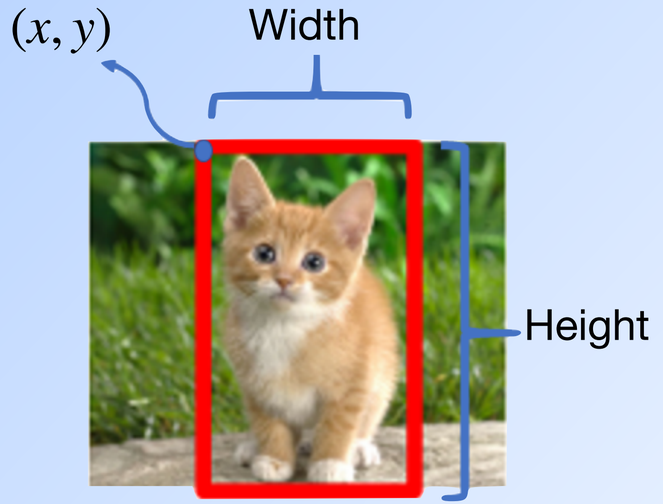
AI61201: Visual Computing With AI/ML

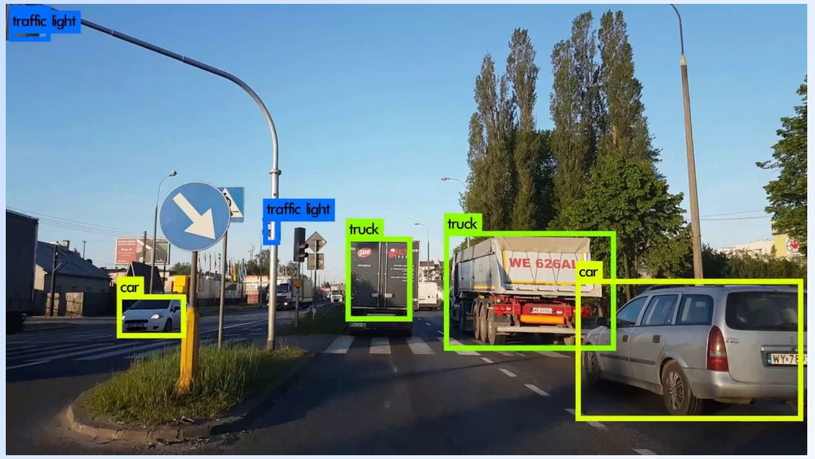
Module 8: Object Detection and Segmentation

Dr. Somdyuti Paul

Object Detection and Localization

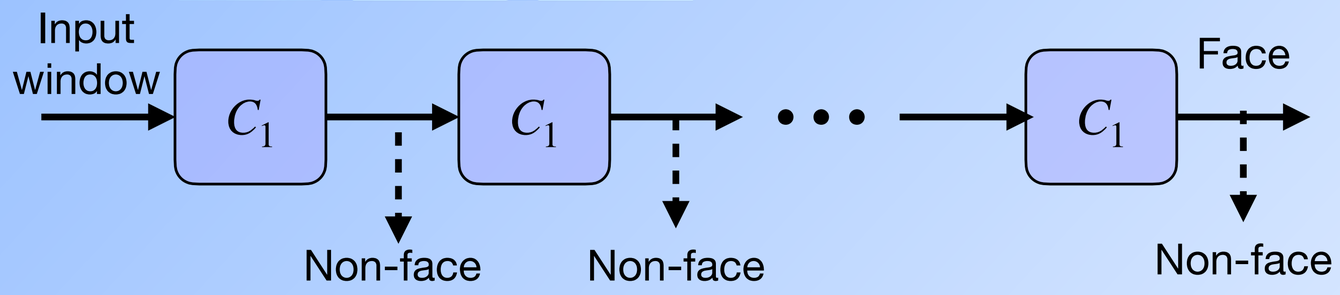
- The goal of object detection is to find one or more objects of interest in an image and their locations.
- The main challenges in object detection include the following:
  - Variability in object appearance.
  - Variations in scale
  - Presence of background clutter
  - Intra-class variability
- Traditional object detectors were based on
  - Features such as Harris, SIFT, SURF, HoG etc.
  - Deformable Parts Model
  - Haar Cascades

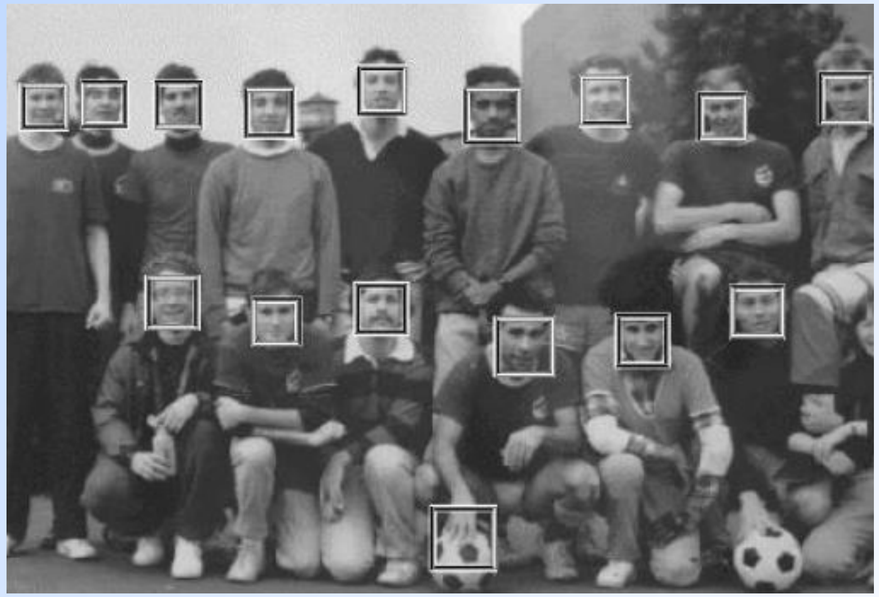




Viola Jones Face Detector Using Haar Cascades

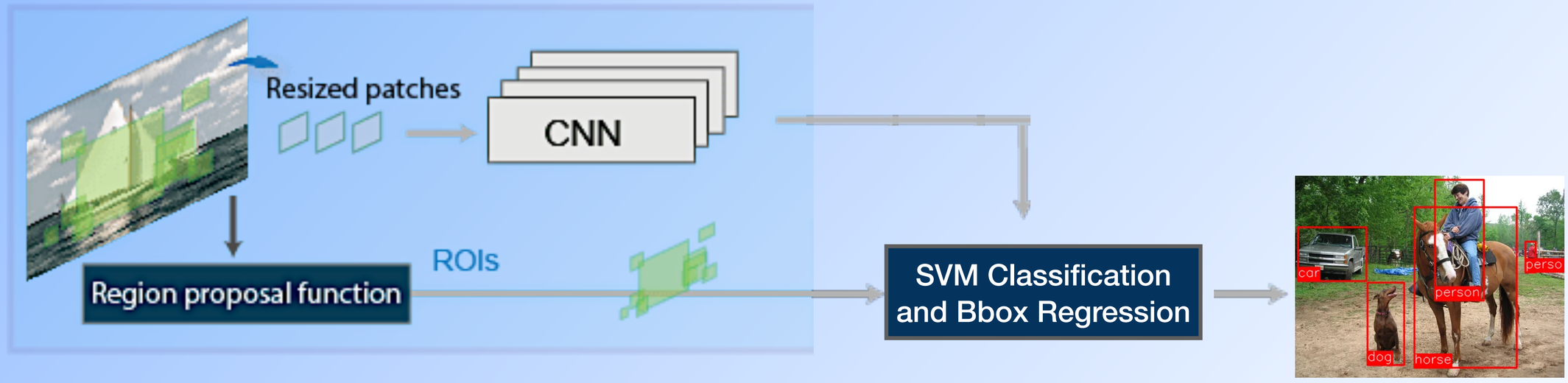
- Uses a multi-stage cascaded classifier was trained for face detection using Adaboost (Viola and Jones, 2001).
- Each state classifies the input as face or non-face using Haar features which are very simple and can be computed efficiently.
- The weak classifier having the highest weight is evaluated first and only positive detections are passed on to the next stage during inference.
- The resulting algorithm, called the Viola Jones object detector was the first algorithm of its kind to demonstrate robust real-time performance.





Object Detection Using R-CNN

- The Region based convolutional neural network (R-CNN) was a pioneering object detection model that integrated traditional computer vision techniques with deep learning to achieve effective object detection (Girshick et al., 2014).
- The R-CNN is a two-stage approach which first generates region proposals and then classifies each region proposal using features learned from a CNN model



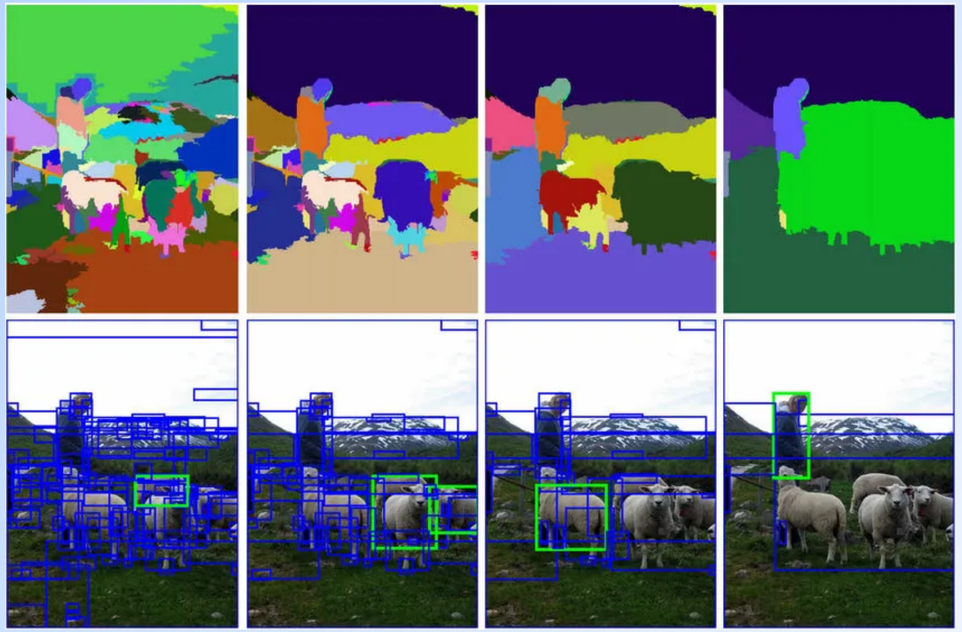
R-CNN Model

Prediction

Object Detection Using R-CNN

Stages in R-CNN model:

1. Region Proposal: the region proposal stage generates a set of regions that are likely to contain objects. This stage relies on external algorithms such as selective search or edge boxes to find about 2000 region proposals.
2. Feature Extraction: the region proposals are padded by 16 pixels on each side to add context and warped to the required input size and given as input to the CNN (AlexNet) model to extract the learned feature embeddings.
3. Object Classification: the CNN features were used to train separate SVM classifiers for each class to determine whether the region proposal contained an instance of the specific classes.
4. Bounding Box Regression: a linear regression model was trained to refine the size and location of the bounding boxes for each class.
5. Non-maximum suppression (NMS): NMS was applied to remove highly overlapping bounding boxes. This step retains the most non-overlapping and confident bounding box predictions.



Iteratively combining regions to generate region proposals

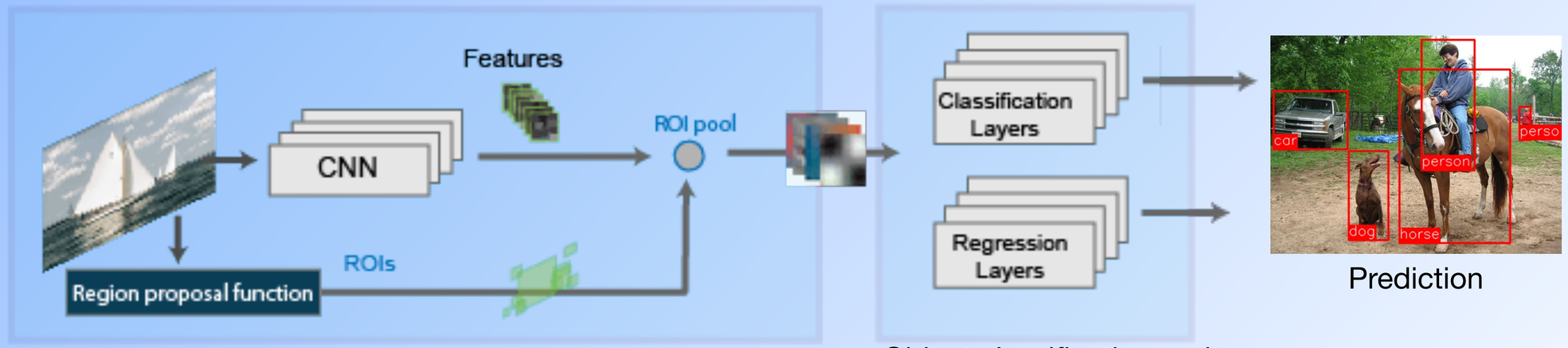
Object Detection Using R-CNN

- R-CNN outperformed contemporary models on object detection datasets such as Pascal VOC and ILSVRC-2013 detection.

Object detections using R-CNN

Fast R-CNN

- The R-CNN model is computationally intensive due to the requirement of CNN feature extraction from a large number of region proposals.
- The individual components in R-CNN were individually trained, i.e. it is not an end-to-end model.
- The Fast-RCN model was proposed to improve the speed and accuracy of RCN predictions (Girshick, 2015).
- Fast-RCN pools the CNN feature maps over the regions corresponding to the region proposals (RoI pooling) so that the entire image could be processed by the CNN at once.



Feature extraction

Object classification and bbox regression

Prediction

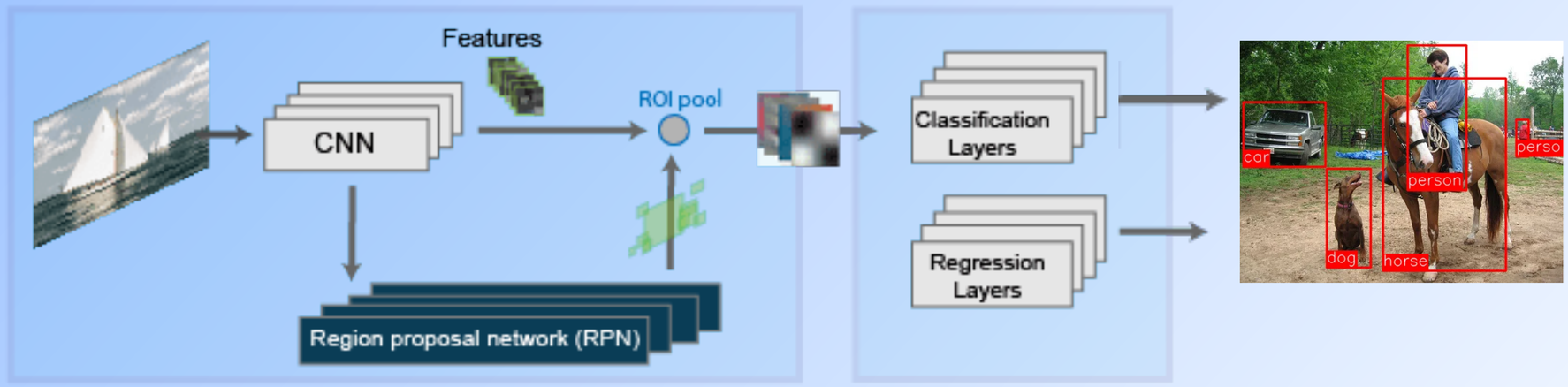
Fast R-CNN Model

Fast R-CNN

- Fast RCN also removed the SVM classifier and linear regression models for performing object classification and bounding box regression.
- Although an external model was still used to generate region proposals, the model was trained end-to-end.
- The loss consisted of two parts - classification loss and bounding box regression loss.
- Fast R-CNN was reported to be 213 times faster than R-CNN at test time, and also attained a higher mean average precision (mAP).

Faster R-CNN

- Faster R-CNN replaced the external region proposal algorithm with a region proposal network (RPN), which enabled end-to-end training of the entire framework (Ren et al., 2015).
- The RPN is a fully convolutional network that operates on the feature maps produced by the backbone CNN to directly generate region proposals.
- Faster R-CNN was able to achieve real-time detection by leveraging the capability of GPUs for generating region proposals.



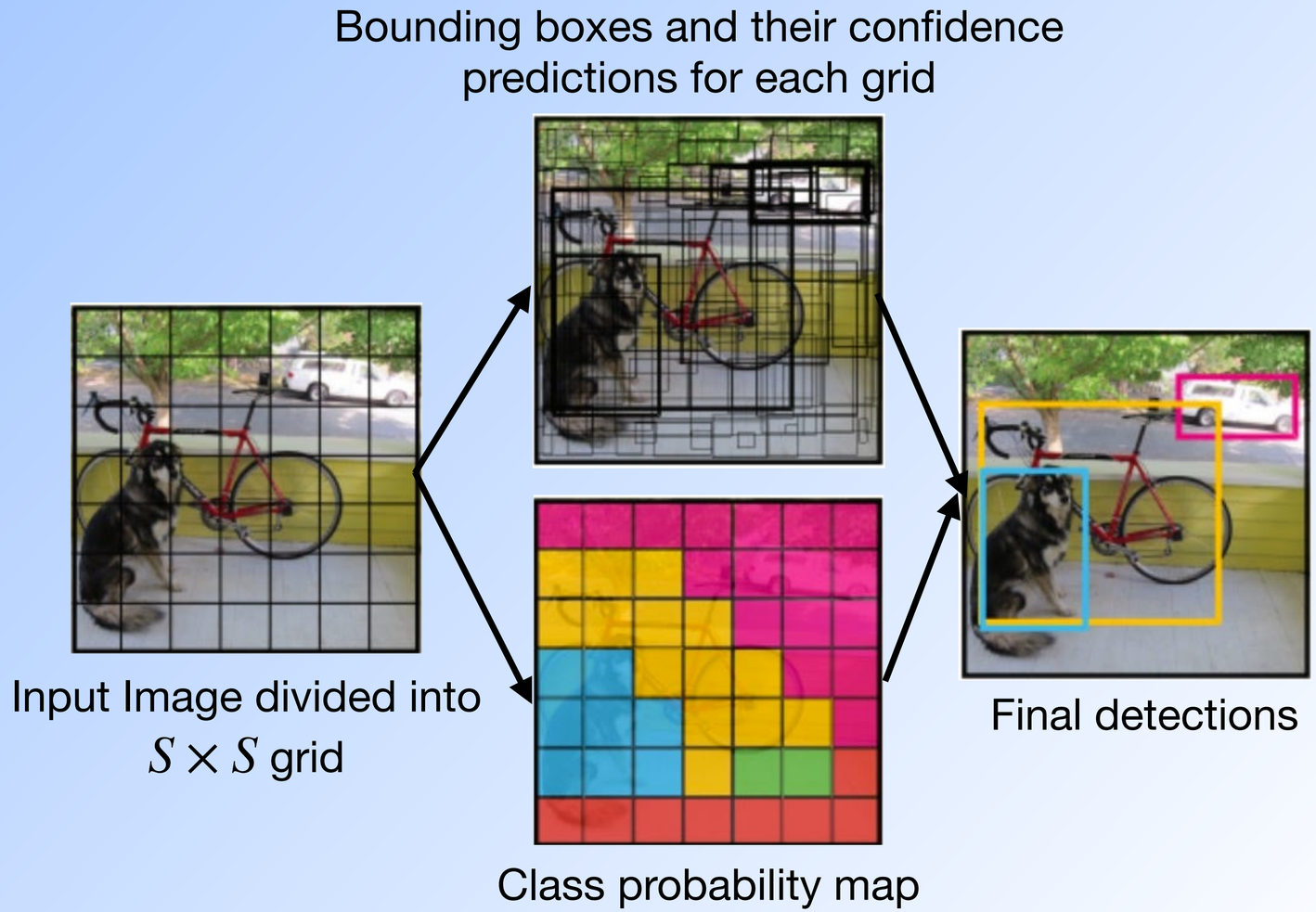
Feature extraction

Object classification and bbox regression

Faster R-CNN Model

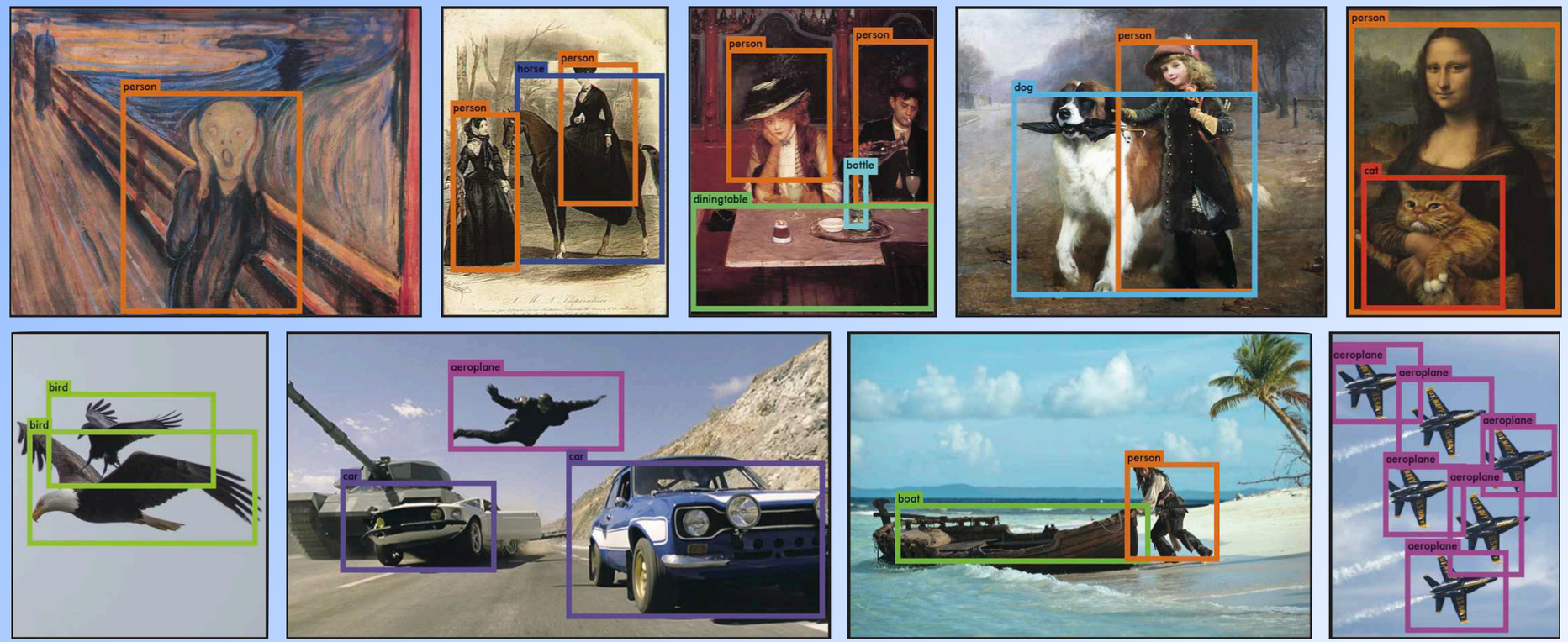
Object Detection Using YOLO

- YOLO (short for You Only Look Once) (Redmon et a., 2016) is a single stage object detector unlike the region proposal based models which achieve object detection in two stages.
- The single stage approach makes the model extremely fast and hence suitable for real time applications.
- YOLO divides the input image into $S \times S$ non-overlapping grids; for each grid, the bounding boxes of the objects whose center falls within the grid, their confidence, and the class likelihoods are predicted.
- The final object predictions are derived by applying NMS on the above predictions.



Object Detection Using YOLO

- The backbone network in YOLO was a 24-layer CNN model pretrained on ImageNet.
- YOLO achieved an inference speed of 21 fps, with mAP of 66.4% when trained using the VGG-16 backbone as compared to 7 fps with mAP of 73.2% achieved by faster R-CNN using the same backbone.



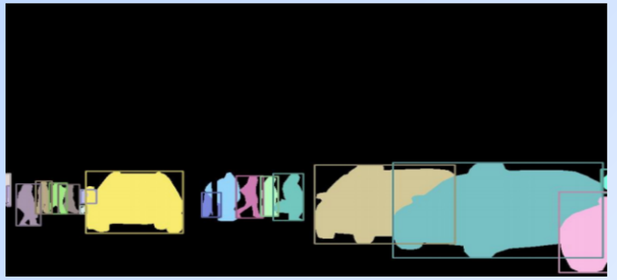
YOLO detections

Image Segmentation

- The goal of image segmentation is to partition images into regions that are semantically uniform.
- Unlike object detection whose goal is to identify the location of objects, segmentation identifies the exact boundary of objects.
- Segmentation could be of the following types:
  - Semantic segmentation - assigns a class label to each pixel in the image
  - Instance segmentation - identifies each occurrence of a class in addition to grouping pixels into classes.
  - Panoptic segmentation - combines semantic and instance segmentation by labeling each pixel and distinguishing between object instances.
  - Boundary based segmentation - focuses on identifying the boundary between different regions, e.g. Foreground vs. background.

Input Image

Semantic Segmentation

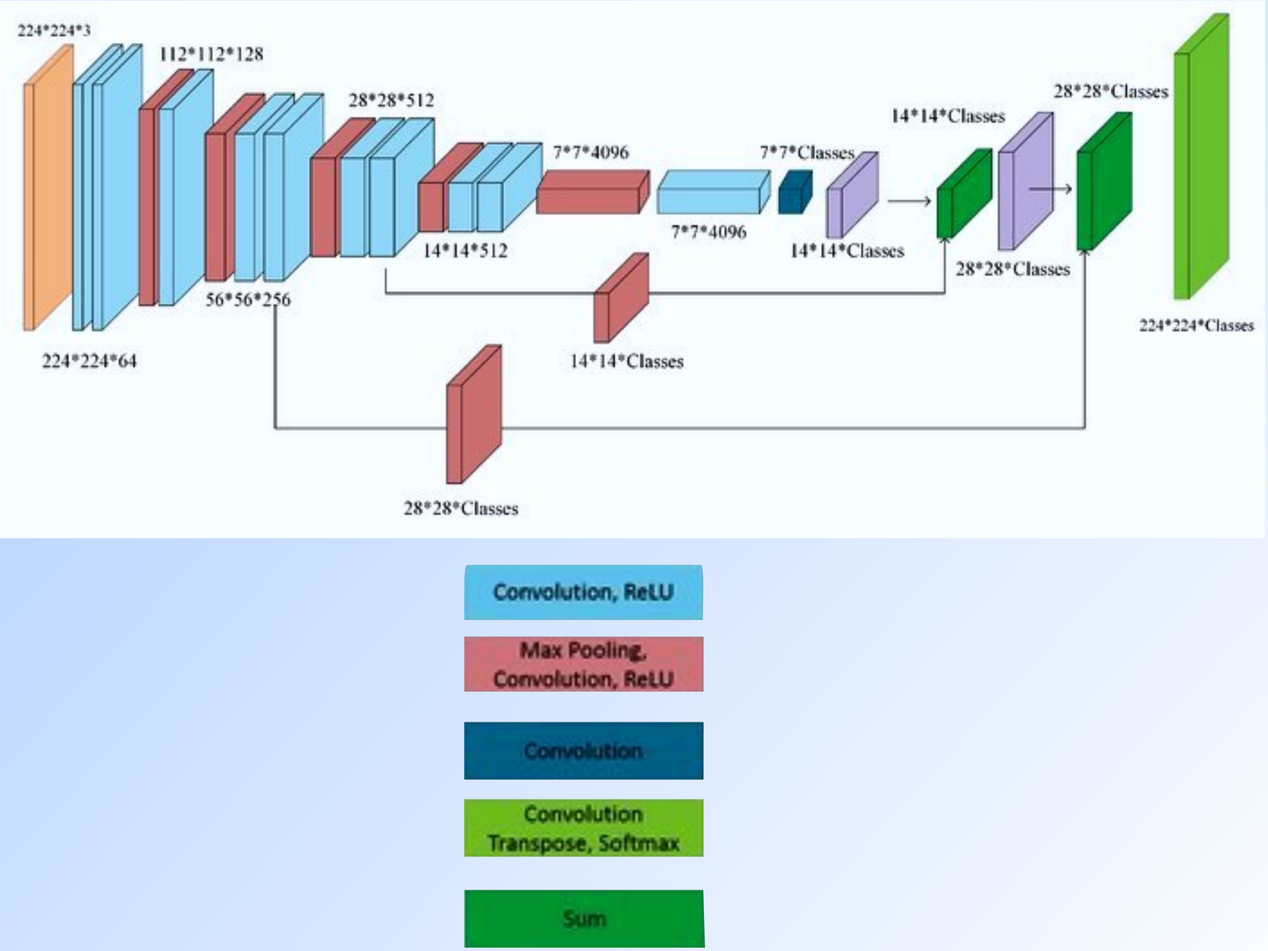


Instance Segmentation

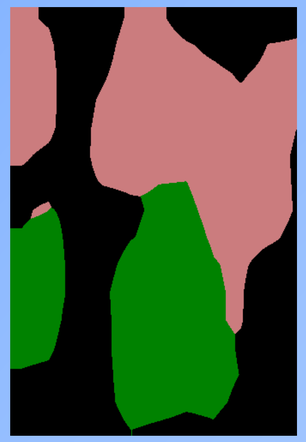
Panoptic Segmentation

Semantic Segmentation Using FCN

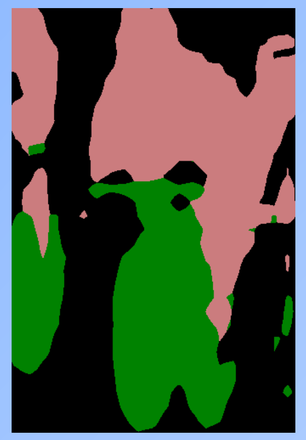
- Fully Convolutional Network (FCN) perform semantic segmentation by replacing fully connected layers with convolutional layers, allowing it to predict segmentation maps for input images of arbitrary sizes (Long et al., 2015).
- FCN used backbone networks such as AlexNet, VGG-16, and GoogLeNet, replacing their fully-connected layers with convolutional layers
- Transposed convolutions were used to upsample the feature map from the last convolutional layer.
- To prevent loss of spatial details, upsampling is done in stages and the feature maps from the shallower layers fused with the output at each stage.



Semantic Segmentation Using FCN

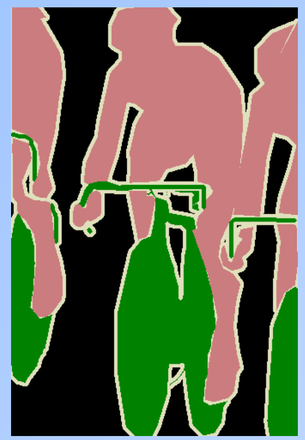


FCN-32s



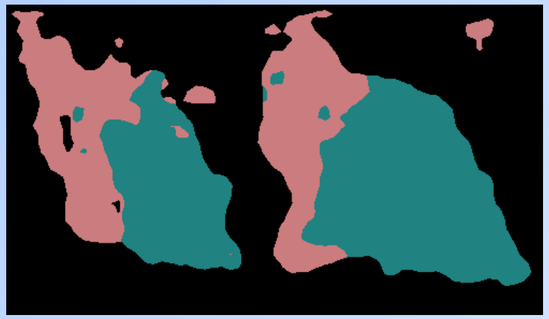
FCN-16s

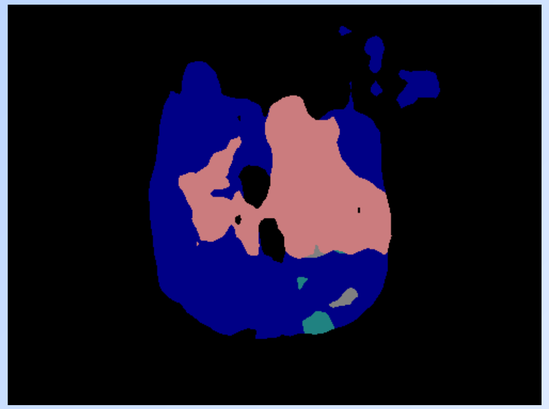
FCN-8s



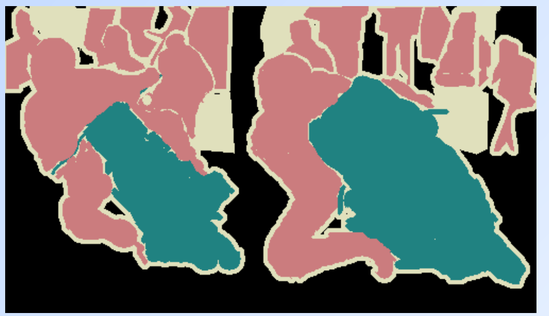
Ground Truth

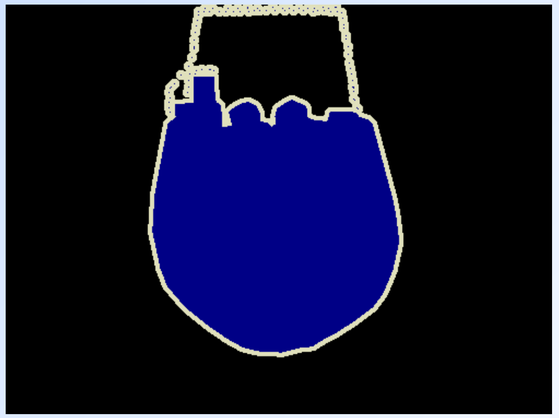
Input





FCN Prediction





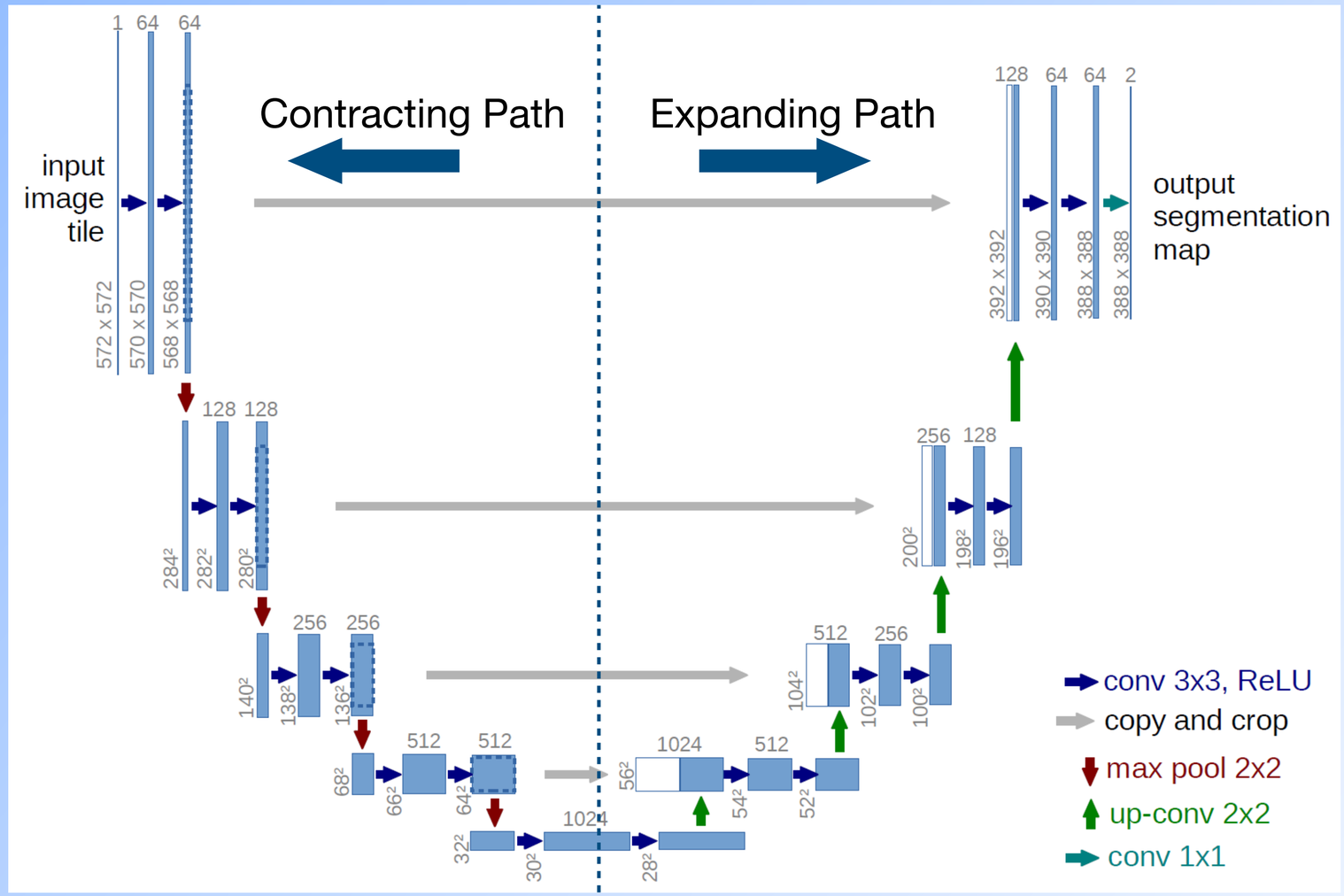
Ground truth

Segmentation Using U-Net

- The U-Net model, originally developed for biomedical image segmentation employs an encoder decoder architecture (Ronneberger et al., 2015).
- The encoder is a contracting path that captures context while progressively reducing the spatial dimensions of the feature map.
- The decoder is the expanding path enables precise localization by progressively increasing the size of the feature map.
- Skip connections are used to concatenate the feature maps from the encoder with the corresponding feature maps in the decoder, allowing the model to retain spatial information that might be lost during downsampling.
- The U-Net is a versatile model that has been used as a backbone for many other image-to-image translation tasks, such as denoising, inpainting, style transfer etc.

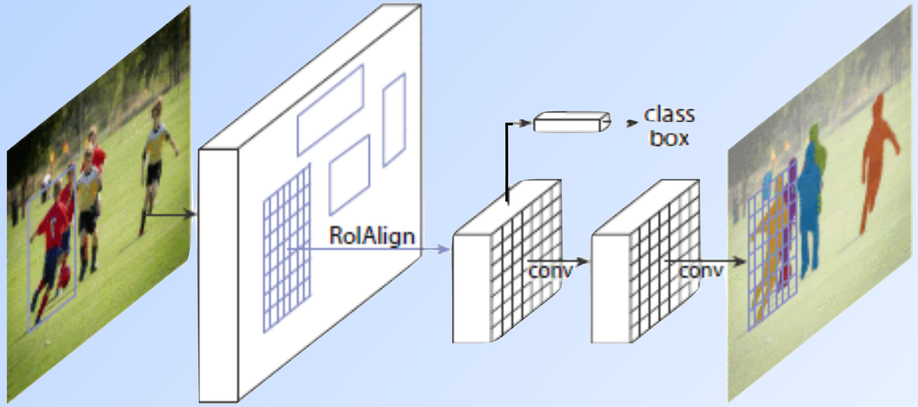
Segmentation Using U-Net

The U-Net Model



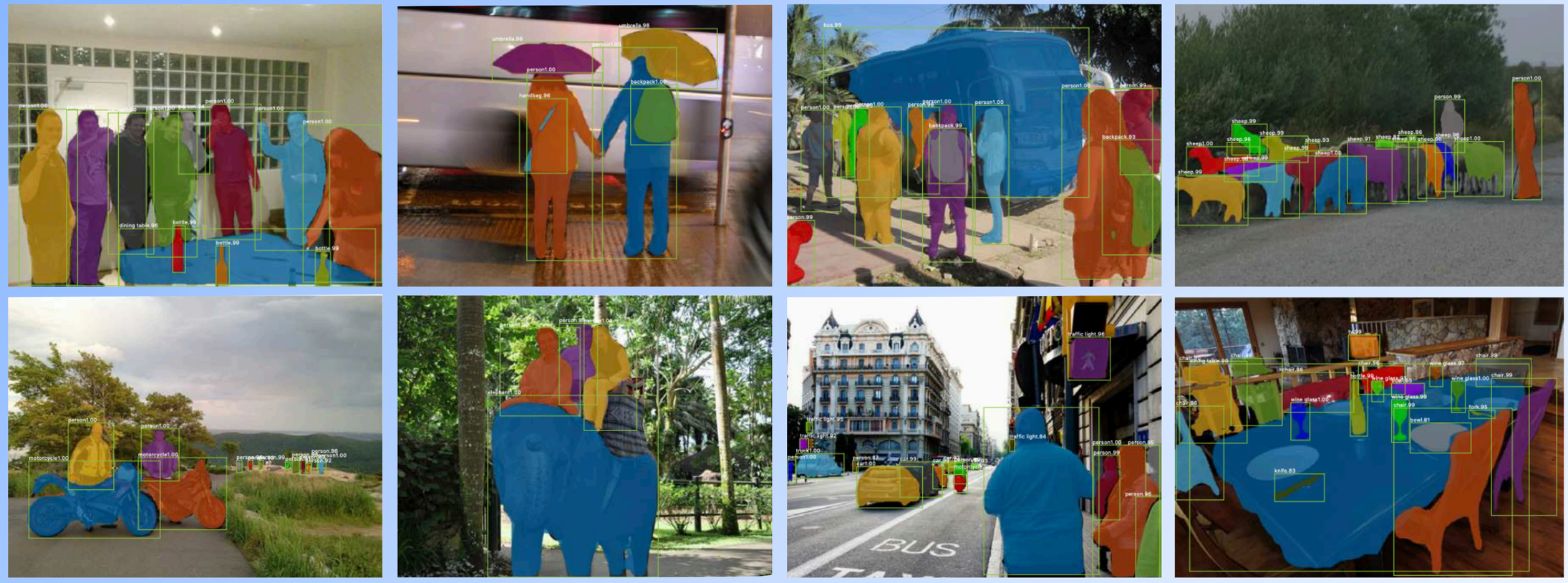
Mask R-CNN for Instance Segmentation

- Mask R-CNN is an extension of the faster R-CNN model, which added a branch for predicting segmentation maps on top of the object detection framework (He et al., 2017).
- Mask R-CNN uses an RoI alignment layer to improve mask predictions.
- In mask R-CNN, the backbone network used was a ResNet architecture combined with a feature pyramid network (FPN) to facilitate multi-scale feature extraction.
- The RoI pooling used in faster R-CNN is not suitable for predicting object masks as it quantizes the RoI coordinates given by the RPN which might introduce misalignment in object mask predictions.
- RoI align addresses this problem by removing the quantization and using bilinear interpolation to compute the feature values at the floating point locations.



Mask R-CNN framework

Mask R-CNN for Instance Segmentation



Mask R-CNN Results