LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Deep Learning

Chapter 6: CNNs - Mask-R-CNN

Mina Rezaei

Department of Statistics – LMU Munich

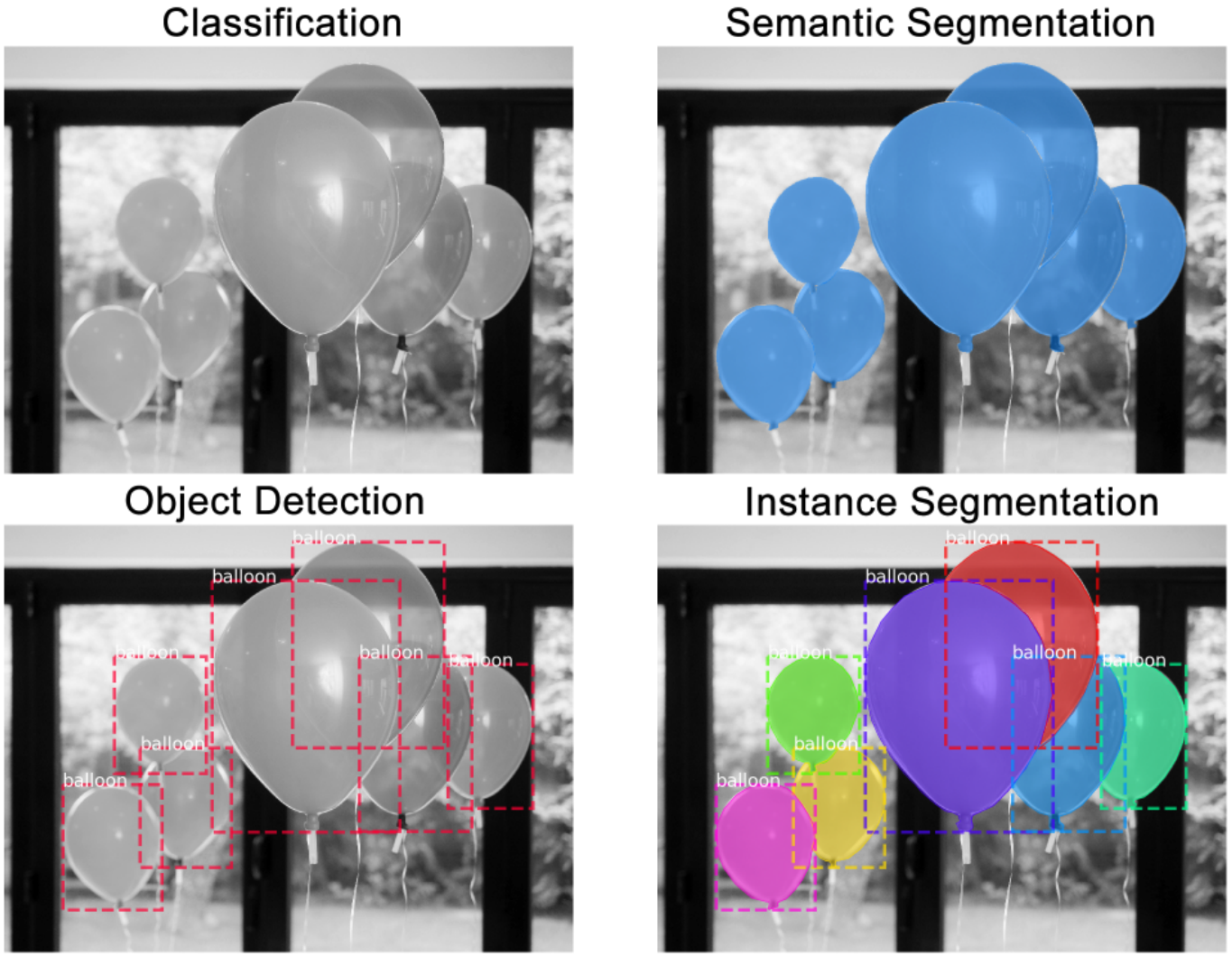
Winter Semester 2020

LECTURE OUTLINE

COMMON MACHINE VISION TASKS

- Image Classification: Assign a *single* label to the whole image.
- Object localization / detection: Draw **bounding boxes** around (and classify) one or more objects in the image.
- Semantic Segmentation: The goal here is to draw **outlines** between classes *without* differentiating between different instances of a given class.
- Instance segmentation: This is a hybrid of the previous two tasks. The goal is to classify *each* object in the image and draw a *separate* outline around it.
- Note: This is not an exhaustive list. There are *many* others.

COMMON MACHINE VISION TASKS



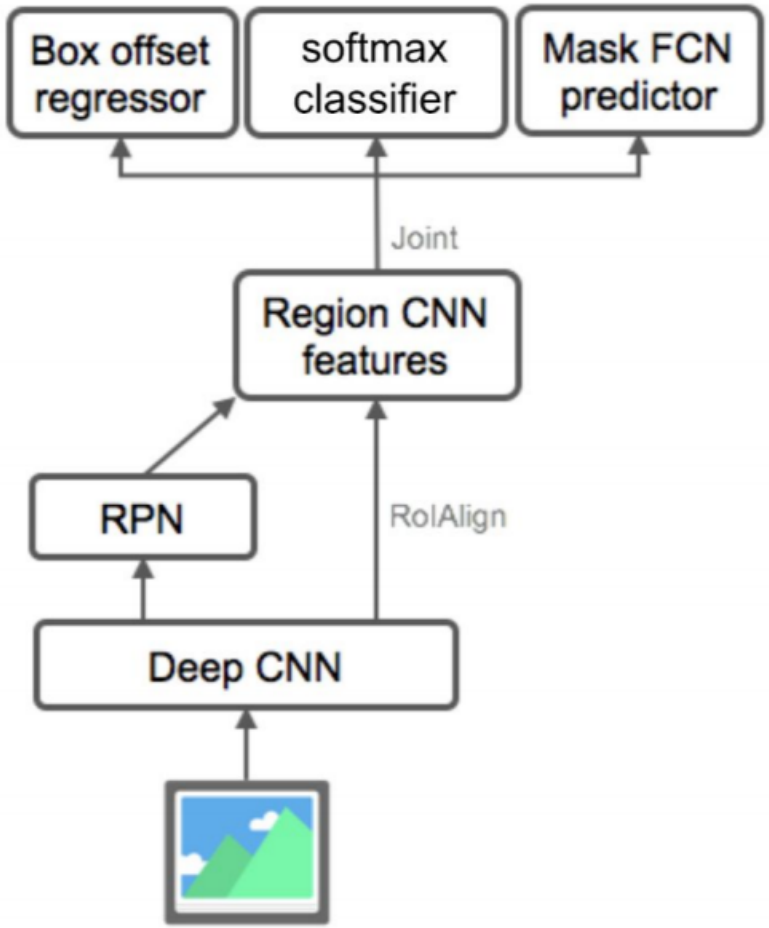
credit: Waleed Abulla

Instance segmentation is the most challenging task!

MASK R-CNN - OVERVIEW

- Mask R-CNN (He et al., 2017) belongs to an important class of CNNs known as Region-based CNNs (or R-CNNs).
- It extends the Faster R-CNN (Ren et al., 2015) architecture (which performs classification and detection) by adding a branch for segmentation (more on this later).
- Therefore, Mask R-CNN performs classification, detection *and* instance segmentation in parallel!
- It was developed at Facebook in 2017 and trained using the COCO (Common Objects in Context) dataset.
- Note: Each training example has ground truth class labels, bounding boxes and segmentation masks.
- This case-study is only meant to be a 'quick tour' of Mask R-CNN. If any of the details are unclear, please read the paper.

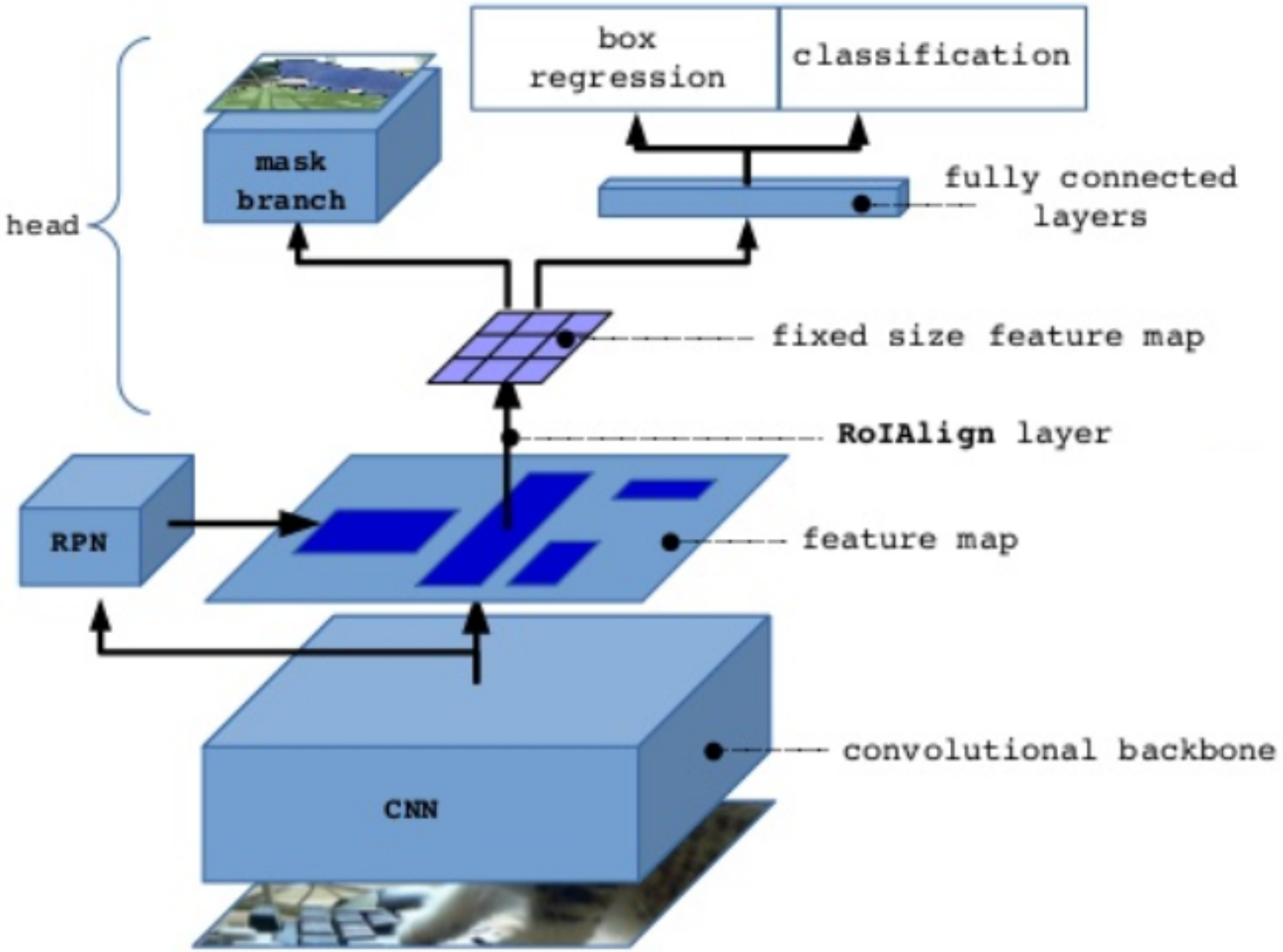
MASK R-CNN ARCHITECTURE



credit: UCSD

Figure: Mask R-CNN architecture.

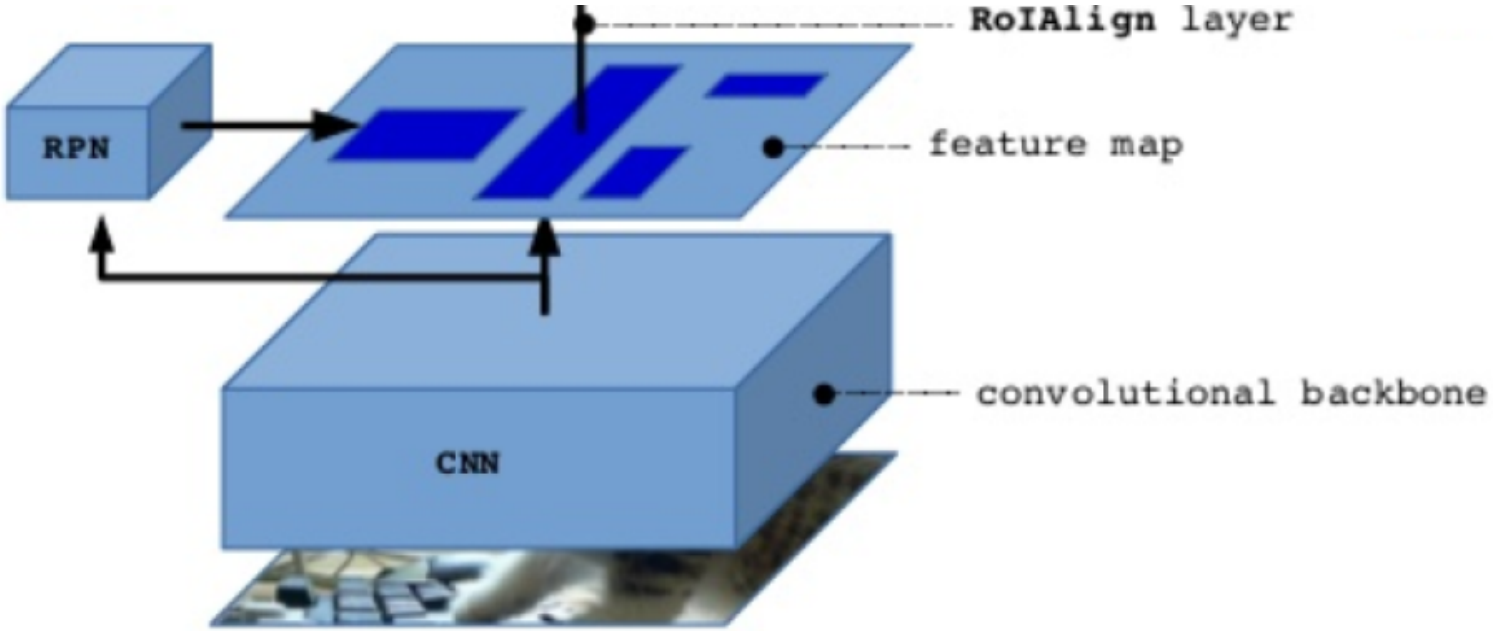
MASK R-CNN ARCHITECTURE



credit: Georgia Gkioxari

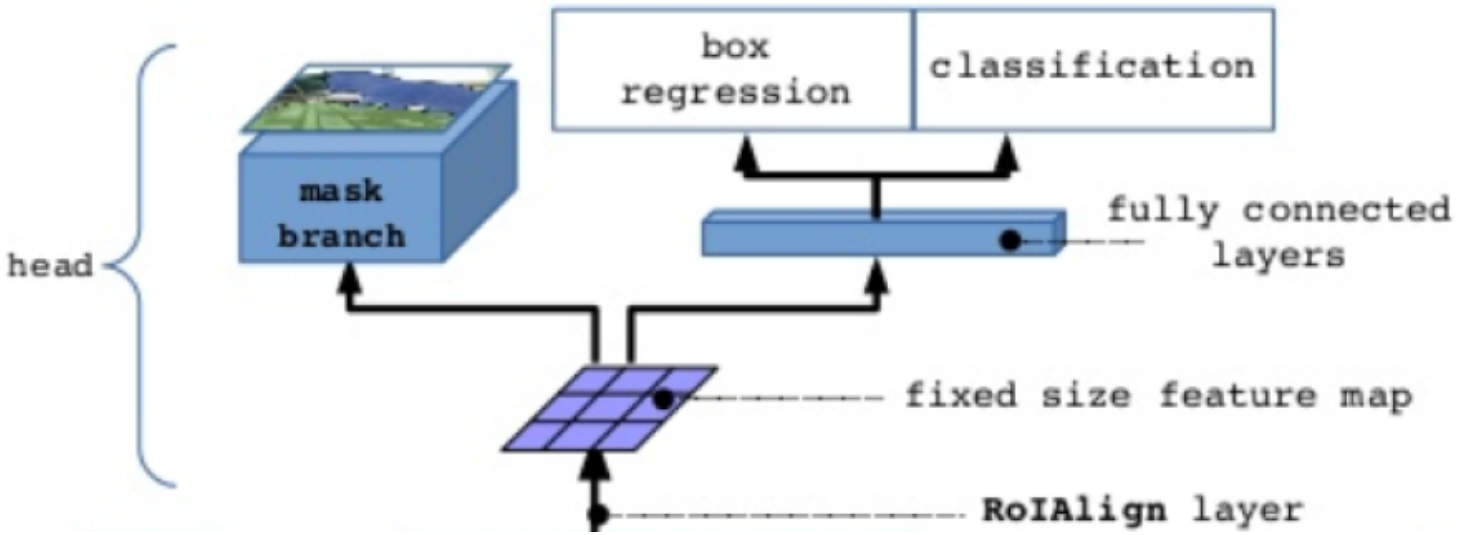
Figure: Mask R-CNN architecture.

MASK R-CNN ARCHITECTURE - STAGE ONE



- *Stage One*: Consists of a "backbone" network and a Region Proposal Network (RPN).
- The backbone network extracts the feature maps from an image and the RPN identifies promising regions, or Regions of Interest (ROIs) that might contain interesting objects.
- The ROIAlign layer then resizes the ROIs so that they all have the same spatial dimensions.

MASK R-CNN ARCHITECTURE - STAGE TWO



- The resized ROIs are then fed to the *Stage Two* network (pictured above).
- This has three branches: one each for classification, detection/localization and segmentation.

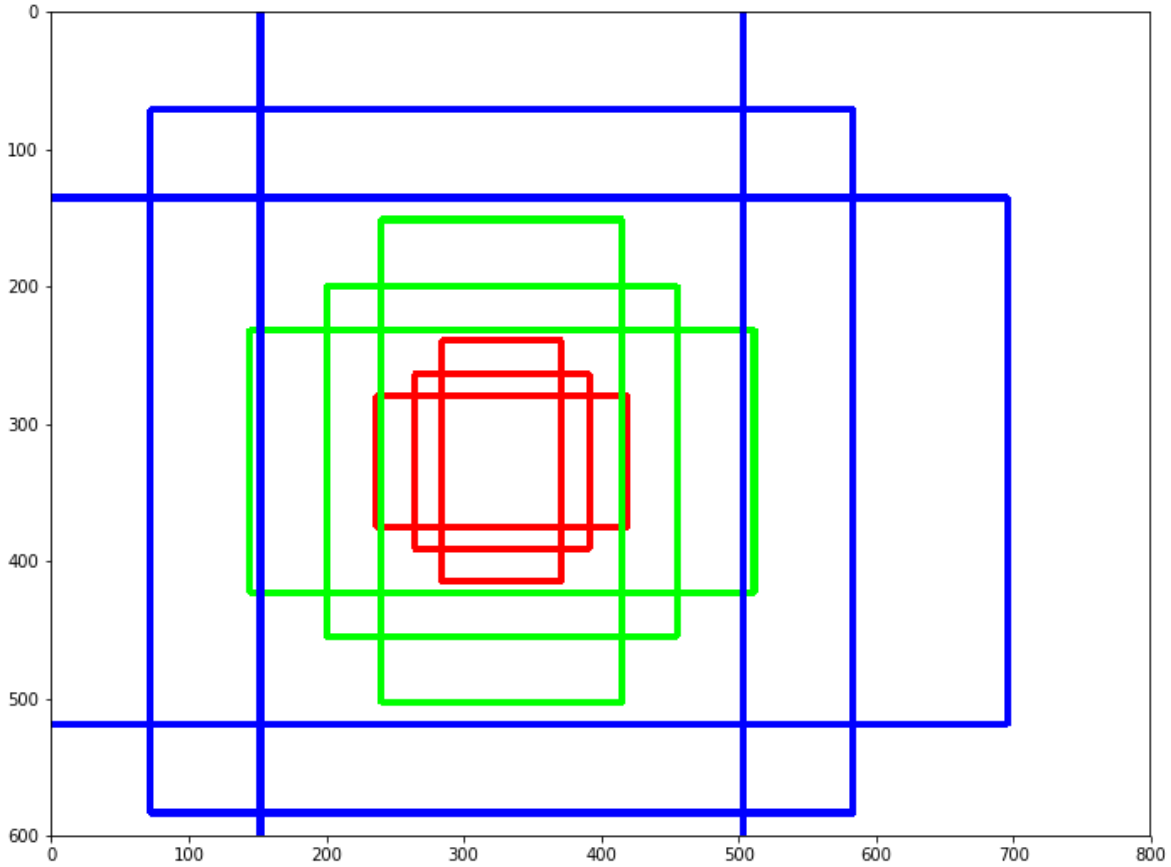
STAGE ONE - BACKBONE

- The backbone architecture can be based on any generic CNN.
- In the paper, the authors implement Mask R-CNN using ResNet.
- ResNets are divided into 5 "stages" where each stage consists of several convolutional layers.
- The authors use the first 4 stages of ResNet-50 or ResNet-101 as the backbone network.
- The output of the backbone network will now serve as the input to the Region Proposal Network.

STAGE ONE - REGION PROPOSAL NETWORK

- Lightweight network that works directly on the final feature map generated by the backbone network.
- Consists of a 3×3 convolutional layer and 1×1 convolutional layers.
- The RPN looks at small $n \times n$ regions of the input feature map (n = 3, in our case).
- At **each** such location, the RPN simultaneously proposes multiple regions of interest (ROIs) that might contain objects.
- In order to accomplish this, each such $n \times n$ region is associated with k **anchor boxes**.
- The number of anchor boxes and their locations, sizes and aspect ratios are all hyperparameters.

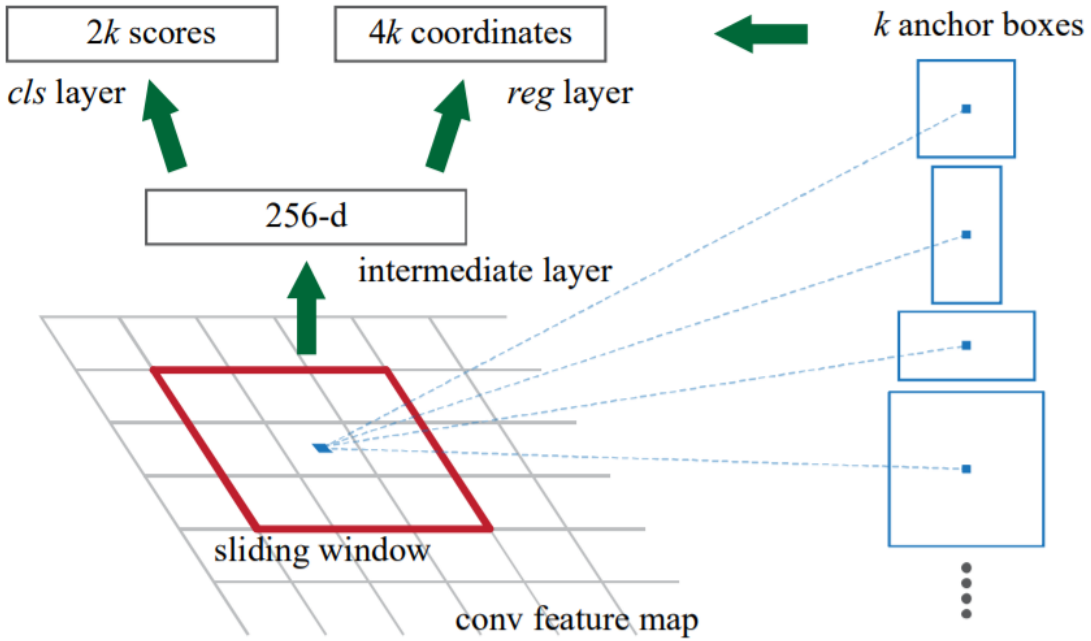
ANCHOR BOXES



credit: Hao Gao

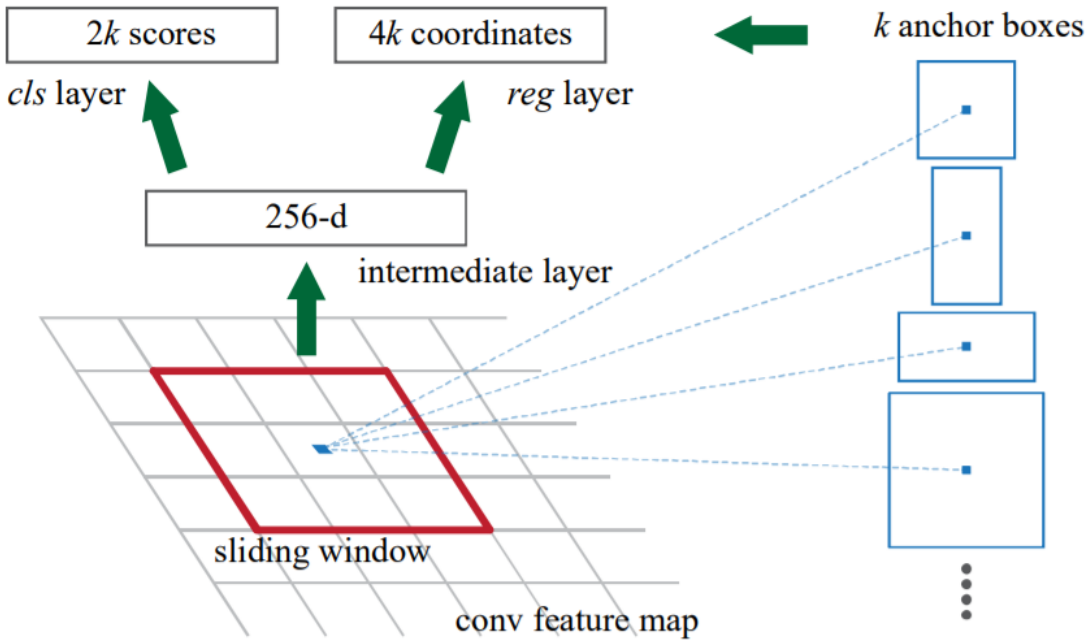
Figure: An example set of anchor boxes in a 600x800 pixel image. Three colors represent three scales or sizes: 128x128, 256x256, 512x512. For a given color, the three boxes correspond to height:width ratios of 1:1, 1:2 and 2:1.

STAGE ONE - REGION PROPOSAL NETWORK



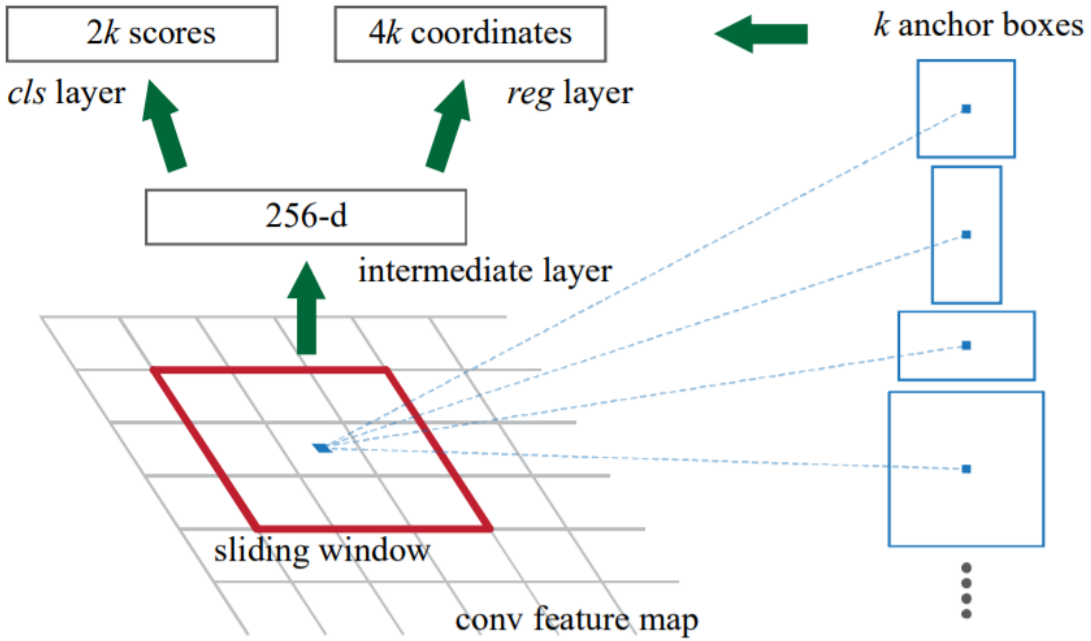
- Every location/'sliding window' in the final feature map is associated with $k = 15$ anchor boxes of varying sizes and aspect ratios.
- For a convolutional feature map of size $W \times H$, there will be $W \times H \times k$ anchor boxes.
- Cross-boundary anchors are ignored during training.

STAGE ONE - REGION PROPOSAL NETWORK



- The classifier ('cls' layer) is a dense layer which outputs $2k$ scores (to which a softmax is applied) indicating the probability of an object being present or *not* present in each of the k anchor boxes.
- Of course, it's also possible to simply output k scores (to which a logistic sigmoid is applied) instead indicating simply whether an object is present.

STAGE ONE - REGION PROPOSAL NETWORK



- The regression ('reg') layer is a dense layer which outputs $4k$ outputs encoding the 4 coordinates of the k region proposals.
- These four values represent the *offsets* to the location of the upper-left corner and the height and width of a (preconfigured, rectangular) anchor box.

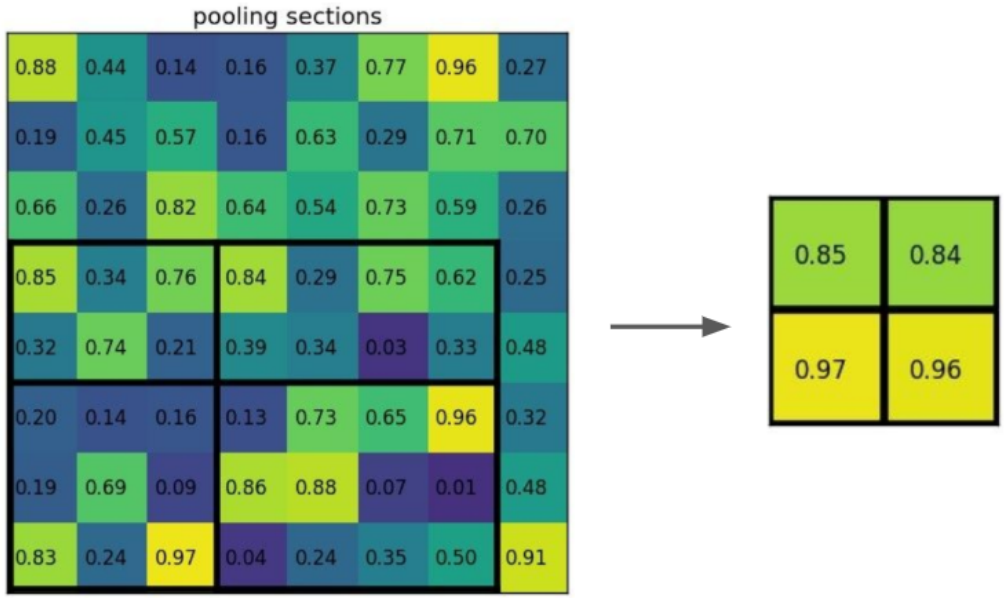
ROI POOLING

- The RPN has its own loss function and can be trained either separately or jointly with the rest of the network.
- Stage One was computing the region proposals. Once we have have them, we enter Stage Two.
- Only the most promising region proposals are chosen for Stage Two using a technique called 'non-max suppression'.
- Please read the paper for details.

ROI POOLING

- The region proposals are in pixel space, not feature space. Therefore, a region of the output feature map (of the backbone network) which roughly corresponds to the region proposal must first be computed.
- These computed regions in the feature map will be of different sizes and aspect ratios. However, the Stage Two network expects a fixed size input.
- Therefore, these regions (in the feature map) must then be *reshaped* so that they all have the same size.

ROI POOLING



credit: deepsense.ai

Figure: A rectangular piece of the feature map which roughly corresponds to a region proposal is further subdivided into four unequal sections.

- To reshape, one option is to divide each rectangular region into unequal sections and perform max-pooling in each section. This is called ROIPool.
- However, the authors implement a more sophisticated method called ROIAlign to perform the reshaping.

STAGE TWO

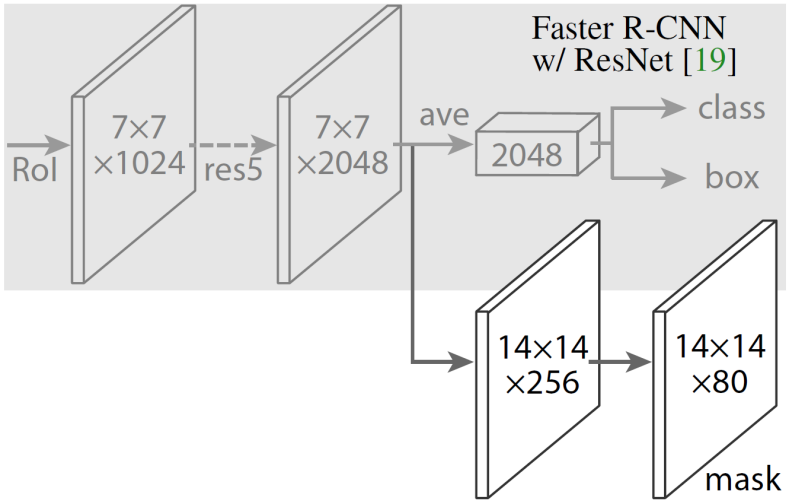
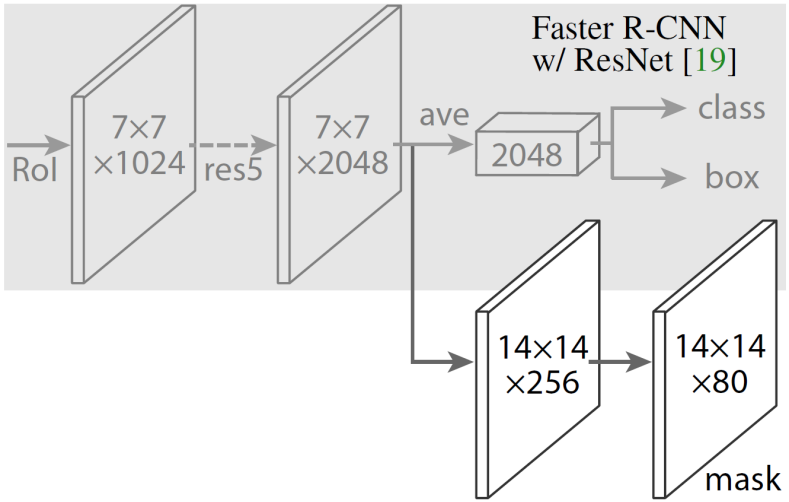


Figure: The three "heads" for classification ('class'), bounding box regression ('box') and instance segmentation ('mask'). The first two heads are also present in the Faster-RCNN architecture from which Mask R-CNN is derived.

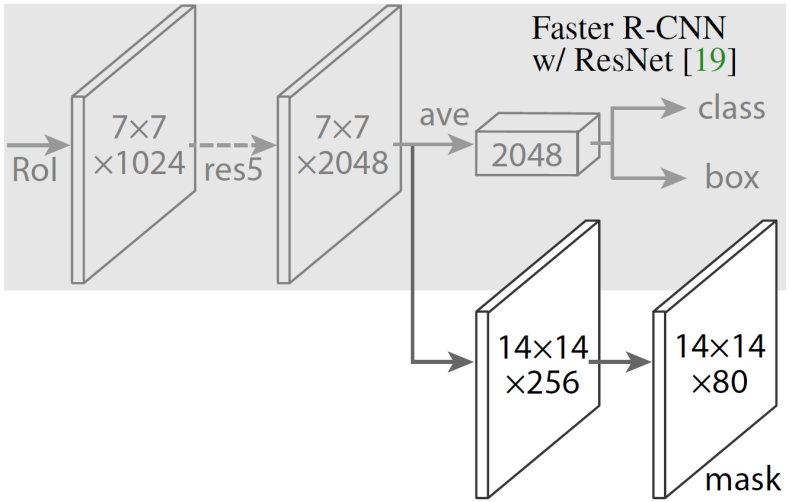
- After reshaping, a $7 \times 7 \times 1024$ ROI is fed to the fifth stage of ResNet ('res5') which outputs a $7 \times 7 \times 2048$ feature map.
- The resulting feature map is then fed to the classification, regression and mask branches.

STAGE TWO



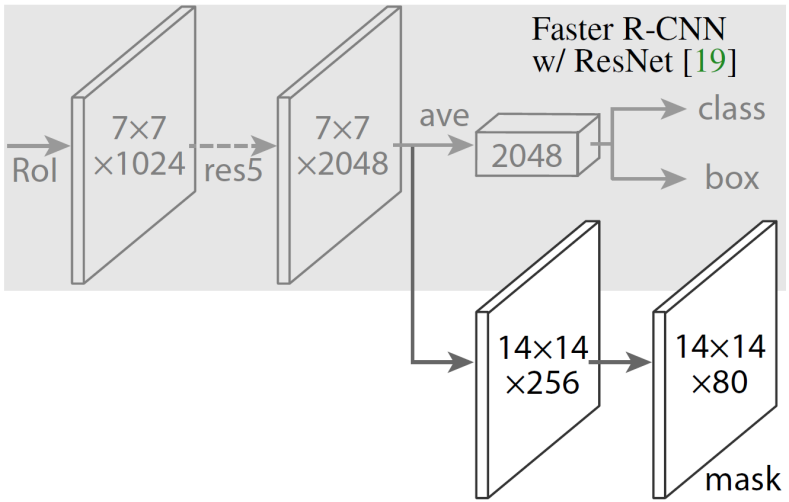
- The $7 \times 7 \times 2048$ output of 'res5' is fed to a dense layer of size 2048.
- The output of the dense layer is then fed to two 'sibling' branches which contain additional dense layers.
- The classification head outputs a vector of length $K + 1$, where K is the number of classes.
- This vector contains the probabilities for each class and, additionally, *no* class / "background" class.

STAGE TWO



- The regression head outputs 4 real valued numbers for each of the K classes.
- Each set of 4 values encodes refined bounding box predictions for one of the classes.

STAGE TWO



- The segmentation head contains a transposed convolution layer which upsamples the 7×7 feature maps to 14×14.
- Finally, the output layer, which is a regular 1×1 convolution layer with a sigmoid activation, outputs a a Km^2 dimensional output ($K = 80$ and $m = 14$ here).
- This encodes K masks of resolution $m \times m$, one for each class.
- The $m \times m$ mask is then resized to the ROI size and binarized (with a threshold of 0.5).

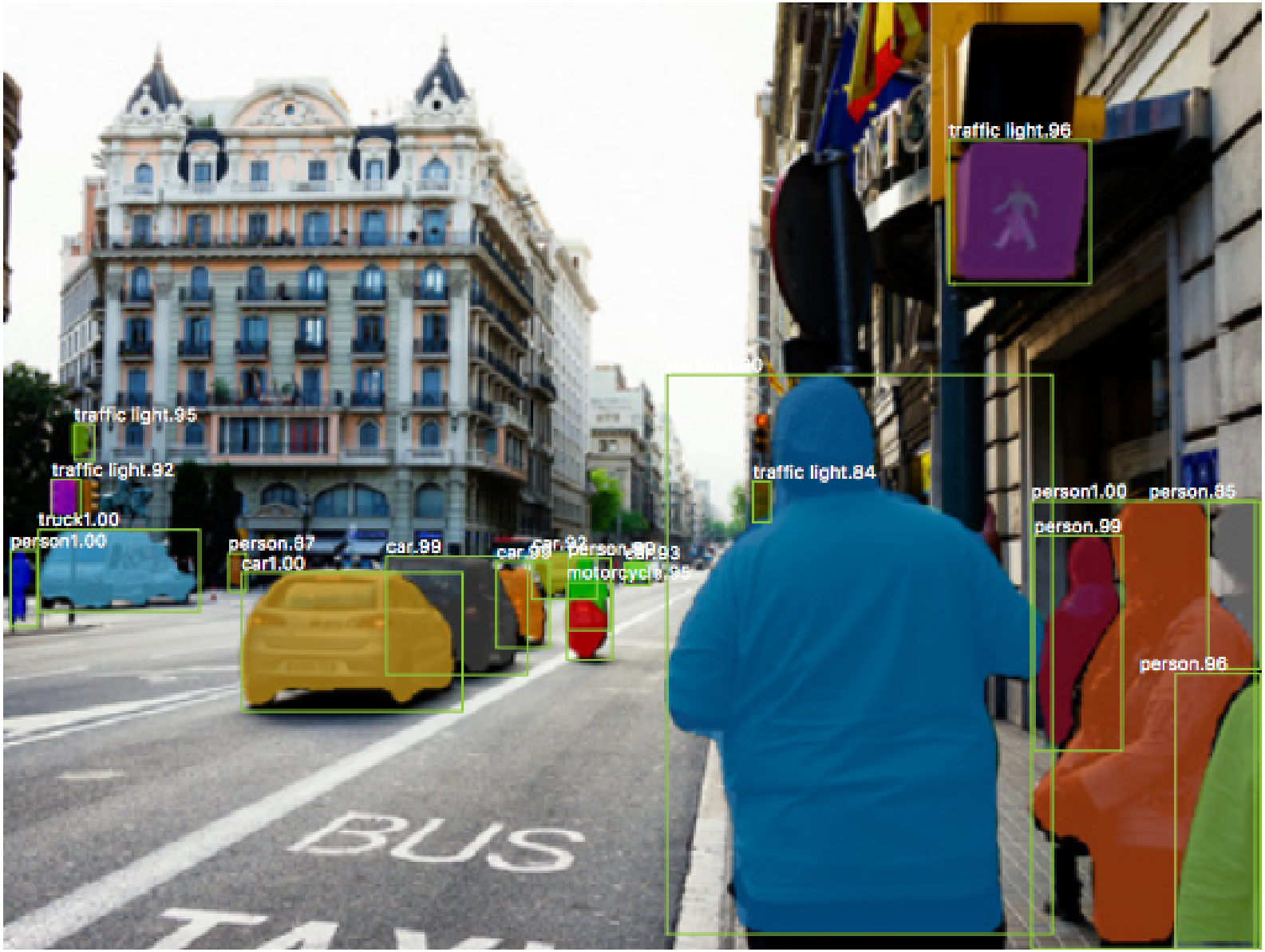
MASK R-CNN - LOSS FUNCTION

- The multi-task loss function of Mask R-CNN combines the losses of the classification, localization and segmentation masks.
- For each ROI, the loss is $L = L_{cls} + L_{box} + L_{mask}$.
- L_{cls} is the negative log likelihood for the true class.
- L_{box} is a modified $L1$ loss between the predicted and true bounding box coordinates (for the true class) which is summed over the 4 values.
- Finally, L_{mask} is the average binary cross-entropy loss.
- For an RoI associated with ground-truth class k, L_{mask} is only defined on the k-th mask (other mask outputs do not contribute to the loss).
- Note: The component losses may be weighted differently.

MASK R-CNN - TRAINING AND INFERENCE

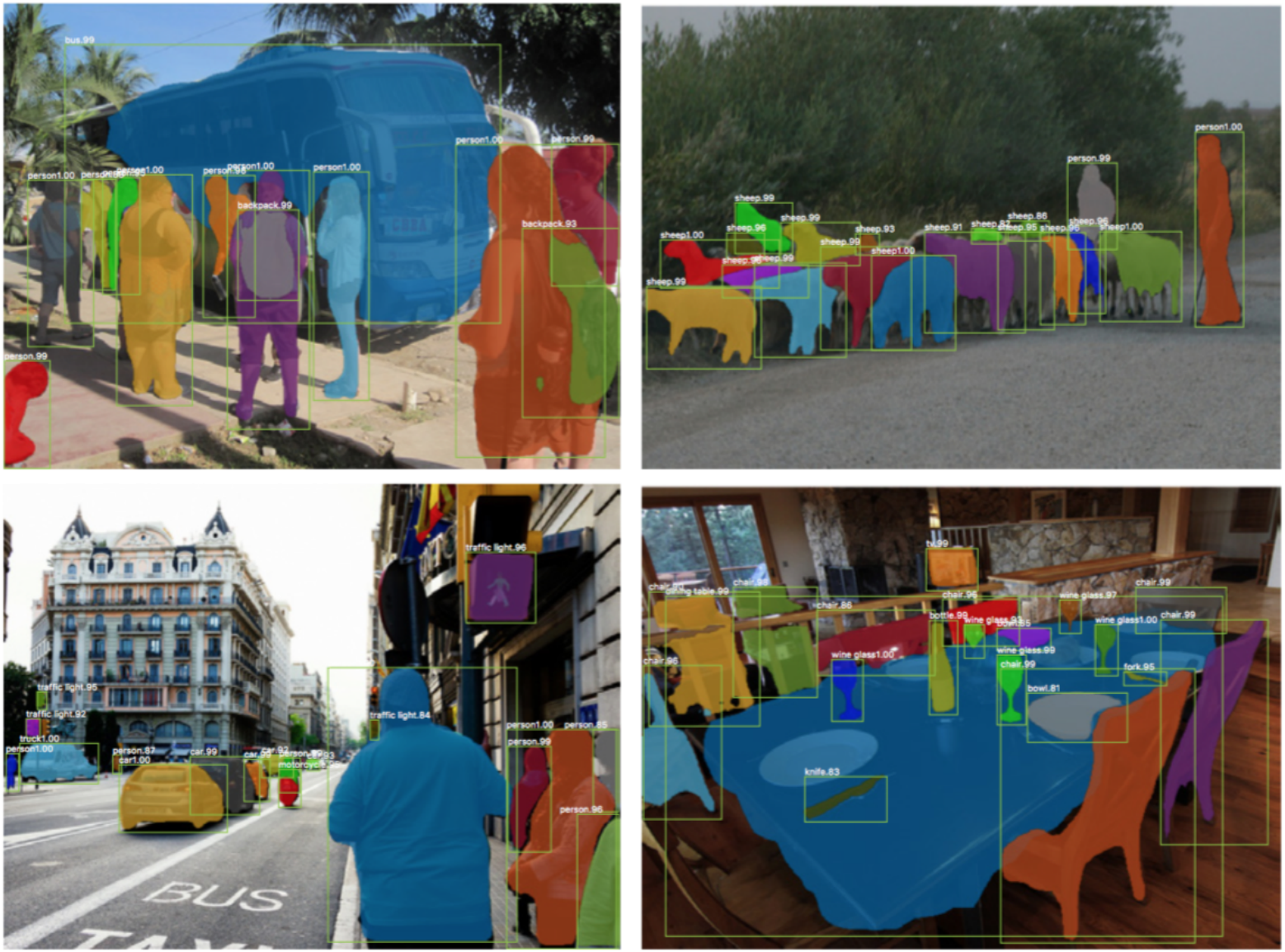
- Dataset: COCO ($\sim$ 100k training images).
- Trained on 8 GPUs with a minibatch size of 16 for 160k iterations.
- Learning rate: 0.02 for the first 120k, 0.002 thereafter.
- Weight Decay: 0.0001, Momentum: 0.9.
- Training time: 32 - 44 hours .
- Inference time: 200 ms - 400 ms on a Tesla M40.

MASK R-CNN - EXAMPLES



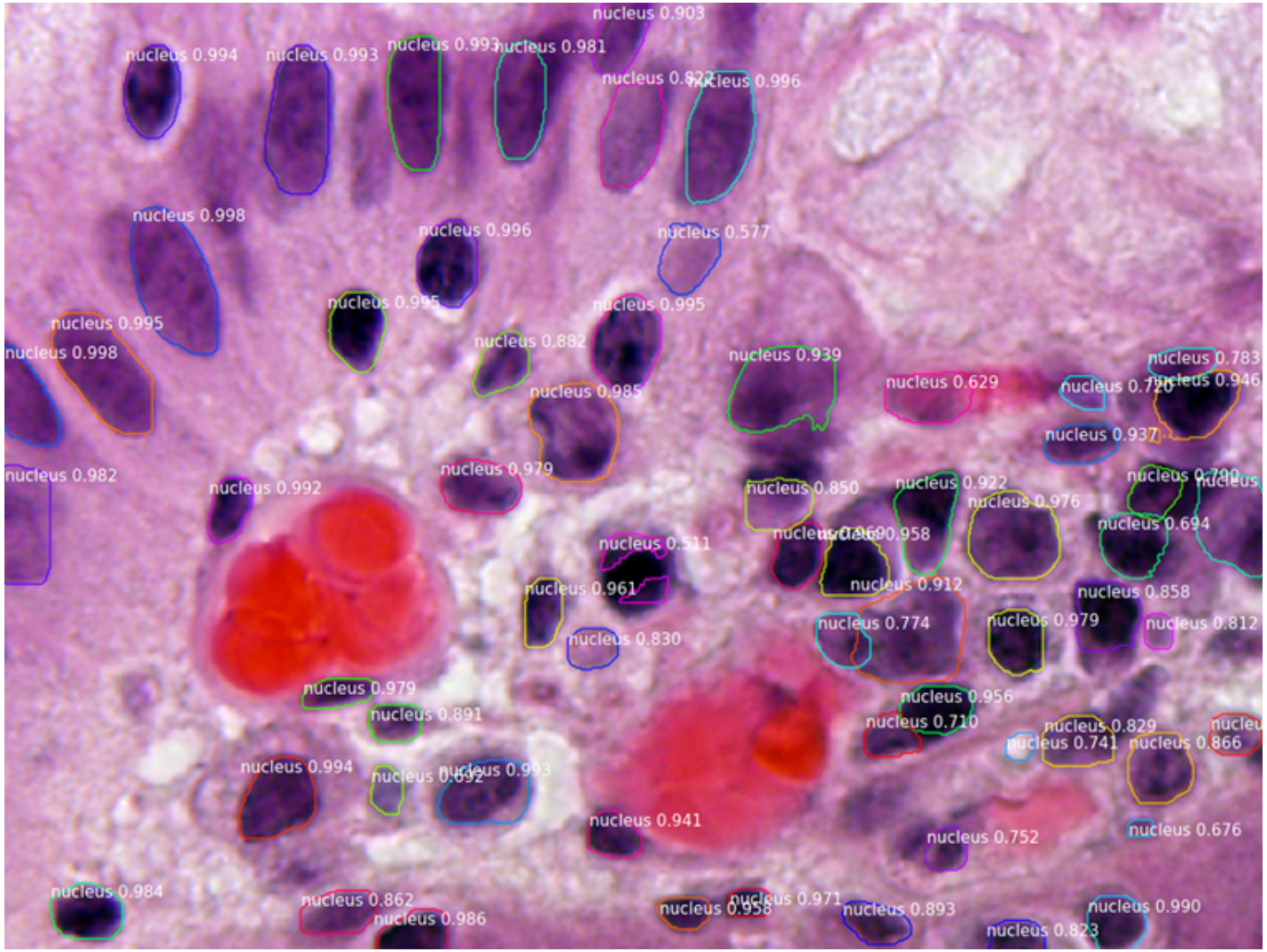
credit: He et al., 2017

MASK R-CNN - EXAMPLES



credit: He et al., 2017

MASK R-CNN - EXAMPLES



credit: Matterport

Figure: Segmenting nuclei in microscopy images.

REFERENCES

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun (2015)

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

https://arxiv.org/abs/1506.01497

Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick (2017)

Mask R-CNN

https://arxiv.org/abs/1703.06870