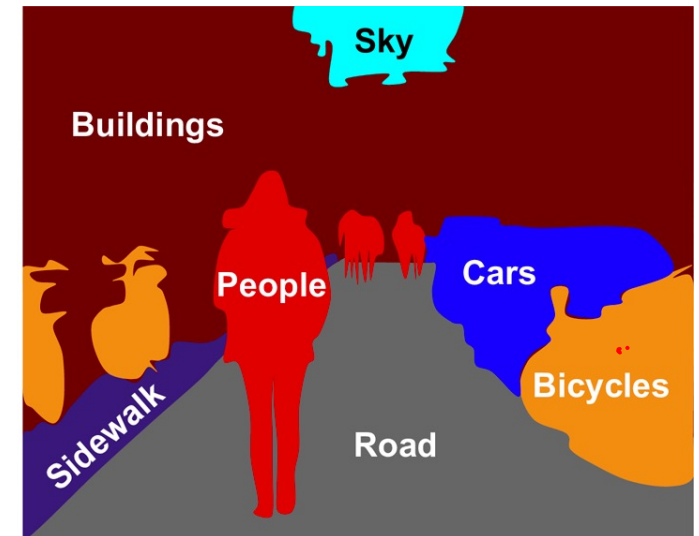
Transposed Convolution: Motivation

- To learn multiscale features efficiently, stride>2 is necessary in the convolutional base

- But, then how to classify each pixel?

- This needs an upsampling layer

- The idea of Transposed Convolution comes from this need

- The simplest way is a bilinear interpolation, but that works only for 4x or even 8x, but not beyond

- Long et al 2015, introduced this Transpose Convolution, in addition to the FCN that we saw earlier

Transposed Convolution

- Think of stretching an image by adding empty rows and columns

- Then on the stretched image do a regular convolution

- Initialize these kernels to do a linear interpolation

- But as the weights are learnable, it does better!

5×7 output

Stride = 2

Kernel size = 3

2×3 input

Skip Connection

- The transpose convolution introduces difficulties in gradient flow
- Skip connections are introduced to make the down sampling kernels learn
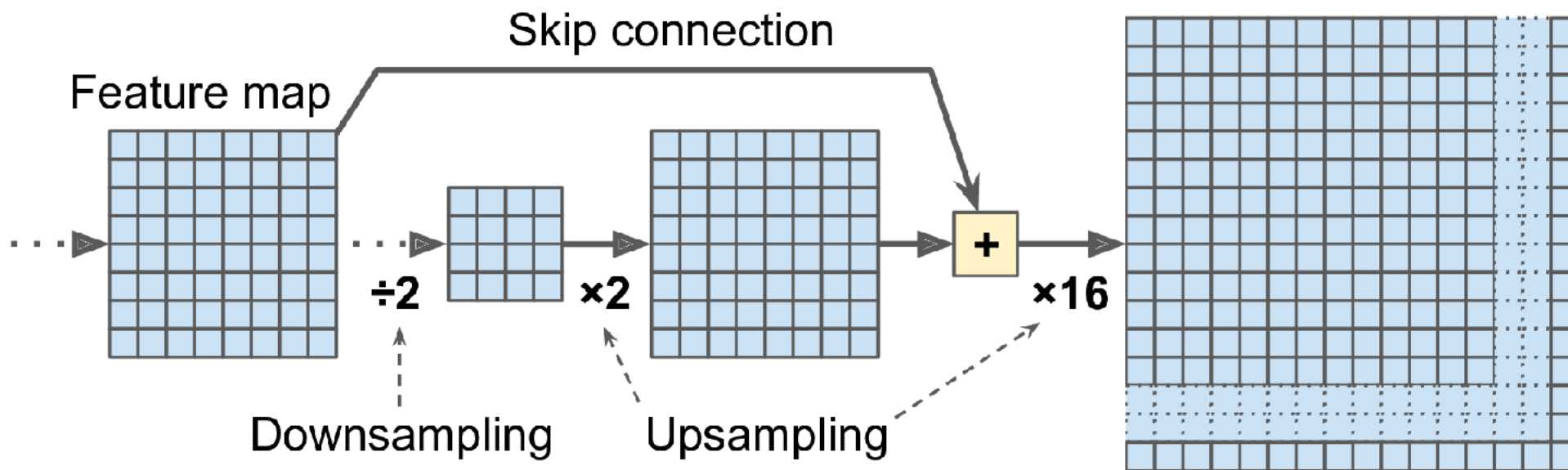
Atrous Convolution

- The convolution field of view is modified by considering a larger area with zeros added to the filter itself

- Number of learnable parameters is the same as regular convolution, but now the field of view has changed
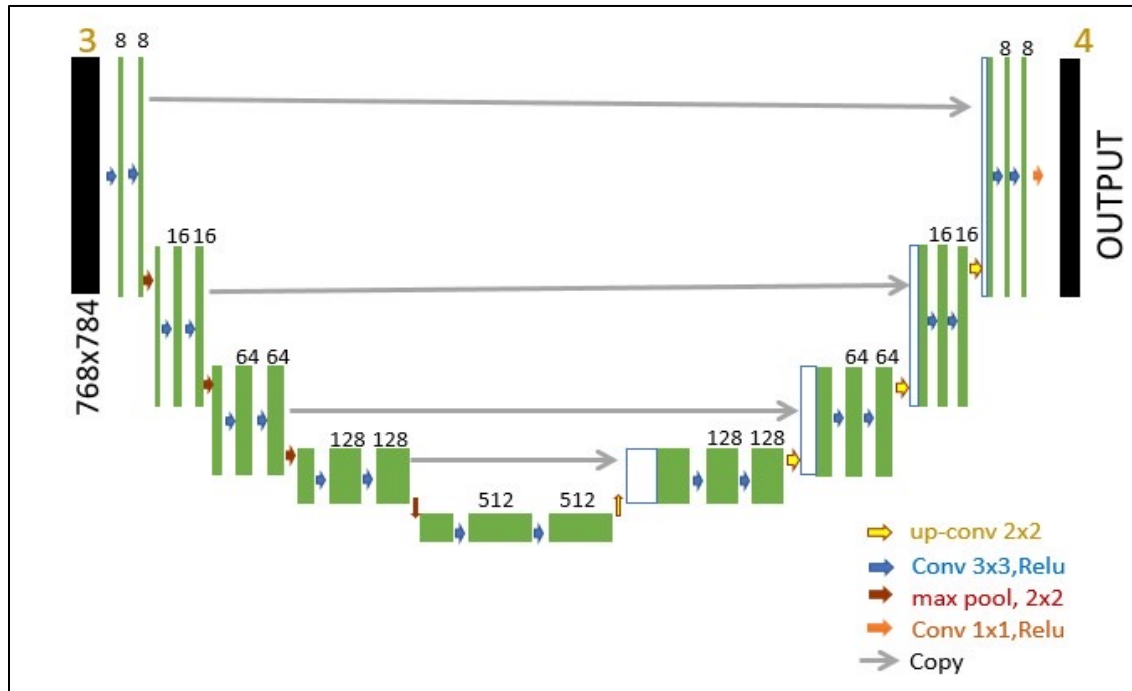
- This is used in Deep Lab
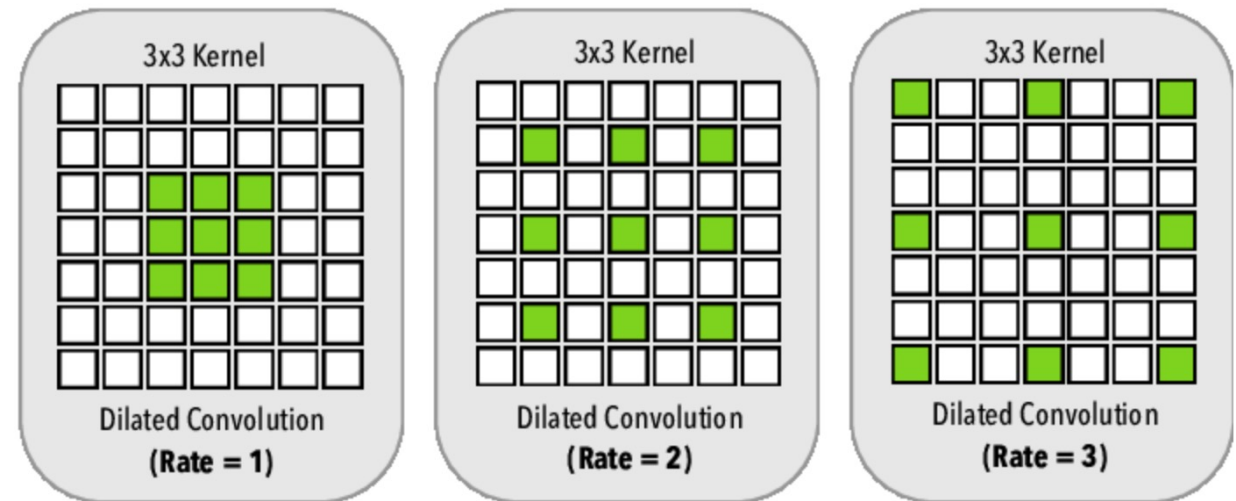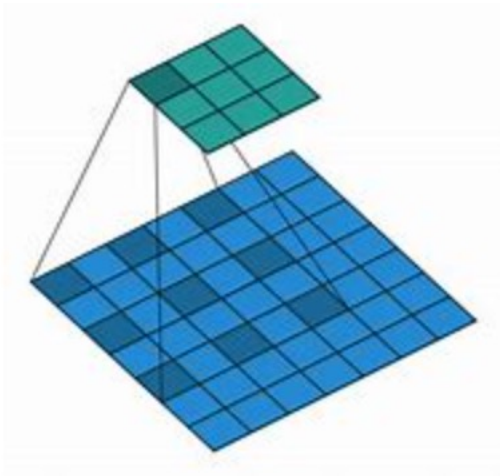
Poll

1. In semantic segmentation, two objects of the same class are distinguished
   - True, False

2. FCN is used for semantic segmentation
   - True, False

Biomedical Image Segmentation – U nets

U-net architecture by Ronneberger and Olaf. 2015.

- Error/Loss is estimated by comparing target and output of U-net.
- Number of maps in output are equal to number of classes in the problem.

State of the Art as of Oct 2023

- https://paperswithcode.com/area/computer-vision
- Image Classification –
  - For speed: Efficient Net (CNN),
  - For accuracy: Vision Transformer
- Semantic Segmentation –
  - For speed: Unet, DeepLabv3
  - For accuracy: Deformable Convolution (InternImage), Vision Transformer (Segment Anything)
- Instance Segmentation - Mask-R-CNN, RetinaNet
- Object Detection –
  - For speed: YOLOv8
  - For accuracy: Deformable Convolution (InternImage)

Other Tasks in CV

- Pose Estimation

- Object Tracking

- Action Recognition

- Motion Estimation

- Monocular Depth

- Content-aware Image Editing

- Scene Reconstruction (NeRF)

Creating Vision Datasets

- Collecting images can be done in multiple ways
    - Mounting a camera at the end of an almond processing line, traffic intersection etc
    - Extracting photographs from a digital catalog
    - Purchasing an archive of satellite imagery
    - Medical image devices

- Capturing photographs
    - Use the highest resolution that is necessary for the noise characteristics of your image and what your ML infra budget can handle
        - Higher-resolution outdoor images in low light have higher noise
        - Collecting high-res images needs a lot of time and bandwidth

Proof of Concept

- In many situations, you may not have the data on hand, and collecting it for a PoC would take too long. What to do?

- Purchase similar data to understand the feasibility of a project before investing in routine data collection.

- When purchasing images,
  - acquire images that are similar in quality, resolution, etc. to the images that you will ultimately be able to use in the actual project.

- Simulate labeled images by modifying existing images
  - In advanced applications of crowd counting etc

Labeling Data

- For image classification there are two manual labeling approaches
  - Move the images to a folder whose name is the label
  - Create an excel file (spreadsheet) with first column having the path of the image and the other columns having the label(s)

- Object detection
  - Needs bounding box (usually counterclockwise starting from top-left)

- Segmentation
  - Needs labels of pixels

Labeling at scale

- OpenCV Computer Vision Annotation Tool
  - https://github.com/opencv/cvat
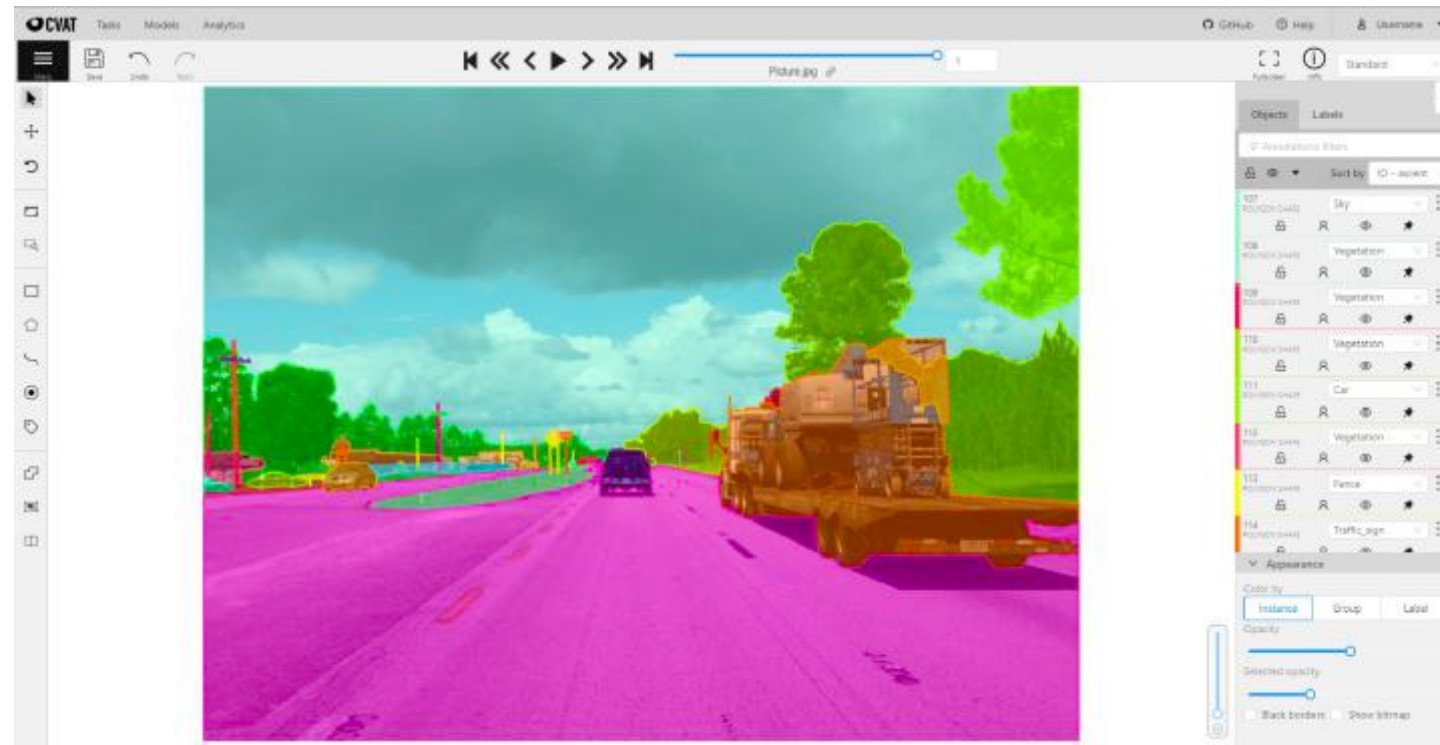
Other Techniques

- Crowdsourcing
- AI Labeling Services - https://cloud.google.com/vertex-ai/pricing#labeling
- Noisy student
  - Manually label, say, 10,000 images.
  - Use these images to train a small CV model. This is the teacher model.
  - Use this model to predict the labels of, say, one million unlabeled images.
  - Train a larger CV model, called the student model, on the combination of labelled and pseudo-labelled images.
  - During the learning of the student model, employ dropout and random data augmentations so that this model generalizes better than the teacher.
  - Iterate by putting the student model back as the teacher.
  - Manually correct the pseudo-labels by choosing images where the models are not confident.

Additional Material

- Feature Pyramid Network

- Satellite Image Analytics Research

Deepak Subramani, deepakns@iisc.ac.in

Feature Pyramid Network

- In CNN, initial layers learn low level semantic information

- Later layers learn high level semantic information

- YOLO uses only the last feature map for detection

- YOLO correctly detects but localization accuracy is limited
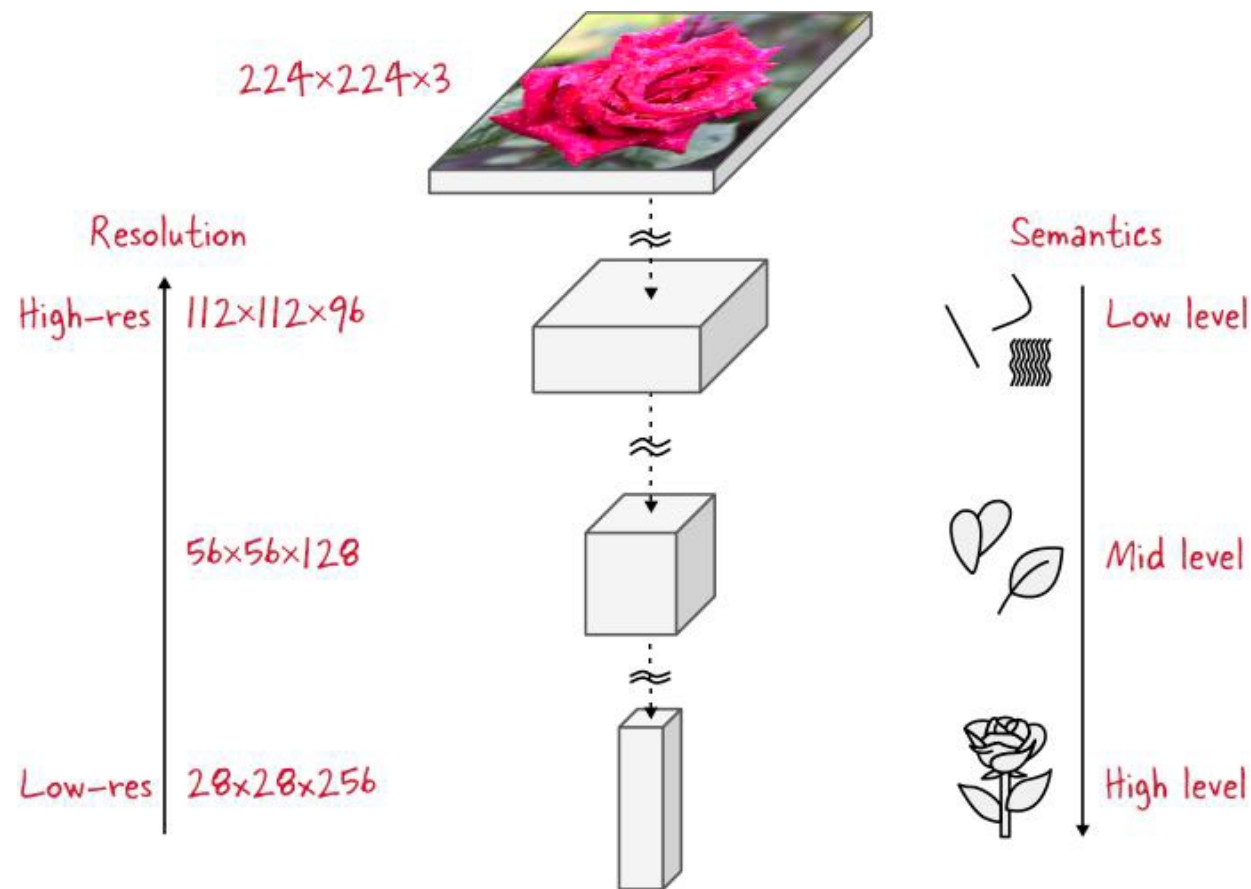
Feature Pyramid Network

- If a detection head is placed at the end of every convolutional block, then localization would be good, but labelling will be difficult

- Single Shot Detector tried the detection head at the end of the last convolutional blocks

- Feature pyramid networks add a detection head and combine them together through couple of additional layers
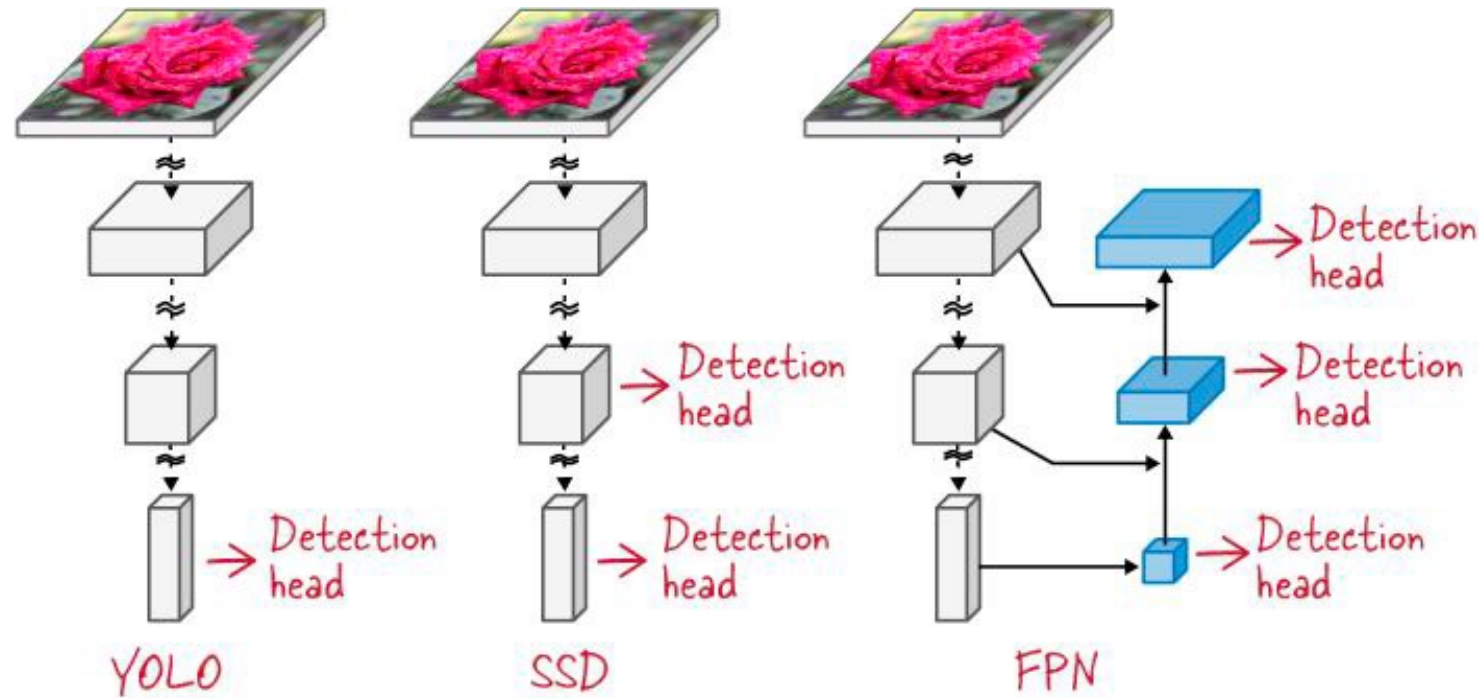
Deepak Subramani, deepakns@iisc.ac.in

Problem Statement
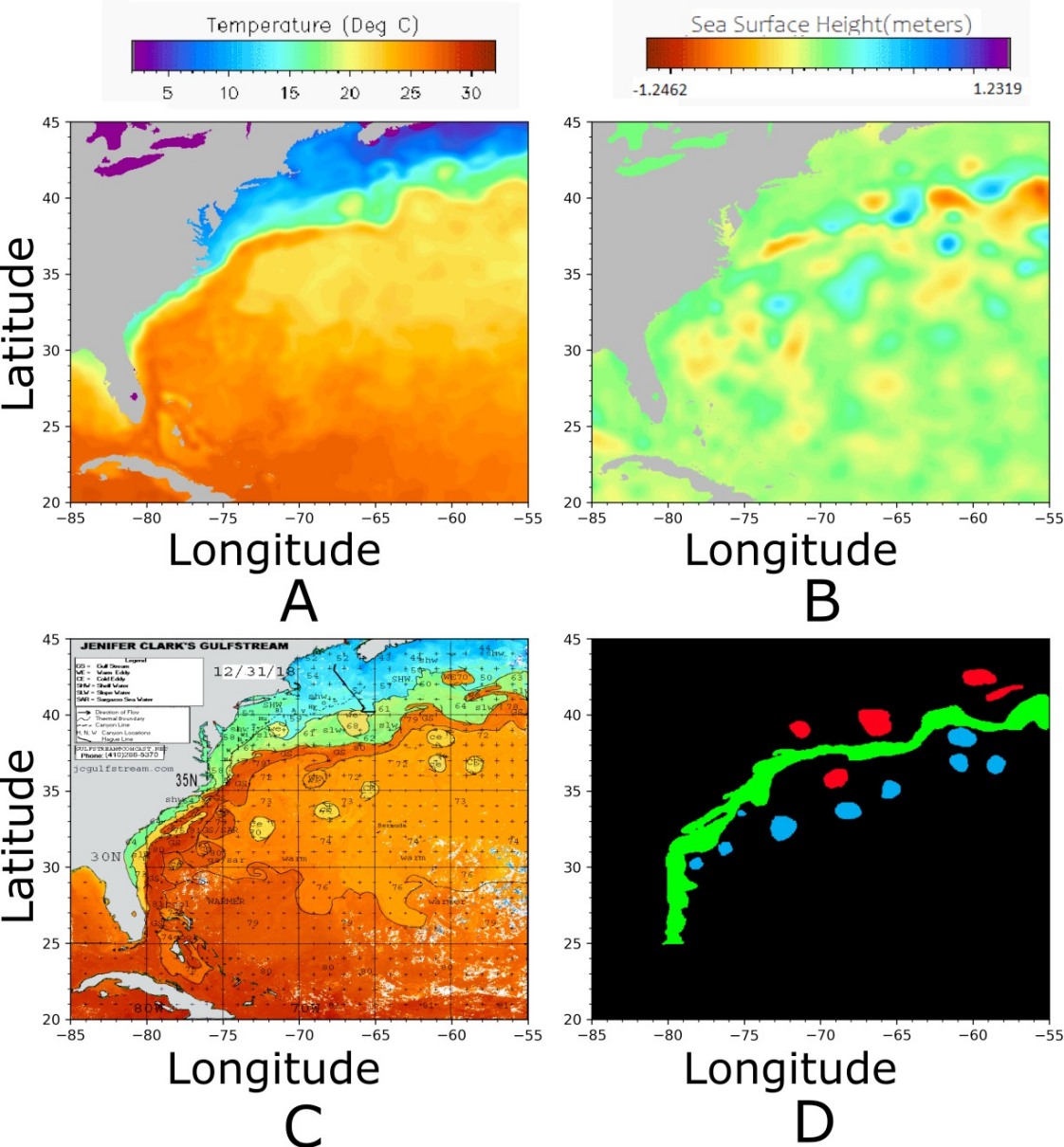
A

B

C

D

W-Net Architecture
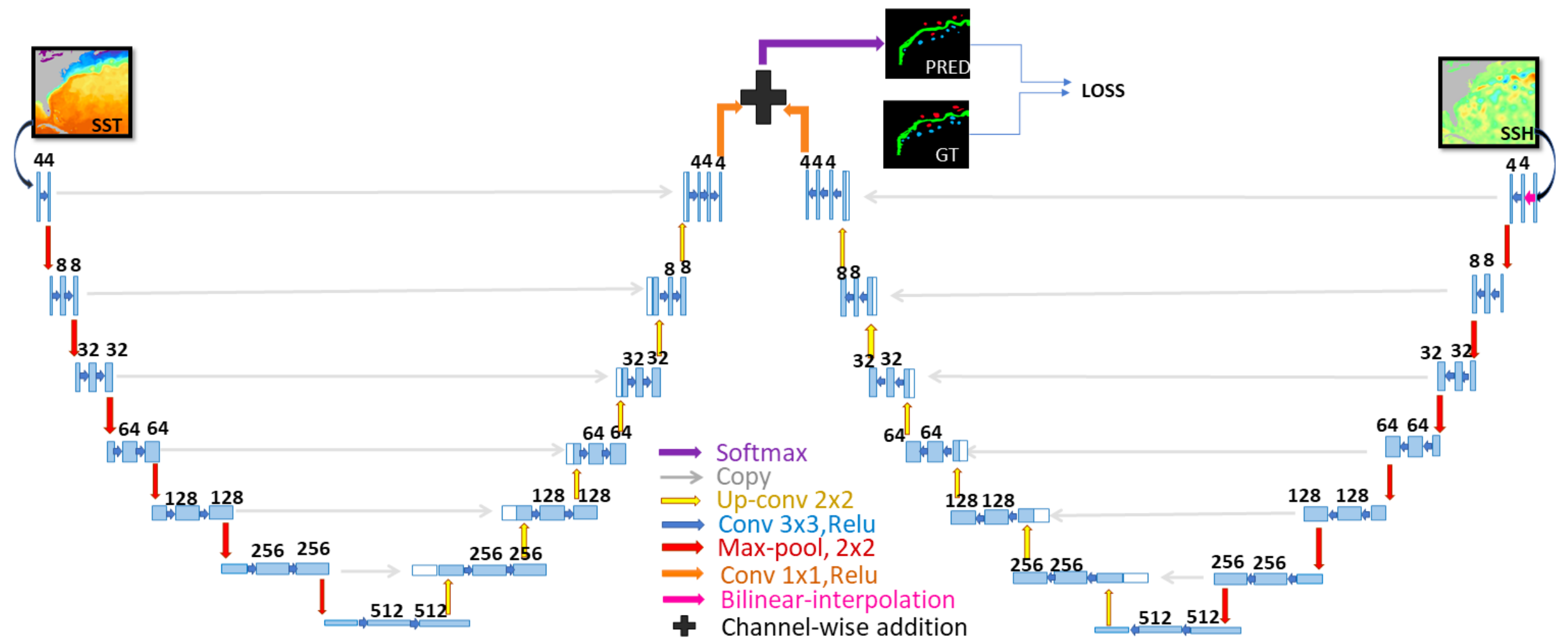
SST input

SSH input

Training in action
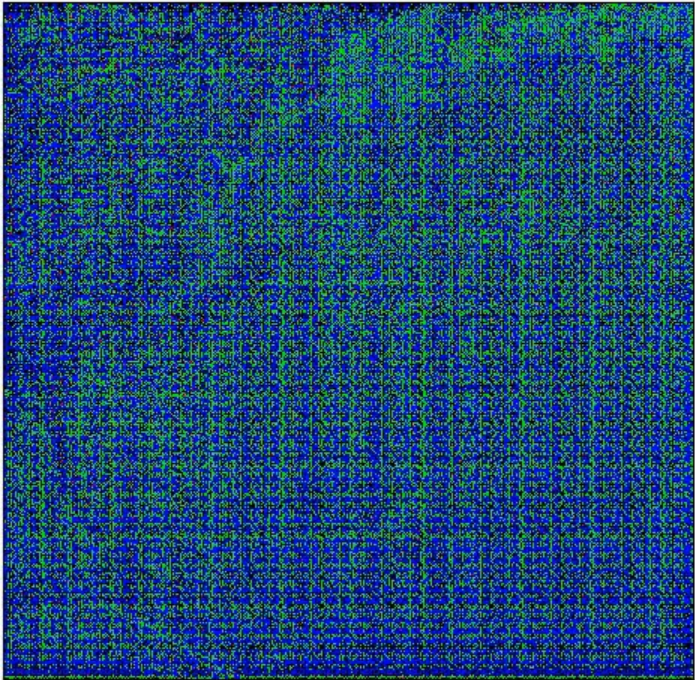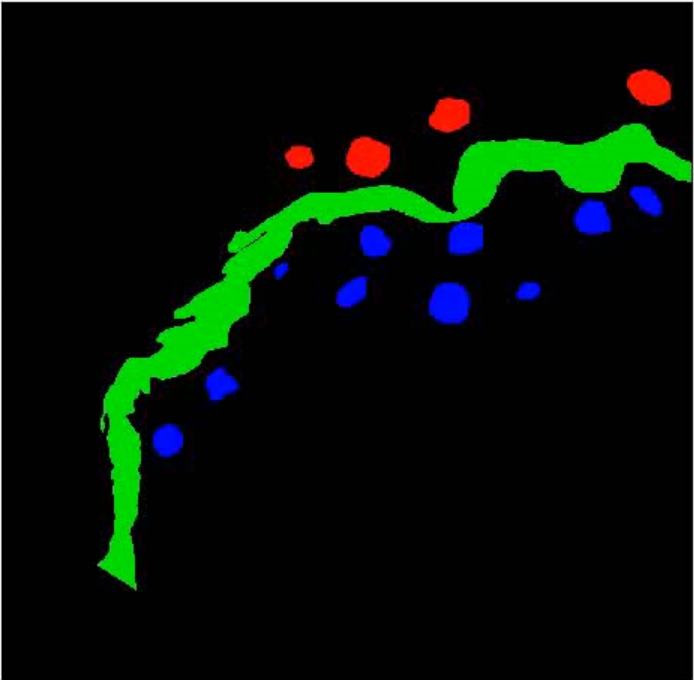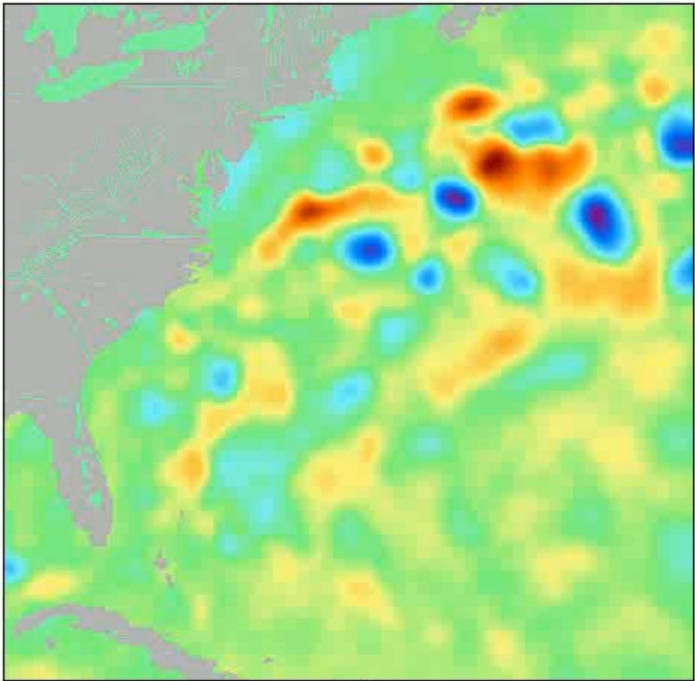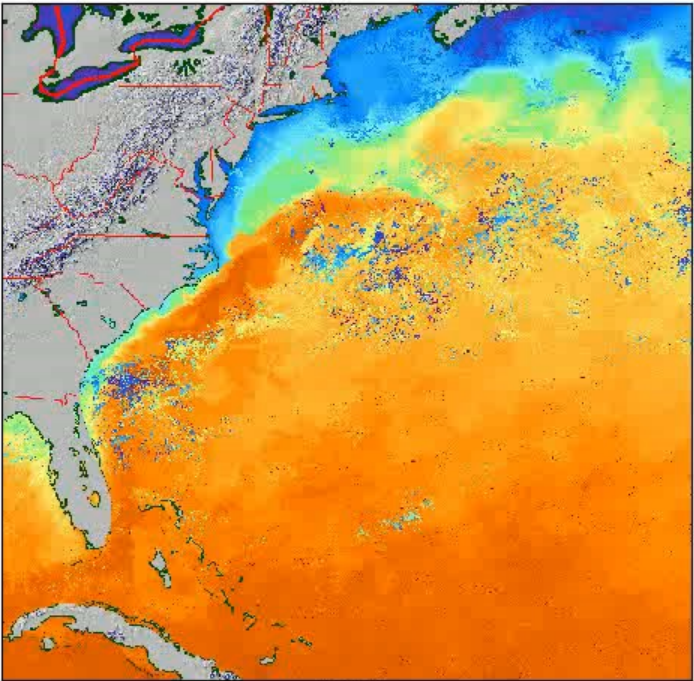
Ground Truth

W-net prediction

Intermediate Predictions
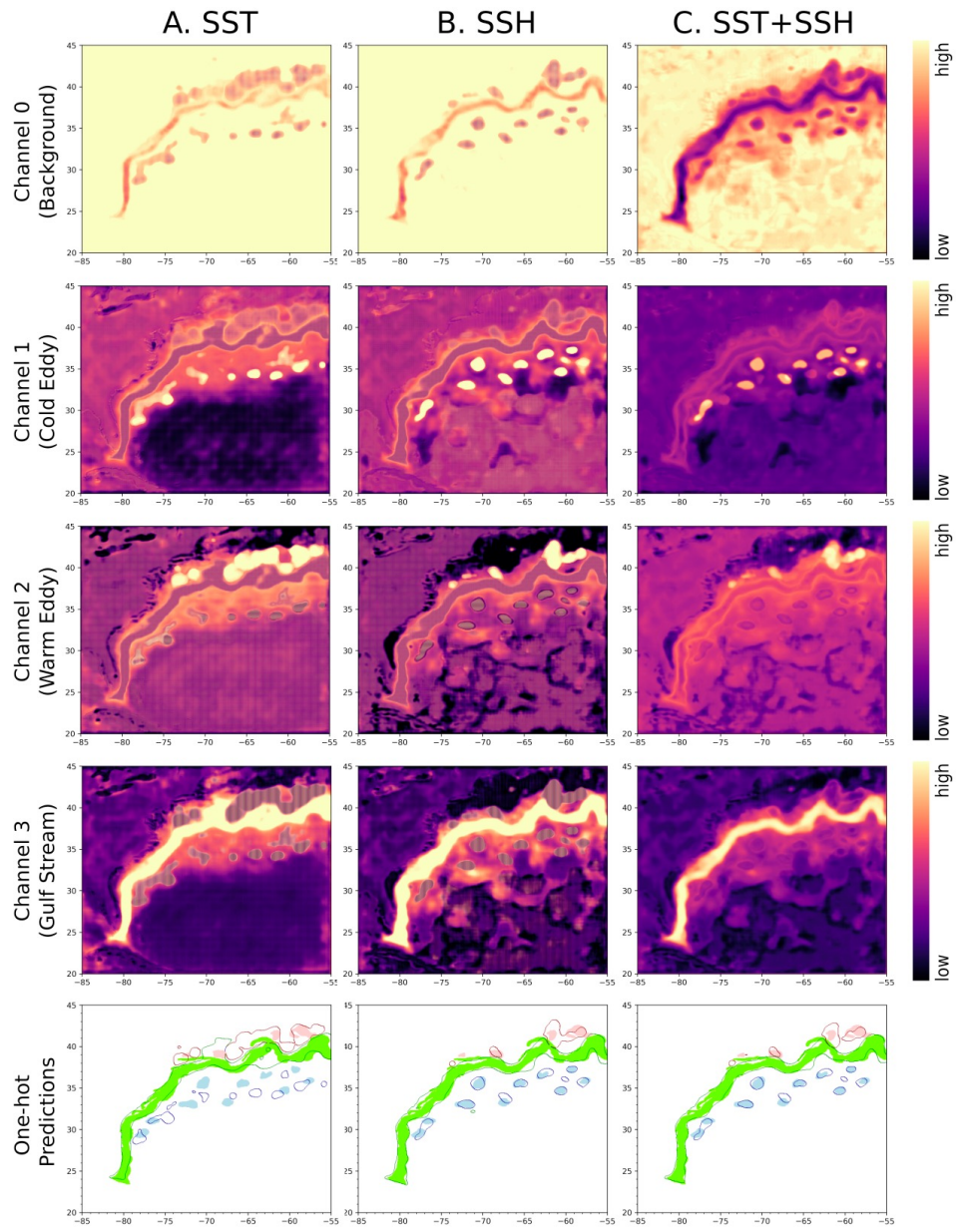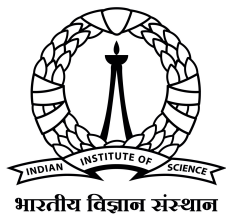
Test Images

YOLO Project Starting

- http://christopher5106.github.io/object/detectors/2017/08/10/bounding-box-object-detectors-understanding-yolo.html
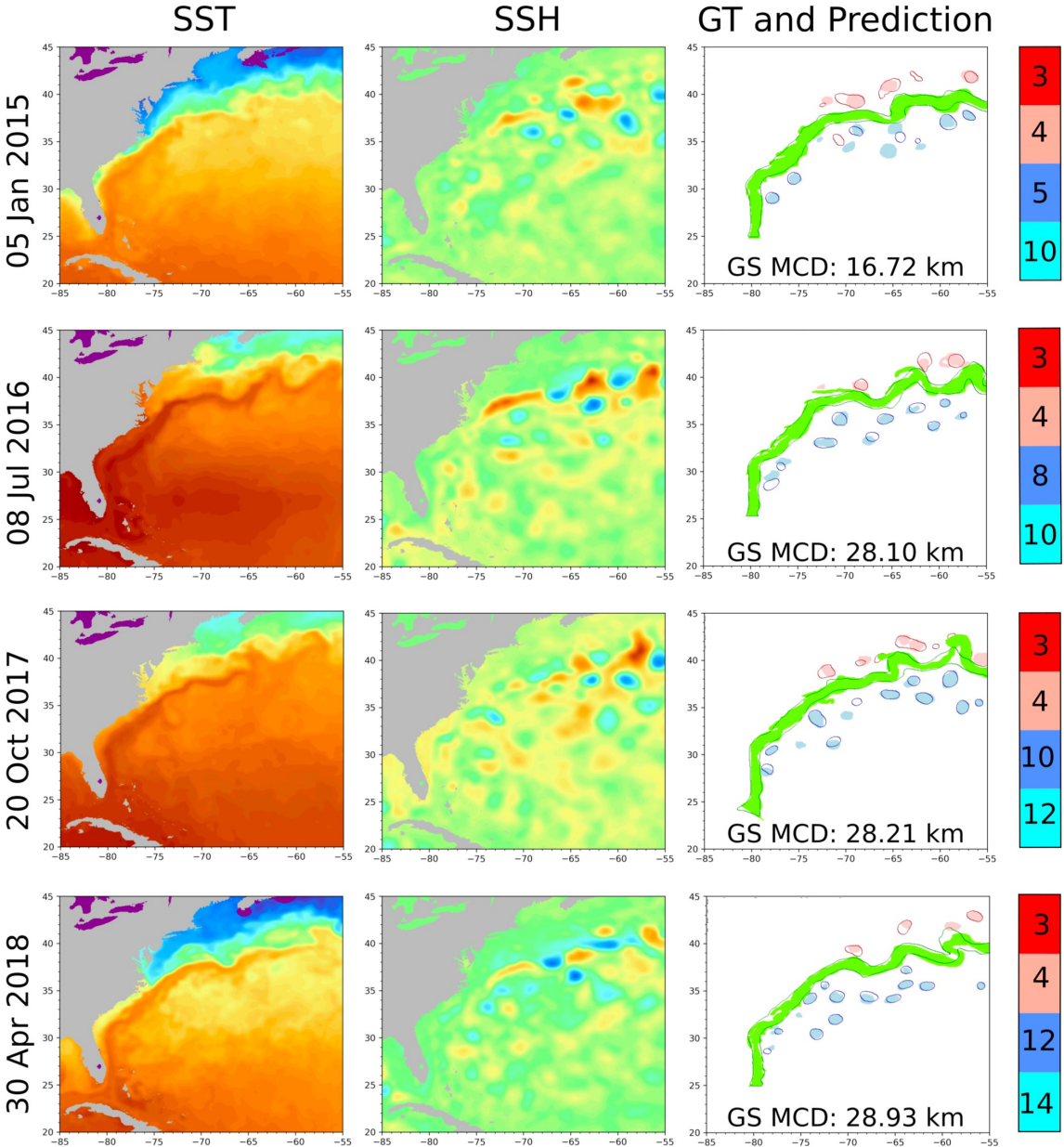
End of Additional Material

- Satellite Image Analytics Research