CS-570 Computer Vision

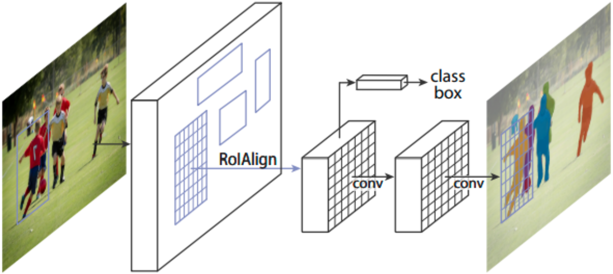
Nazar Khan

Department of Computer Science
University of the Punjab

12. Mask R-CNN

Mask R-CNN

- Comprehensive model for
  1. Object detection
  2. Classification
  3. Semantic Segmentation
- Elegantly combines multiple ideas from CV and DL.

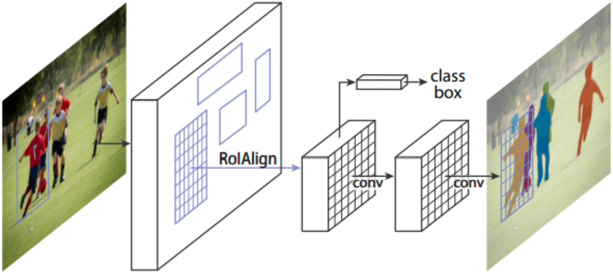


Kaiming He et al. "Mask R-CNN". In: *CoRR* abs/1703.06870 (2017). arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870.

Mask R-CNN
Outline

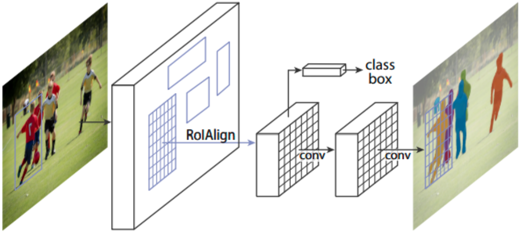
1. Extract features through a CNN.
2. Select image regions *potentially* containing objects.
3. For each potential object region, use CNN features of that region to
   3.1 classify,
   3.2 localize, and
   3.3 segment

   the object.



Mask R-CNN
Outline with terminology

1. **Backbone Network**: CNN for feature extraction.
2. **Region Proposal Network (RPN)**: detects image regions *potentially* containing objects.
3. For each proposed region
   3.1 **Region of Interest Align (ROIAlign)**: Extract backbone CNN features.
   3.2 **Classify**
   3.3 **Predict Bounding Box**
   3.4 **Predict Segmentation Mask**: Pixel level classification into object and background.



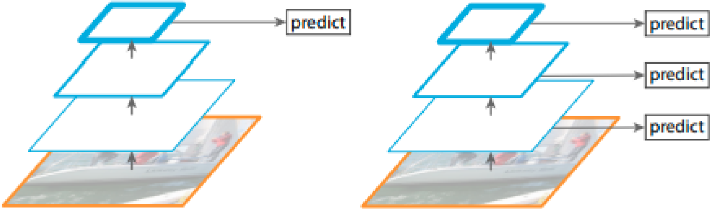
Stage 1: Backbone Network

- Any CNN can work.
- Primary purpose is to extract
  1. multi-scale features that are
  2. rich with meaning (semantics)
- A very good example is the *feature pyramid network*[2].

[2] Tsung-Yi Lin et al. "Feature Pyramid Networks for Object Detection". In: *CoRR* abs/1612.03144 (2016). arXiv: 1612.03144. URL: http://arxiv.org/abs/1612.03144.

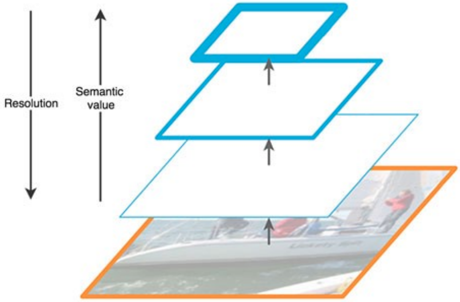
Feature Pyramid

- Recall that a Gaussian pyramid is a multi-scale image representation.
- SIFT descriptors from a Gaussian pyramid represent a *feature pyramid*.
- CNNs are inherently multi-scale because of subsampling.
- But why use the lowest-resolution scale only?



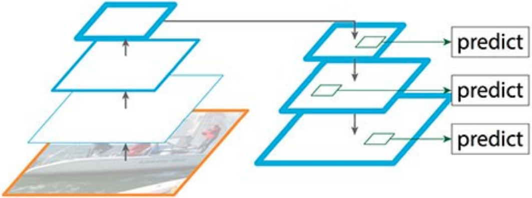
Feature Pyramid

- Deep layers represent *lower spatial resolution* but *higher semantic value*.
- Shallow layers represent *higher spatial resolution* but *lower semantic value*.

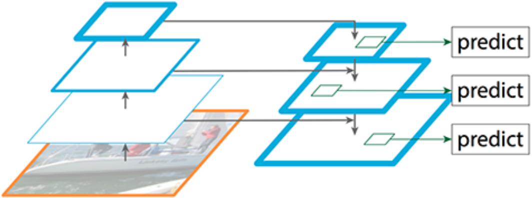


Stage 1: Feature Pyramid Network

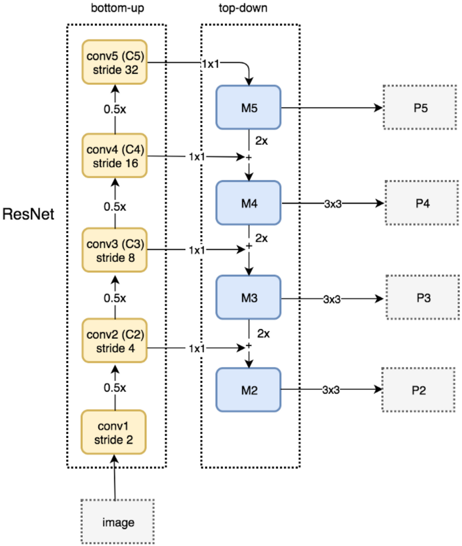
- Recompute high-res features *from* sematically-rich low-res features.



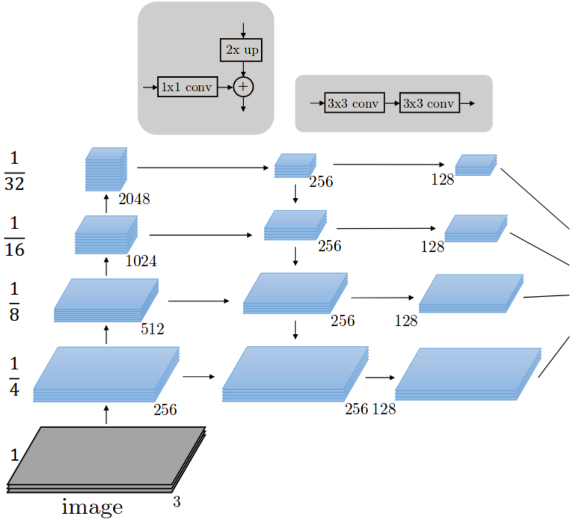
- Add lateral skip connections to *improve localization* and *stabilize training*.



Stage 1: Feature Pyramid Network



Stage 1: Feature Pyramid Network



Stage 2: Region Proposal Network

- A *binary classifier* that *proposes image regions* that can *potentially* contain some object.

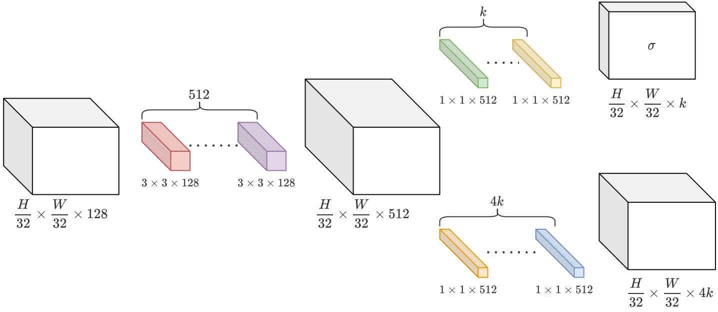
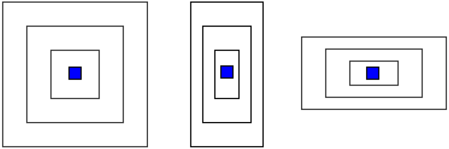


Figure: Region proposal network. Author: N. Khan (2021)

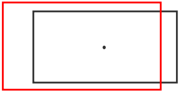
Shaoqing Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.". In: *NIPS*. ed. by Corinna Cortes et al. 2015, pp. 91–99.

Stage 2: Region Proposal Network
Anchor Boxes

- A set of $K = 9$ boxes around one location.
- 3 aspect ratios and 3 scales.



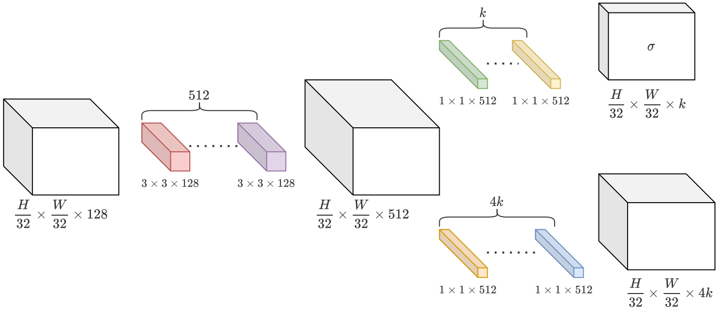
- Implicit assumption: object centered at a location can be covered by one of the K boxes.
- RPN will *refine* these K fixed anchor boxes to cover the object more accurately.



Stage 2: Region Proposal Network
Interpretation of output

For each of the $K = 9$ anchor boxes at each location,

- *Classification head* produces $P(\text{object}|\text{box}_k)$.
- *Regression head* edits/refines the *fixed* anchor boxes.
  - $\Delta_x, \Delta_y, \Delta_w, \Delta_h$
  - predicted $\text{box}_k \longleftarrow$ anchor $\text{box}_k + (\Delta_x, \Delta_y, \Delta_w, \Delta_h)$



Stage 2: Region Proposal Network
Loss Function

- Ground-truth for anchor boxes can be constructed using actual GT boxes

$$p_i^* = \begin{cases} 1 & \text{highest IoU with a GT box} \\ 1 & \text{IoU} > 0.7 \text{ with } \textit{any} \text{ GT box} \\ -1 & \text{IoU} < 0.3 \text{ with } \textit{all} \text{ GT boxes} \\ 0 & \text{otherwise} \end{cases}$$

- Anchors that are neither positive nor negative are not used for training.
- Use cross-entropy loss for classification head

$$L_{\text{cls}}(\{p_i, p_i^*\}) = -\sum_{i: p_i^* \neq 0} p_i^* \log p_i + (1 - p_i^*) \log(1 - p_i)$$

Stage 2: Region Proposal Network
Loss Function

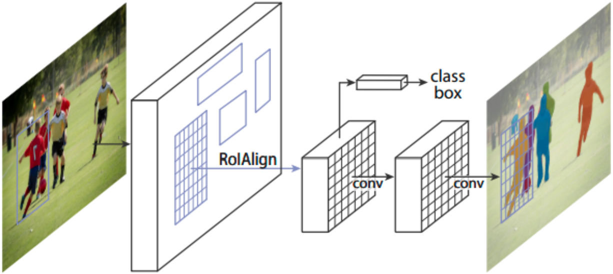
- Use ℓ_1-loss on *positive anchors* for regression head

$$L_{\text{reg}}(\{\mathbf{b}_i, \mathbf{b}_i^*\}) = \sum_{i:p_i^*=1} \|\mathbf{b}_i - \mathbf{b}_i^*\|_1$$

- Overall RPN loss

$$L_{\text{RPN}}(\{p_i, p_i^*\}, \{\mathbf{b}_i, \mathbf{b}_i^*\}) = L_{\text{cls}}(\{p_i, p_i^*\}) + \lambda L_{\text{reg}}(\{\mathbf{b}_i, \mathbf{b}_i^*\})$$

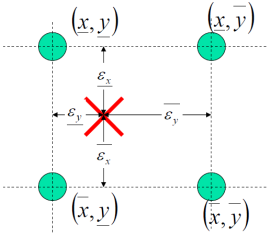
Stage 3: RoIAlign



Stage 3: RoIAlign

Bilinear Interpolation: Lesson from Image Warping

- Recall that for warping images, to find the color at *real coordinates* (x', y') we interpolated from the 4 *quantized* neighbours.

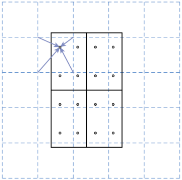


- Color could be a scalar gray-scale value or it could be an RGB vector!

$$I'(x', y') = \bar{\epsilon}_x \bar{\epsilon}_y I(\underline{x}, \underline{y}) + \underline{\epsilon}_x \bar{\epsilon}_y I(\bar{x}, \underline{y}) + \bar{\epsilon}_x \underline{\epsilon}_y I(\underline{x}, \bar{y}) + \underline{\epsilon}_x \underline{\epsilon}_y I(\bar{x}, \bar{y})$$

Stage 3: RoIAlign

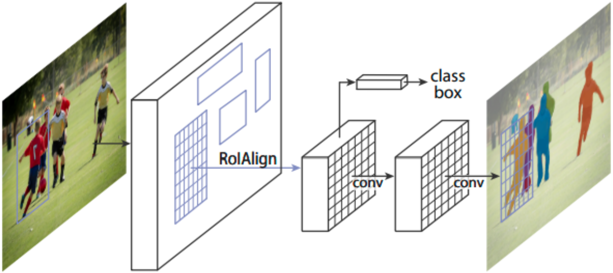
- Given *real* location (x', y'), we can interpolate the CNN feature vector as well.
- RoIAlign step: Given a bounding box with *real* coordinates
  1. Make a uniform grid of (real) locations within the bounding box.
  2. Bilinearly interpolate CNN features for each location.



- Yields a *fixed-size feature volume* irrespective of bounding box size.

Stage 4: Classification, Localization, Segmentation

- From each ROIAlign feature volume,
  - predict class probabilities *via softmax*,
  - predict per-class bounding boxes *via regression*, and
  - predict instance segmentation mask *via logistic sigmoid*.
- Training images contains GT for all three predictions.
- Multi-task loss function.



Summary

- We have covered the architecture of the Mask R-CNN, a state-of-the-art model for multiple CV tasks.
- Design decisions reflect the evolution of CV as well.
- An excellent example of multi-task learning.