REPUBLIC POLYTECHNIC
DISCOVER. TRANSFORM. ACHIEVE

Advanced Deep Learning in Computer Vision

Day 2

Image Segmentation and Face Recognition

Programme

Day 1	Transfer Learning Activity – Transfer Learning Fine Tuning Object Detection and Localization Activity: – Localization using Haar Cascades	More Object Detection and Localization Activity – Using YOLOv3 and SSD Annotation Activity – Annotation Hands-on
Day 2	Image Segmentation Activity – - OpenCV Mask RCNN - Keras Mask RCNN - Training Customized Mask RCNN	Activity: - Using Customized Mask RCNN Face Detection and Recognition Activity: - Create Face Database - Face Recognition
Day 3	Advanced Generative Adversarial Network Activity - DCGAN for small color photographs - Conditional GAN	Customised Dataset with Yolo Activity - Thermal Images - Aerial Images

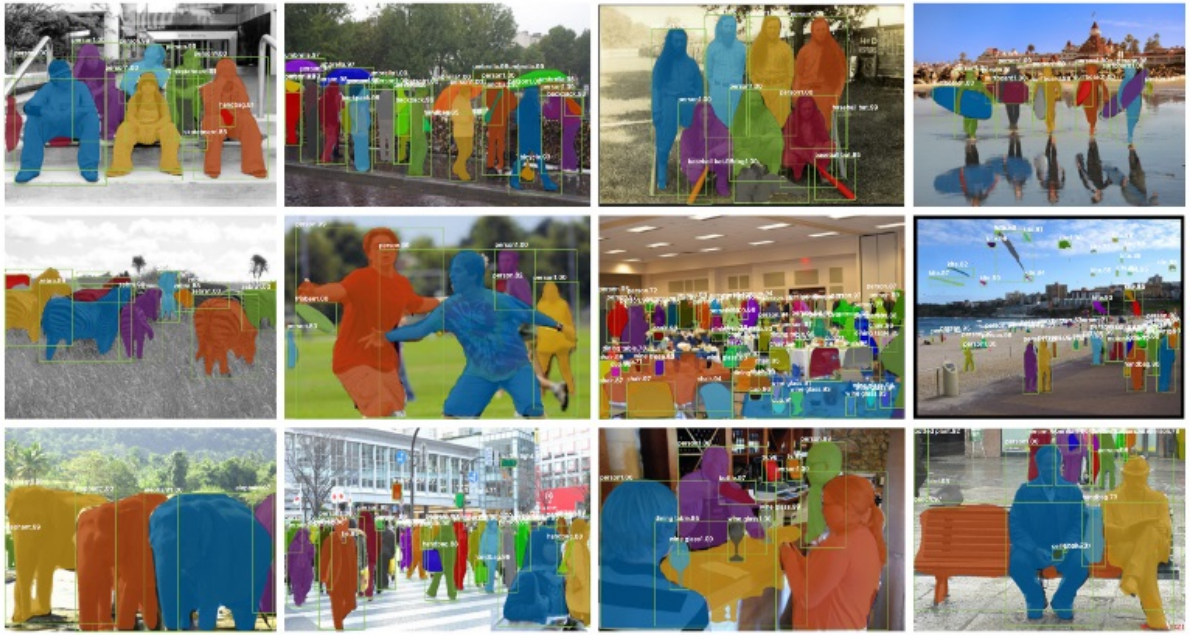


Image Segmentation

What is Image Segmentation

- Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as image objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics..
  - Wikipedia

What is Image Segmentation

- Examples

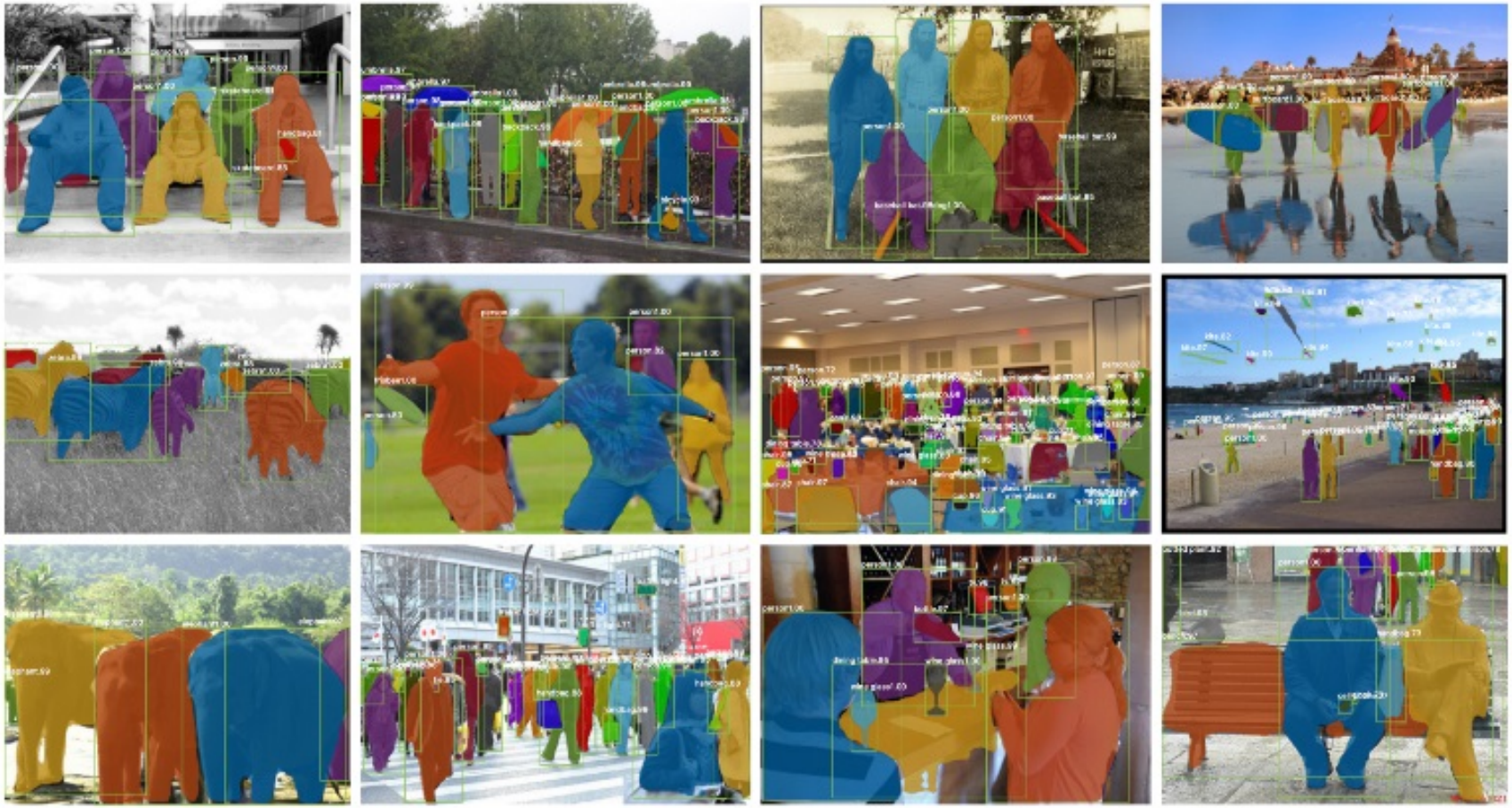


Image from: https://www.topbots.com/object-detection-and-instance-segmentation/

What is Image Segmentation

Image Classification helps us to classify what is contained in an image

Image Localization will specify the location of single object in an image.

Object Detection specifies the location of multiple objects in the image.

Image Segmentation will create a pixel wise mask of each object in the images. We will be able to identify the shapes of different objects in the image.

What is Image Segmentation

- Three types of Segmentation
  - Semantic Segmentation – e.g Apple, Orange
  - Instance Segmentation – e.g First Apple, Second Apple, etc
  - Panoptic Segmentation – combine both Semantic and Instance
    - Instance segmentation for class of interest, other class is semantic segmentation.

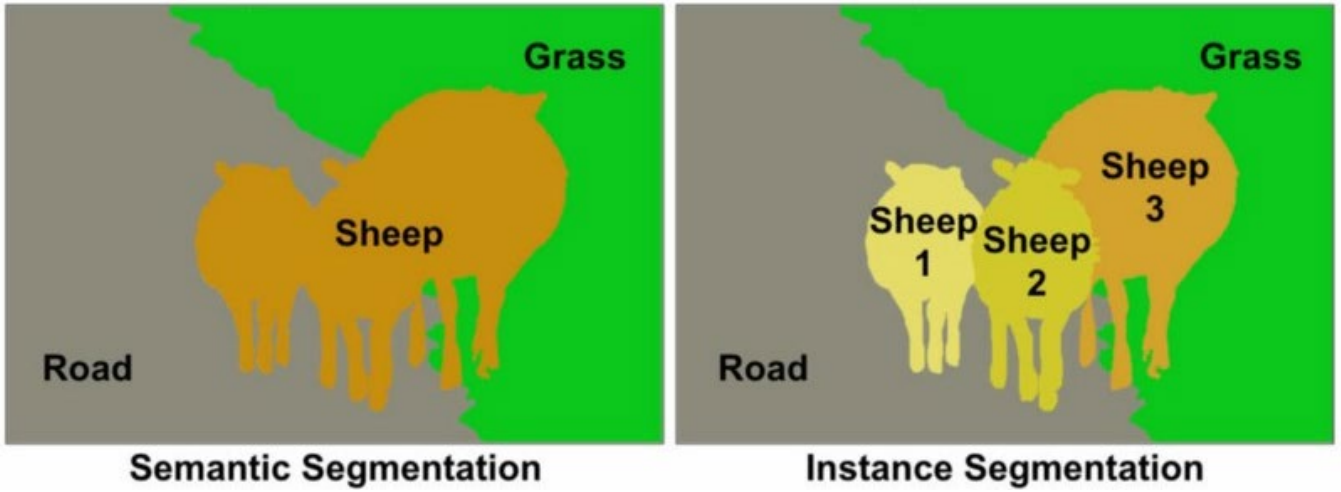


Image from: https://towardsdatascience.com/detection-and-segmentation-through-convnets-47aa42de27ea

How to do Image Segmentation

- **Primitive (but effective) algorithms**
  - Super-Pixels
  - Watershed
  - Conditional Random Fields (CRF)

- **Deep Learning algorithms**
  - Fully Convolution Network (FCN) Segmentation
    - DeepLab (v3)
  - Mask R-CNN

How to do Image Segmentation

- **DeepLab (v3)**
  - What is Fully Convolution Network (FCN)?

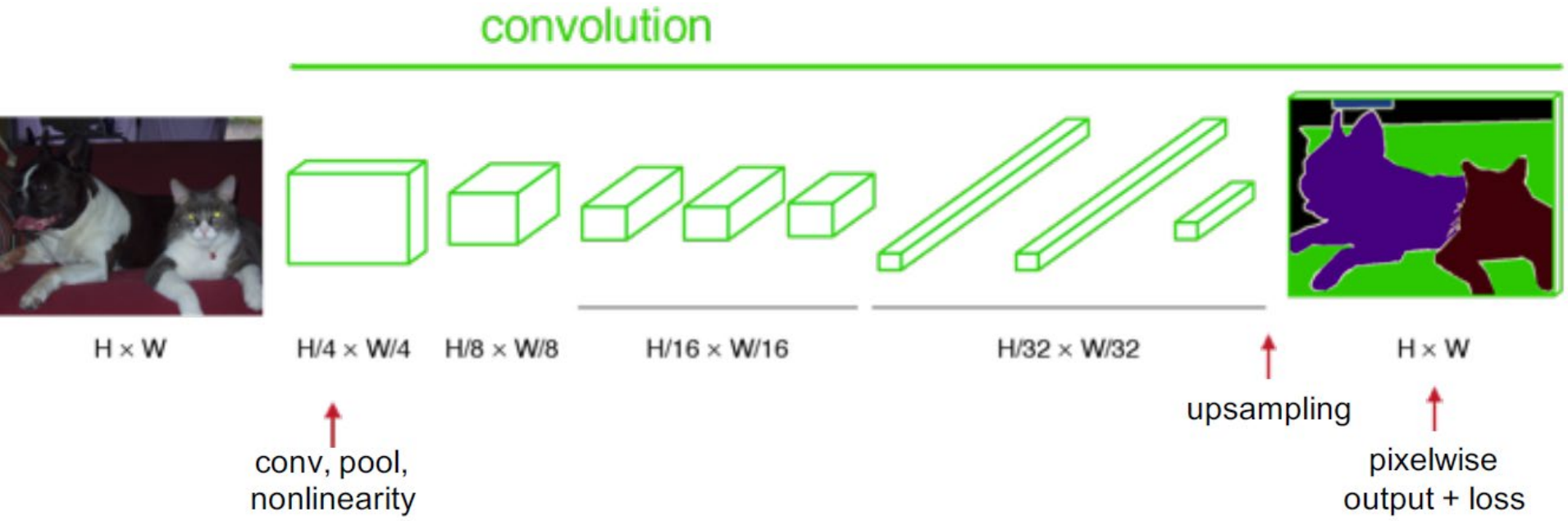
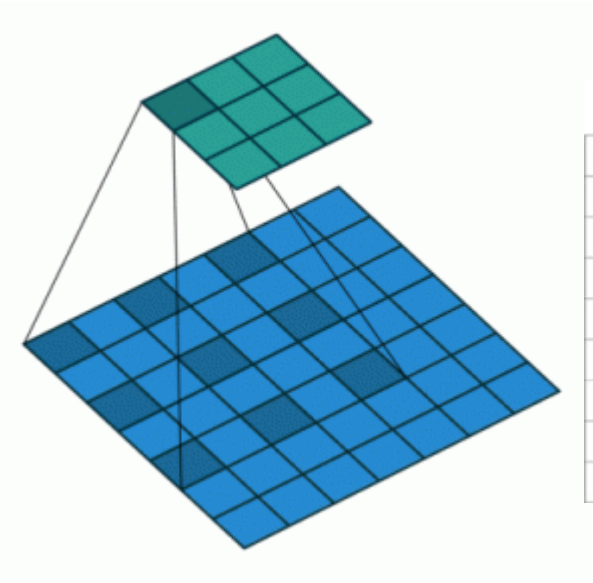


Image from: https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1

How to do Image Segmentation

- **DeepLab (v3)**
  - Introduced “Atrous Convolution” , or dilated convolutions, to extract more dense features where information is better preserved given objects of varying scale.



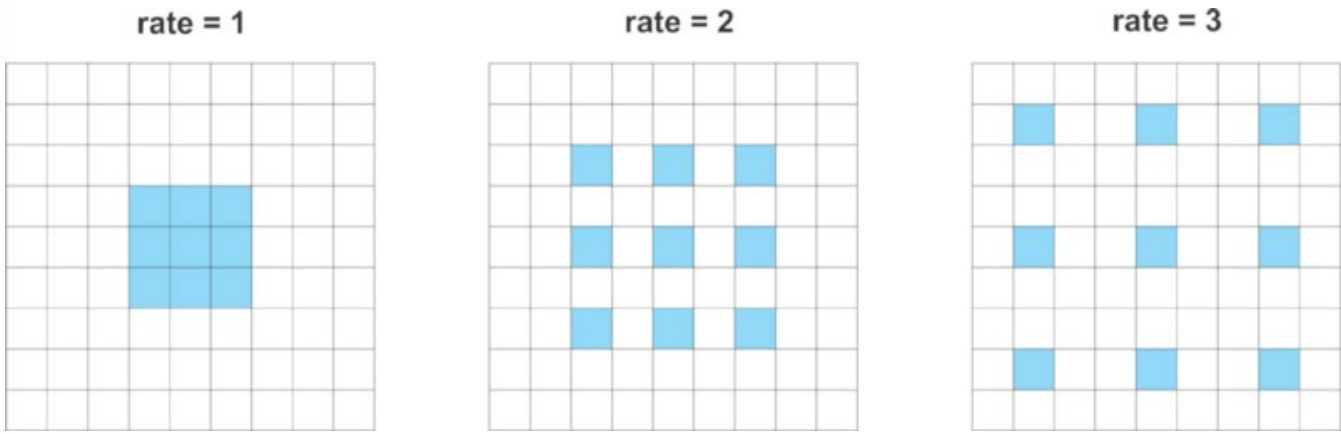


Image from: https://www.analyticsvidhya.com/blog/2019/02/tutorial-semantic-segmentation-google-deeplab/

How to do Image Segmentation

- **DeepLab (v3)**

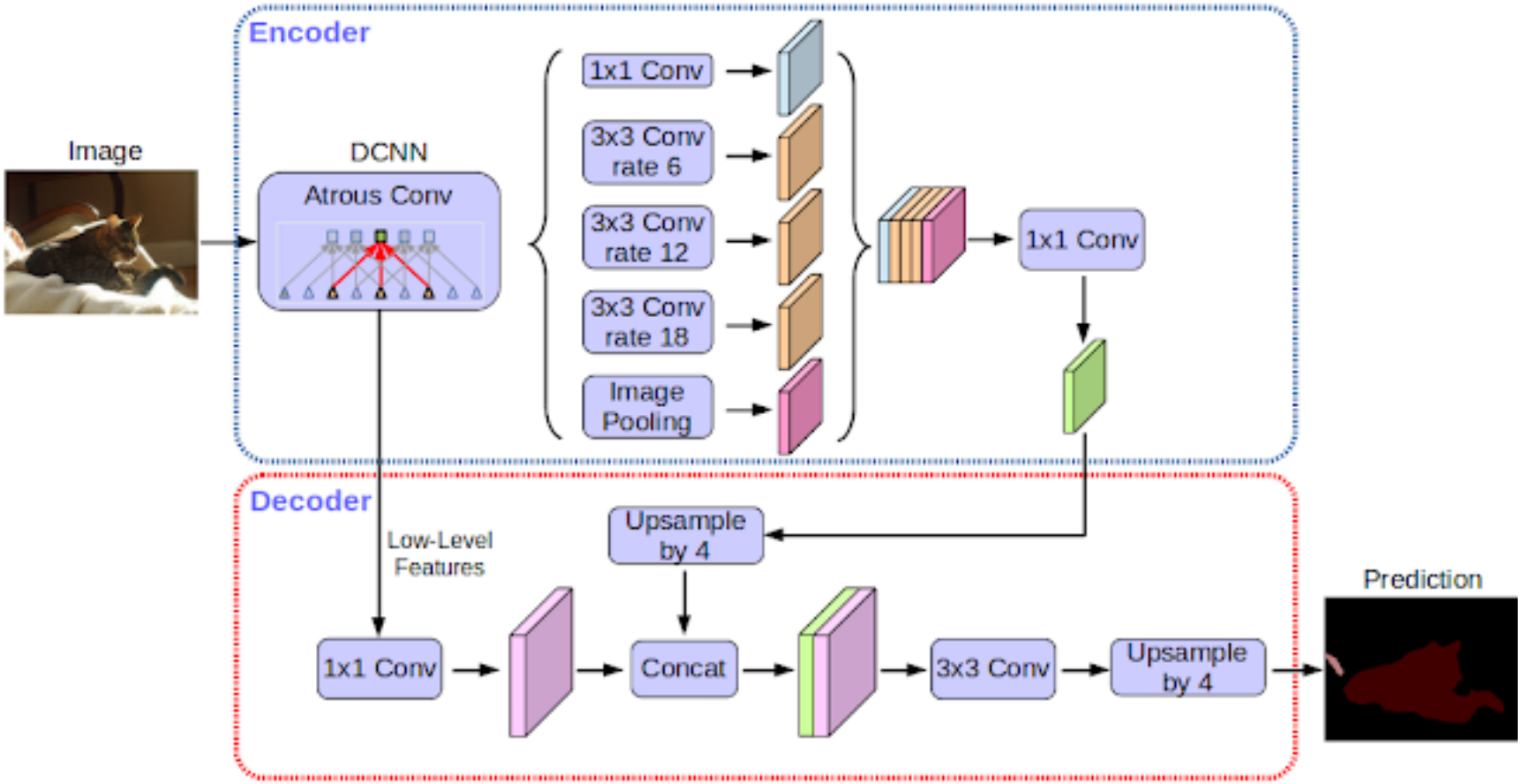
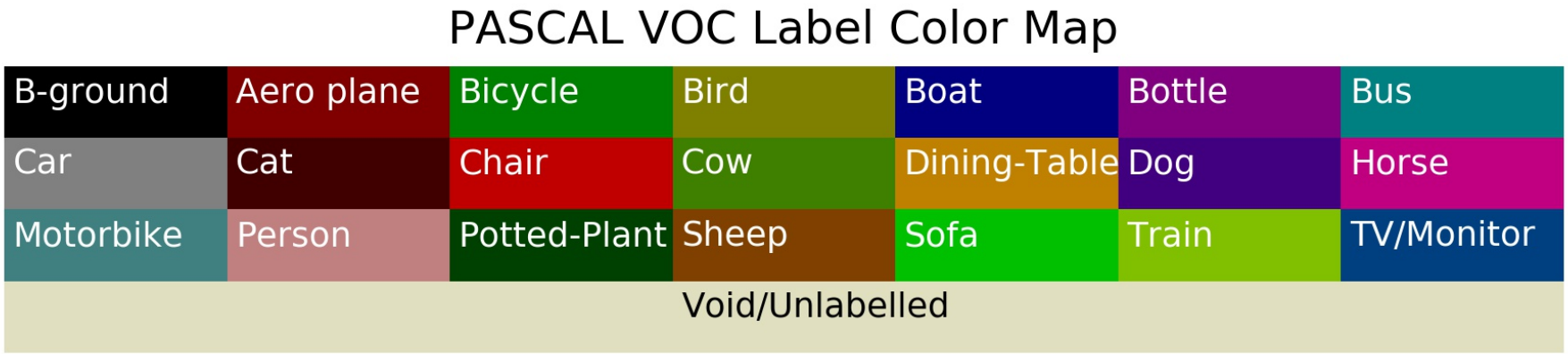


Image from: https://ai.googleblog.com/2018/03/semantic-image-segmentation-with.html

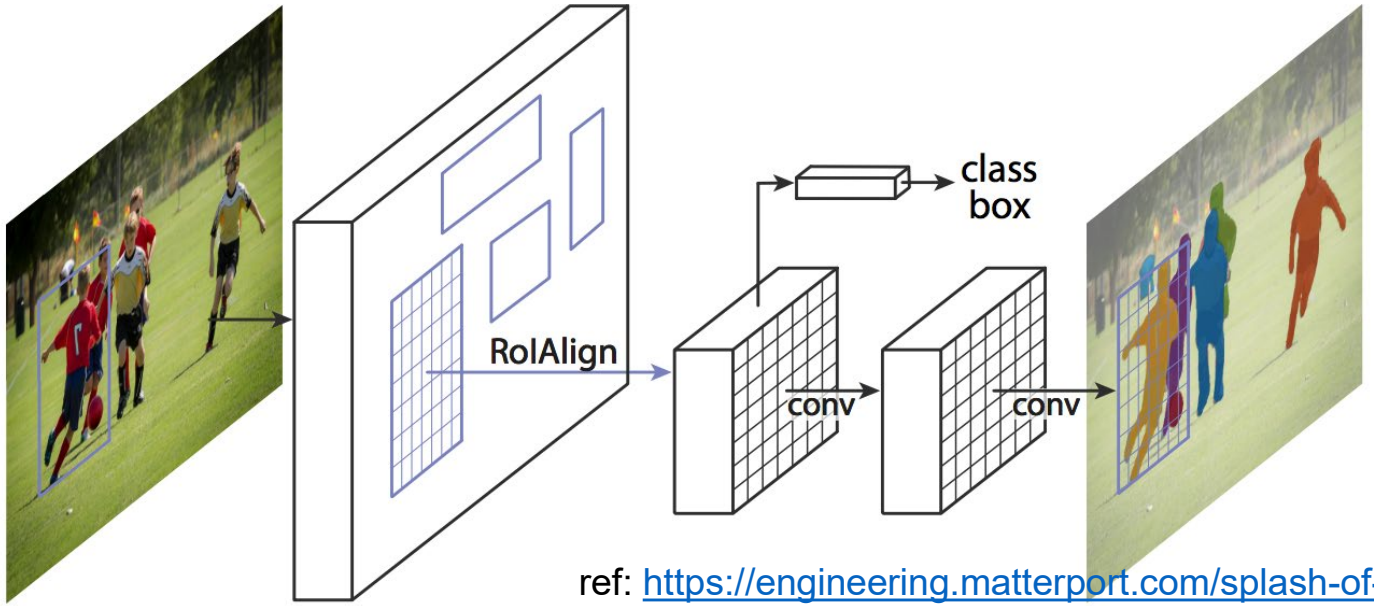
Pascal VOC

- **21 classes**



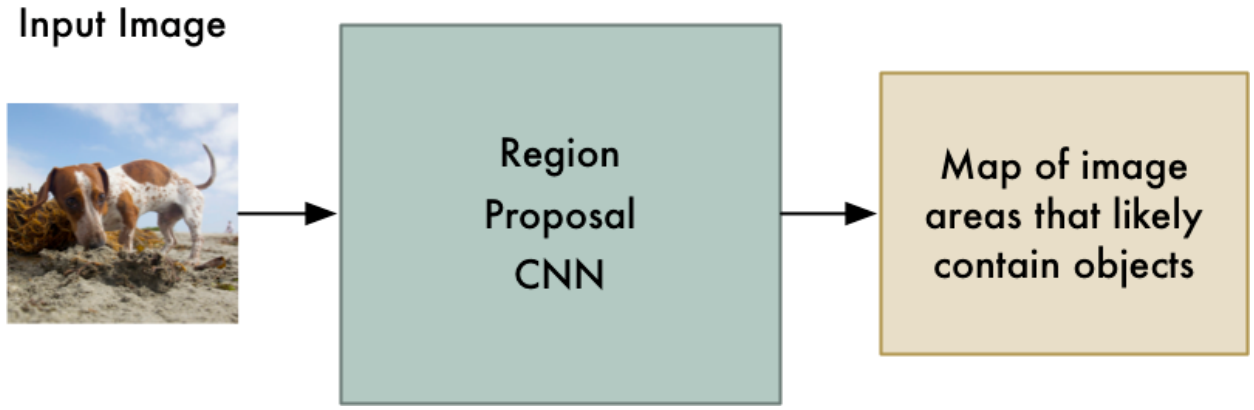
How to do Image Segmentation

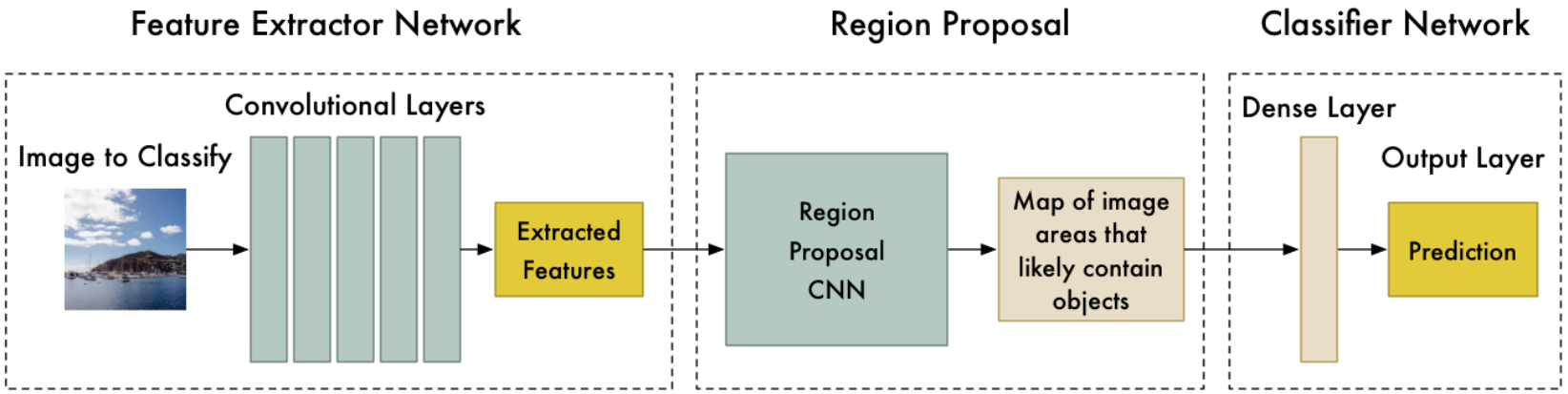
- **Mask R-CNN**
  - Mask R-CNN (regional convolutional neural network) is a two stage framework: the first stage scans the image and generates proposals(areas likely to contain an object). And the second stage classifies the proposals and generates bounding boxes and masks.



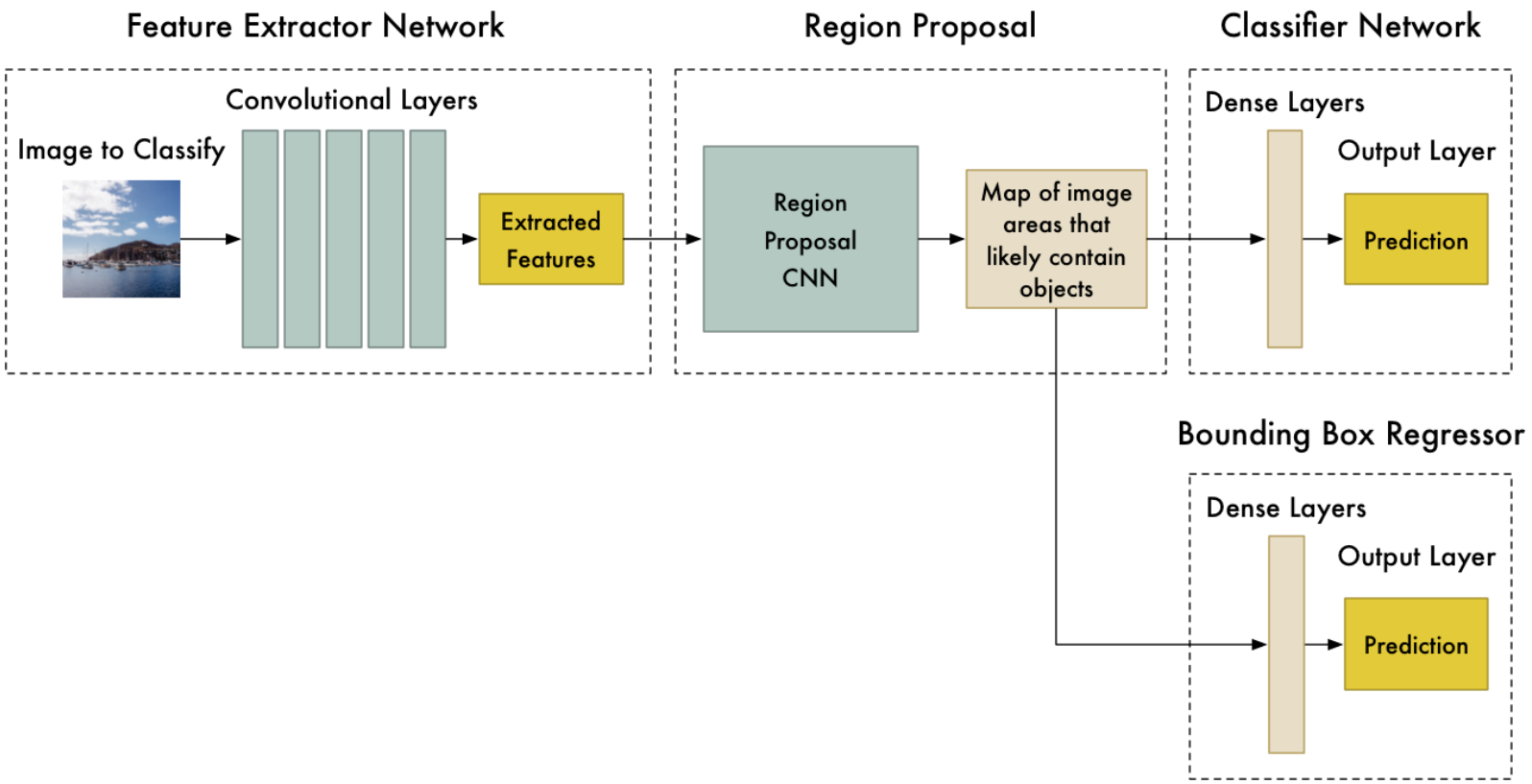
ref: https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46

Region Proposal Network





Region Proposal Network

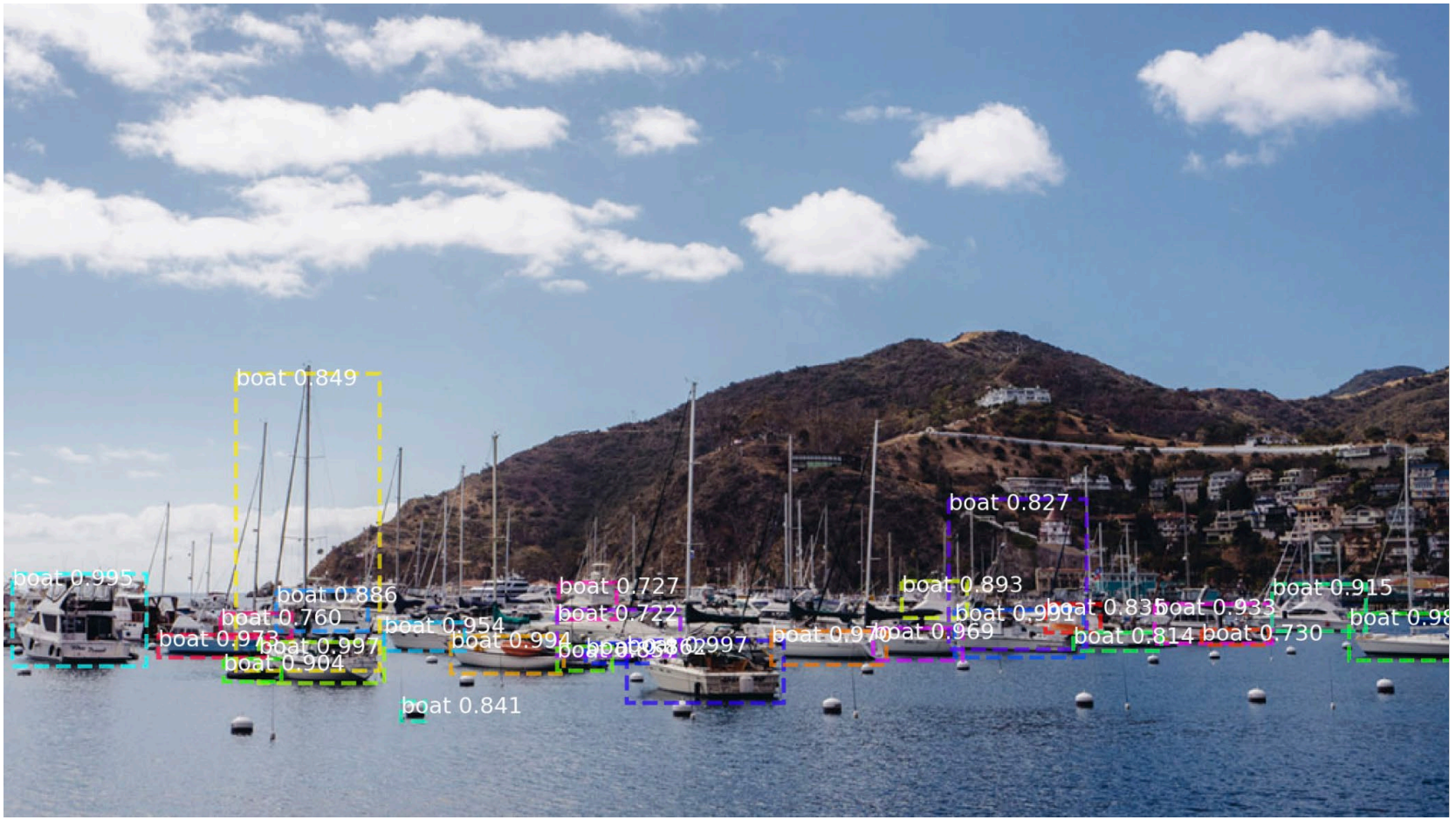


Localization Approach 2: Region Proposal Network

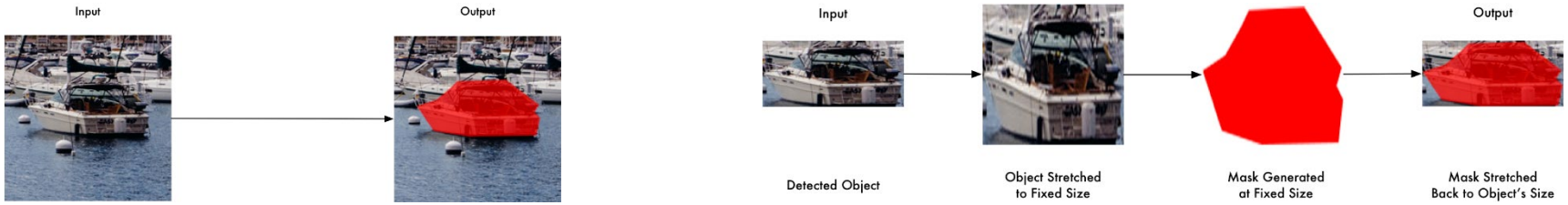
Localization Approach 2: Region Proposal Network

non-maximum suppression algorithm

Classifying the Detected Objects

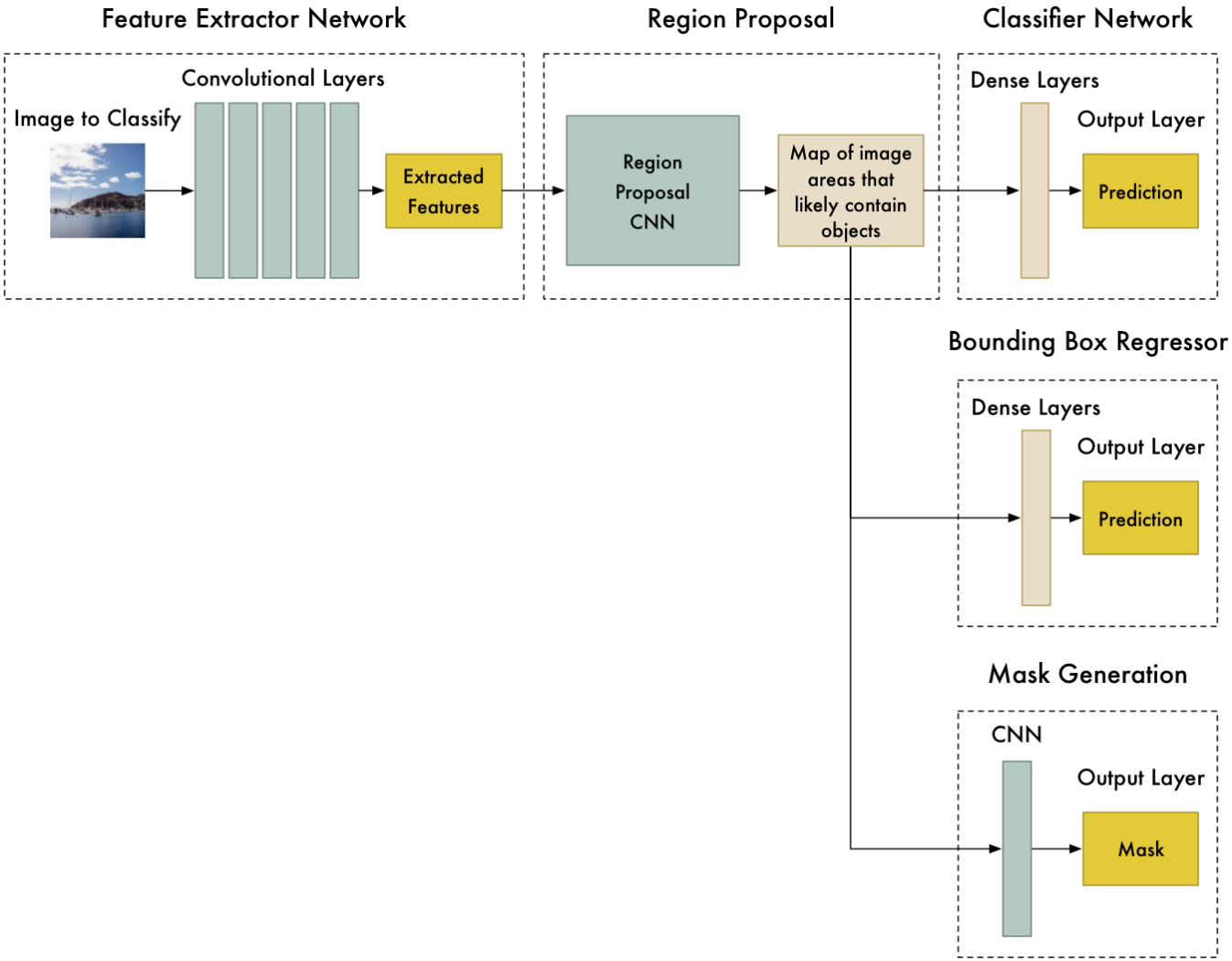


Adding Object Mask Generation to Our Model



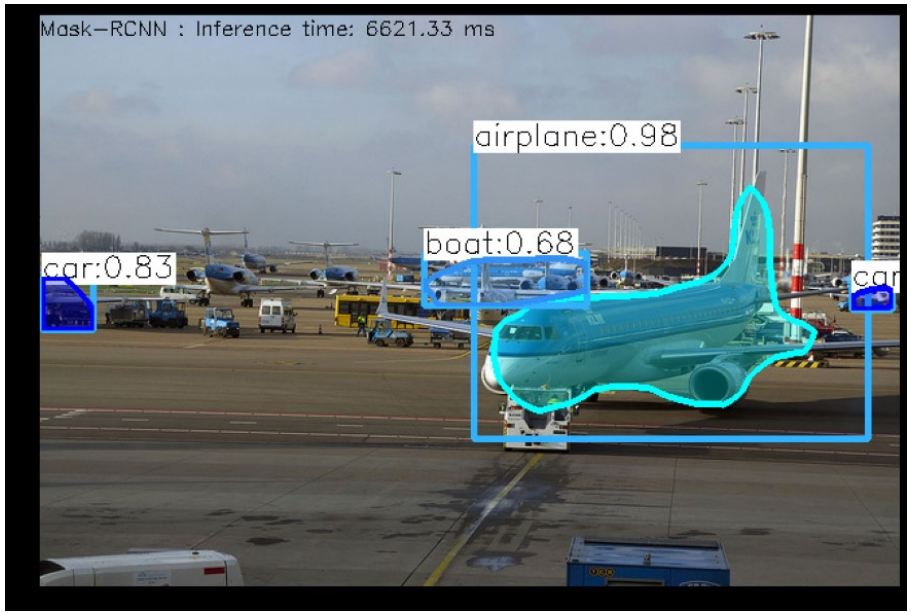
Predicting pixel-by-pixel masks is harder than predicting a bounding box because you have to separate the object from the background image. To train a system like this, all your training data needs to be segmented in the same way. It takes a lot more work to manually trace out shapes in training images than to just draw boxes around objects.

The complete model



Activity – OpenCV Mask RCNN

- **Activity: 3_1_Mask_RCNN_using_OpenCV**



Exercises:
- Download an image of your choice
- Modify the script to make prediction on this image with different confidence and mask threshold

Step 1:
Watch and listen to the instructor's demonstration

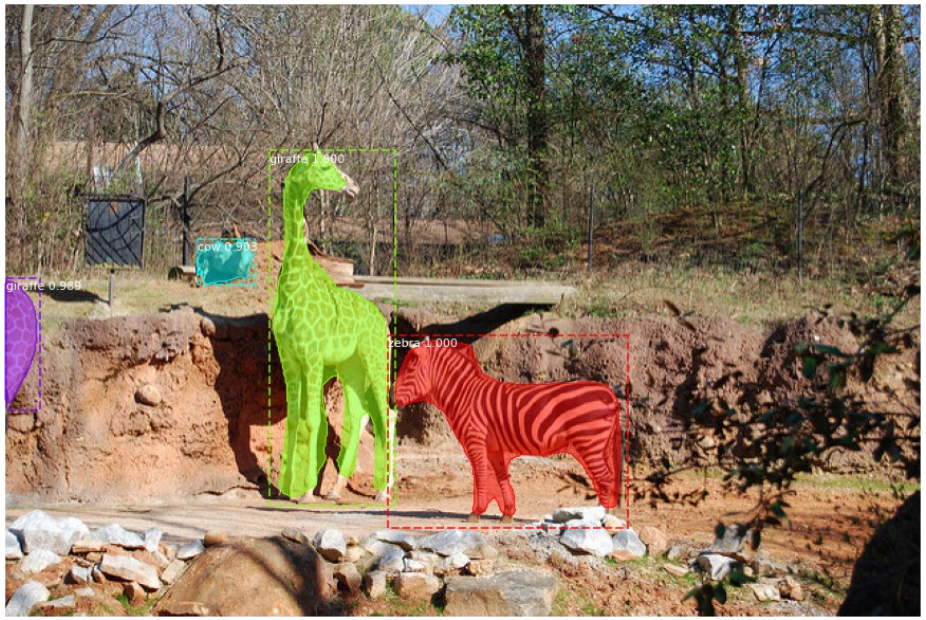
10 mins

Step 2:
Work through the activities

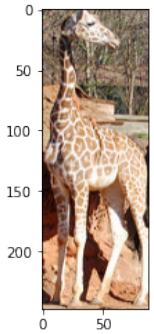
20 mins

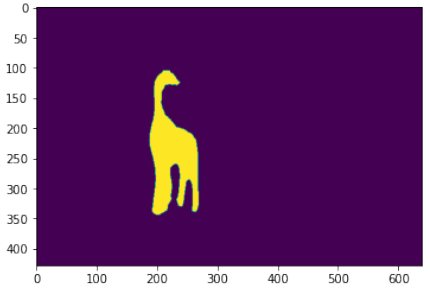
Activity – Keras Mask RCNN

- **Activity: 3_2_Mask_RCNN_using_Keras**



Exercises:
- Explore r['class_ids'] and r['rois'], save/display the first giraffe detected (cropped)
- Explore r['masks'] and save/display the mask for the first giraffe detected





Step 1:
Watch and listen to the instructor's demonstration

10 mins

Step 2:
Work through the activities

20 mins

Activity – Training Custom Mask RCNN

- **Activity: 3_3_Training_Customized_Mask_RCNN**

Warning: This project is especially computationally intensive and it may take a long time to run. You can reduce the number of training epochs (from 30 to 1) to see less accurate results in less time.

Step 1:
Watch and listen to the instructor's demonstration

10 mins

Step 2:
Work through the activities

20 mins

Activity – Using Custom Mask RCNN

- **Activity: 3_4_Using_Customized_Mask_RCNN**

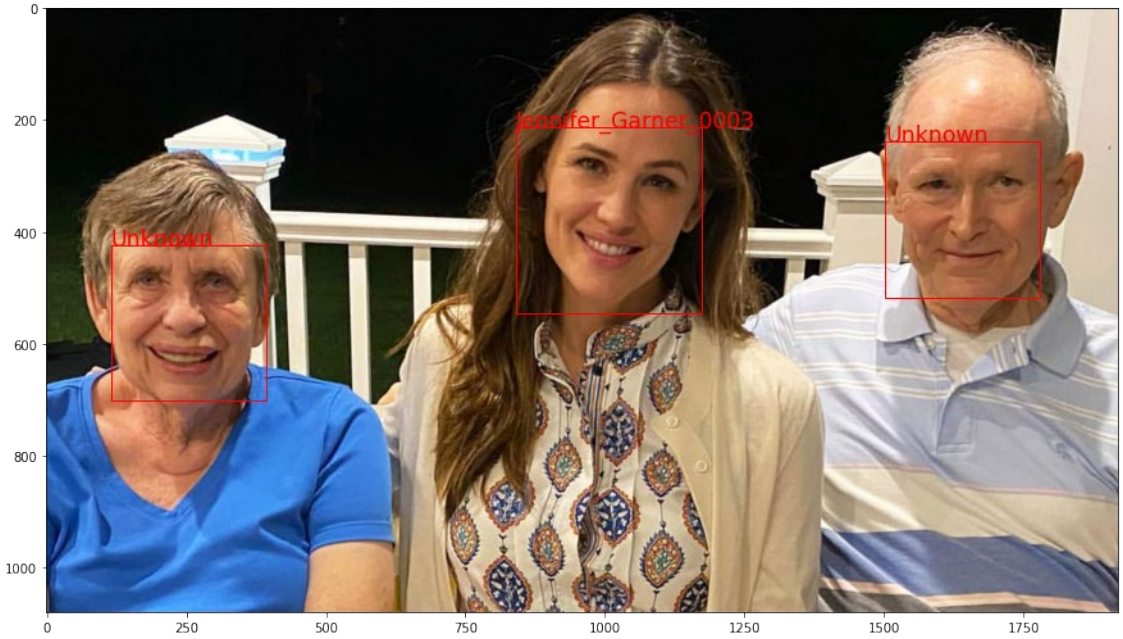
Exercises:
- Use one of the kick scooter images you annotated in day 1 to perform a prediction. Share the result of the class

Step 1:
Watch and listen to the instructor's demonstration

10 mins

Step 2:
Work through the activities

15 mins



Face Recognition

What is Face Recognition

- **A *facial recognition system* is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source.**
  - Ref: Wikipedia

- **Applications of facial recognition**
  - Unlocking phones
  - Access control to facilities
  - Track attendance
  - Identity verification
  - Security and crime prevention
  - Ref: https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition
  - Ref: https://www.facefirst.com/blog/amazing-uses-for-face-recognition-facial-recognition-use-cases/

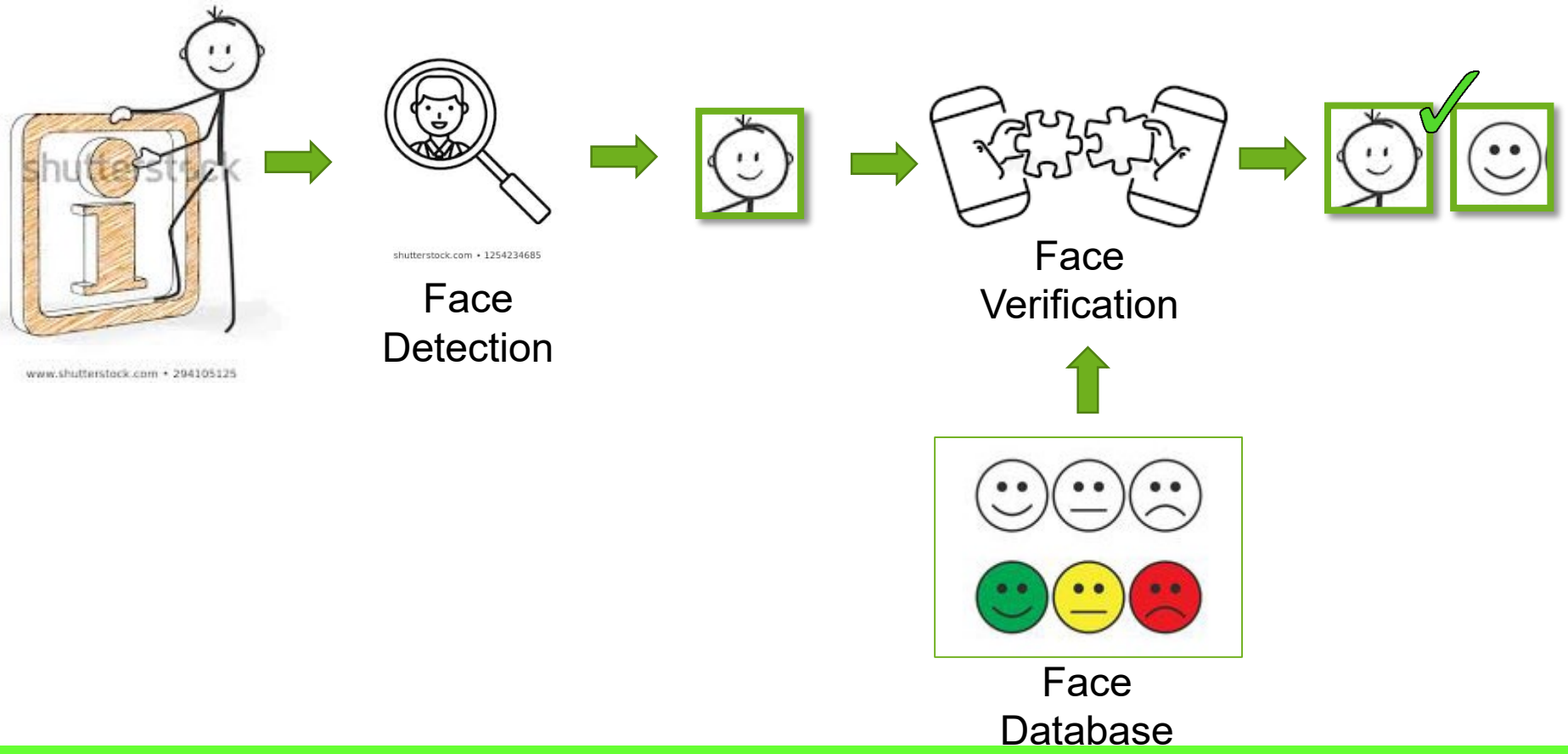
What is Face Recognition

Facial image of individuals are protected by law.

(PERSONAL DATA PROTECTION ACT)

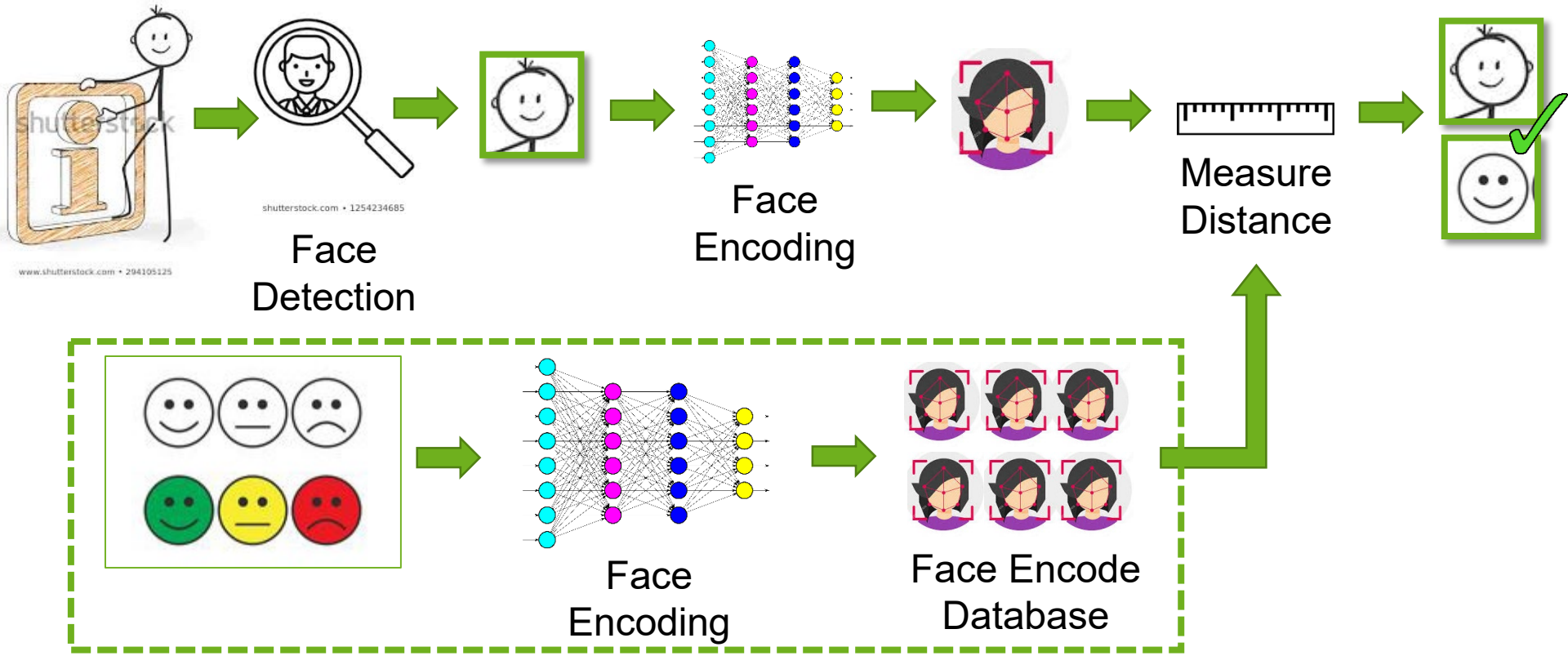
How to do Face Recognition

- Typical workflow



How to do Face Recognition

- Deep Learning approach (a.k.a Face Encoding)



face_recognition module

Built using dlib's state-of-the-art face recognition built with deep learning. The model has an accuracy of 99.38% on the Labeled Faces in the Wild benchmark.

```
[-0.23, -0.54, ..., 0.27]
```

Facial recognition via deep learning and Python using the face_recognition module method generates a 128-d real-valued number feature vector per face.

face_recognition API

.face_locations() → .face_encodings() → .face_distance() → .compare_faces() → Obtain name from index

Returns an array of bounding boxes of human faces in a image

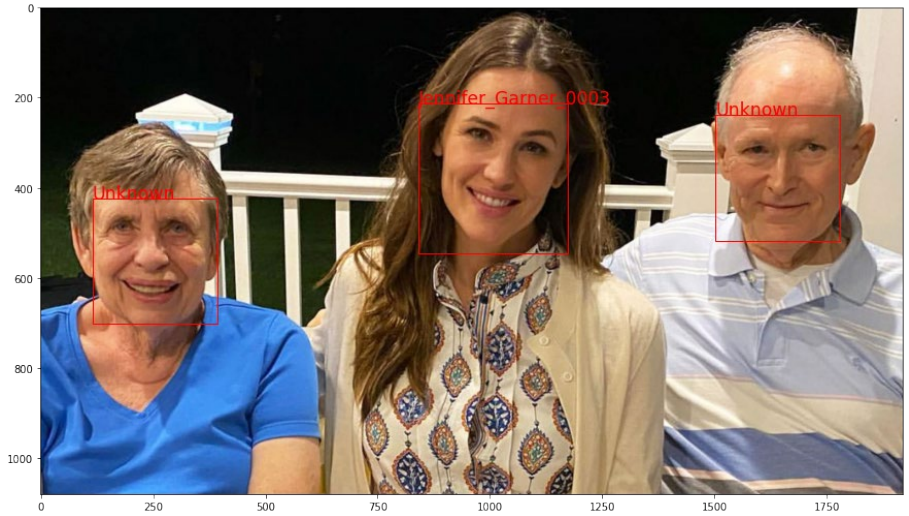
Given an image, return the 128-dimension face encoding for each face in the image.

Given a list of face encodings, compare them to a known face encoding and get a euclidean distance for each comparison face. The distance tells you how similar the faces are.

Compare a list of face encodings against a candidate encoding to see if they match.

Activity – Face Recognition

- **Activity: 4_1_Create_Face_Database**
- **Activity: 4_2_Face_Recognition**



Exercises:
- Create a face encoding dataset to include yourself and the person seated to the left and right of you.

Step 1:
Watch and listen to the instructor's demonstration

10 mins

Step 2:
Work through the activities

25 mins

Thank you