“Learning to Segment Object Candidates”

– Pedro Pinheiro, Ronan Collobert and Piotr Dollar

Presented by -

Sivaraman, Kalpathy Sitaraman,
M.S. in Computer Science,
University of Virginia
Facebook Artificial Intelligence Research (FAIR)

FAIR’s AI trinity

- **DeepMask** captures high-level understanding of the general object shape.

- **SharpMask** accurately places the boundaries by going back all the way down to the pixels.

- DeepMask is not very selective and can generate masks for image regions that are not especially interesting - **MultiPathNet** allows information to flow along multiple paths through the net, enabling it to exploit information at multiple image scales and in surrounding image context.
Object Proposals

• First Pre-processing stage

• High Recall (As many objects, in the image)

• High Recall with minimum # of proposals

• Proposals should be accurate with ground truth
Principal idea

Given an image patch:

1. Generate a segmentation mask
2. Likelihood of patch being centered on a full object
• A very large number of binary classification problems.

• First, for every (overlapping) patch in an image: Does this patch contain an object?

• Second, if the answer to the first question is yes for a given patch, then for every pixel in the patch: Is that pixel part of the central object in the patch?

• Use deep networks to answer each yes/no question.

• Networks (and computation) shared for every patch and every pixel to segment all the objects in an image.
Related Work

• AlexNet, GoogLeNet, VGG, R-CNN

• Edge Boxes => By P. Dollar (when at MSR)

• Closely related to MultiBox - generates bounding box object proposals; however, less informative
Too many proposals for the same object.
What we can see.

For us, segmenting objects is very intuitive, given our natural visual training and intelligence.

What the machine sees.

They see only pixels, numbers, DeepMask taps right into the pixels to understand objectness, bypassing low-level features like edges or super pixels.
DeepMask Network
• Predicts Segmentation Mask for a given input patch, assigns score of the patch’s objectness

• Achieved using a single ConvNet

• Last layers are task-specific

• Tasks trained-jointly. Reduces capacity. Increases the speed of full-scene inference at test time.
• For each training sample:
  i) RGB input patch (x)
  ii) Binary Mask (m) : +1 or -1
  iii) Patch objectness (y) : +1 or -1

• y = 1; if patch contains object (roughly in the centre) AND object is fully contained in the given scale range

• y = -1; if not; partial presence not graced either. Mask not considered.
• Down-sample the image from 224x224 to 14x14

• Segmentation: Single 1x1 conv layer + ReLU
  => Pixel Classification Layer

**Segmentation:** The branch of the network dedicated to segmentation is composed of a single $1 \times 1$ convolution layer (and ReLU non-linearity) followed by a classification layer. The classification layer consists of $h \times w$ pixel classifiers, each responsible for indicating whether a given pixel belongs to the object in the center of the patch. Note that each pixel classifier in the output plane must be able to utilize information contained in the **entire** feature map, and thus have a complete view of the object. This is critical because unlike in semantic segmentation, our network must output a mask for a single object even when multiple objects are present (e.g., see the elephants in Fig. 1).

Each pixel is a classifier
• Locally connected (partial view) pixel classifiers vs Fully Connected (redundant parameters) pixel classifiers - both have drawbacks.

• Instead, decompose the classification layer to two linear layers with no nonlinearity in between.

• Output up-sampled to original size.
Joint Learning

• Jointly infer the pixel-wise segmentation mask and object score.

• Loss function used: Binary Logistic Regression

• Binary logistic regression estimates the probability that a characteristic is present (e.g. estimate probability of "success") given the values of explanatory variables.

• Segmentation reduced to a yes/no problem. [ slide 7]
\[ \mathcal{L}(\theta) = \sum_k \left( \frac{1+y_k}{2w^o h^o} \sum_{ij} \log(1 + e^{-m_k^{ij} f_{\text{segm}}^{ij}(x_k)}) + \lambda \log(1 + e^{-y_k f_{\text{score}}(x_k)}) \right) \]

if \( y_k = -1 \): Ignore the segmentation mask

**Segmentation**

**Objectness**

- \( f_{\text{segm}}^{ij}(x_k) \) - Prediction of Segmentation network at location \((i,j)\)
- \( f_{\text{score}}(x_k) \) - Predicted Objectness score
- \( \lambda \) - Score output (= 1/32)
Full Scene Inference

• Applied densely at multiple locations & scales

• Each object has at $\geq 1$ patch where it is central
Implementation details

• Learning rate = 0.001

• SGD, batch size = 32, momentum = 0.9, decay = 0.00005

• Pre-tained VGG features

• Weights randomly initialized

• 5 days to train on Tesla K40m
Training

- Annotations from MS COCO
- Training patches
- Creating Canonical examples
- Jitters, Translations, Flips, upscaled & downscaled

Black box is annotated one. Green ones are canonical positives
Take very canonical positive, and its corresponding mask for each object in the image for training.

Flip it horizontally to augment training data.

Notice, that only the mask of a single is trained as the output, not the entire bounding box.
Results
DeepMask & its Variants

- DeepMaskZoom (DMZ) => Smaller scale; longer inference time
- DeepMask20 => Trained with objects belonging to only PASCAL
- DeepMask20* => same as above, uses DeepMask's scoring network
- DeepMaskFull => Each 56 x 56 connected to CNN
Experiments

- Compared against Edge Boxes, SelectiveSearch, GeoDesic, Rigor, MCG.
- DeepMask + R-CNN => Tops object detection, best Average Recall
- High IoU threshold: other models do better.
- Downsampling the output at every scale likely the reason.
- Can be resolved using skip connections or multiscale approach
Average recall versus number of box and segmentation proposals on various datasets.

AR versus number of proposals for different object scales on segmentation proposals in COCO.
Recall vs IoU threshold on COCO

(g) Recall with 10 proposals.
(h) Recall with 100 proposals.
(i) Recall with 1000 proposals.
Results on the MS COCO dataset for both bounding box and segmentation proposals.
Highlights

• Class-agnostic segmentation plus training on only positive examples (would generate a patch even if not a known object) - generalizes to unseen data as well.

• No usage of low-level features (like edge, superpixels)

• Does not generate BB Proposals, but Segmentation proposals. Can generate BB by finding the closing box around the segmentation.

• More accurate image classification, in the future.

• Learn higher level representations about an image.
The code is Open Source!!
Sources

• https://research.facebook.com/blog/learning-to-segment/

• https://onlinecourses.science.psu.edu/stat504/node/150

• http://mscoco.org/dataset/#download

• Author’s images, from respective personal websites