A review of modern instance segmentation techniques

Author Name: Akshay Tondak
Unique Name: akshayt UMID:23878154

1 Introduction

Instance segmentation is an active field of research with applications in a multitude of image and video analysis use-cases. With the advent of improved GPUs, DNNs have made huge improvements in bettering the accuracy of multiple object detection in both single frames (static images) as well the as collection of frames as videos, snapshots etc. Instance segmentation broadly inculcates two sub-problems, namely object detection and semantic segmentation. Object detection aims to identify the objects present in the frame and semantic segmentation aims to detect boundaries around those object, usually rectangular boxes. Instance segmentation goes one step further and aims to classify each pixel into a fixed category and has wide-reaching uses in upcoming areas of autonomous driving, intelligent video surveillance, remote sensing and thus worth reviewing the novel techniques in this field. Thus, is the topic of survey for this paper.

2 Problem statement

Image segmentation is particularly challenging since it requires precise classification of individual objects and also the classification of each pixel into fixed categories. A typical modern object detection and image segmentation architecture (like YOLO) looks like:

All the methods discussed in this survey play in the latter 3 blocks i.e. tweak the methods used as backbones, or change the path of data flow through networks, or change how the output from convolution layers are perceived and back-propagated to improve on the overall accuracy of either object detection or image segmentation or both.

3 Survey of methods

In this review paper, the following 5 state-of-the art pieces of research have been discussed which have iteratively and significantly improved object detection accuracy over time. "Rich feature hierarchies for accurate object detection and semantic segmentation" (Girshick et al. (2013)) uses static techniques to find potentially important regions and trains DNNs using these individual regions for object detection. "Fast R-CNN" (Girshick (2015)) improves the training time of RCNN using selective search from output feature vector. "Faster RCNN" (Ren et al. (2015)) further improves on training time by completely doing away with algorithmic region detection and using a learning model for region detection. "Masked R-CNN" improves the overall accuracy of object detection and solves image segmentation by predicting a mask for individual objects. Later, "Path Aggregation Network for Instance Segmentation" (Liu et al. (2018)) and "CBNet" (Liu et al. (2020)) fundamentally change how object detection has been done over time by augmenting the backbone networks and achieve even better results.

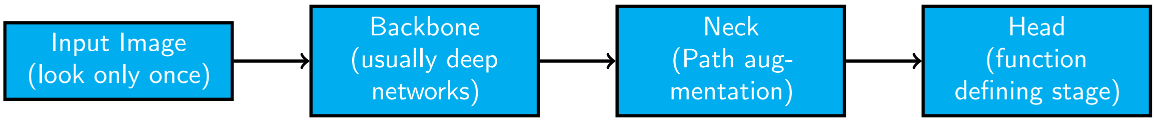

Figure 1: Block diagram of YOLO architecture

3.1 2 stage models (Multi-headed)

These methods come under the family of multi-headed techniques since the output of CNN layers is multi-fold and the model is trained inculcating the minimization of losses corresponding to all the outputs.

3.1.1 R-CNN

R-CNN (Girshick et al. (2013)) is one of the most influential object detection technique based on Deep Neural Networks. It is a region based CNN implementation which works on the idea of finding out 'region proposals' and implementing CNNs on each region parallelly. The original implementation extracts 2000 regions (algorithmically) and then computes features for each proposal using large CNNs. Finally an SVM classifies the object into an appropriate category.
Since the target is 2 folds, the loss which needs to be minimized is also 2 folds:
1. Cross Entropy loss (for classification)

$$CE(y, \hat{y}) = -\sum_{i=1}^{n} y_i \log(\hat{y})$$

2. Mean squared Error Loss (L2 loss for regression)

$$L(x_1, x_2) = 1/n \sum_{i=1}^{n} (x_{1_i} - x_{2_i})^2$$

Localization refers to detecting regions of interest in the image. Szegedy et al. (2013) propose a solution that deals with localization as a regression problem. But the results achieved through their technique did not fare well as they reported a mAP of 30.5% as opposed to >55% achieved by Girshick et al. (2013) on the VOC 2007 dataset. A sliding window approach to solve localization has also been considered by Sermanet et al. (2013) for applications like pedestrian detection. But precise localization in sliding window approach is not always feasible with requirement for high strides and large receptive fields. On top of that, to achieve decent results, higher computation power is required since the classification task is to be perform on every sliding window.Girshick et al. (2013) overcome this problem by resolving to one of their own earlier implementation (Gu et al. (2009)) which works on static images and looks for erratic boundaries in the image though a max margin framework.

Naturally, there are enough ***Drawbacks*** as well for R-CNN. The 'regions of interest' could be different in size, but the convolutional neural network requires images of fixed size. Thus this technique requires all cropped images to be warped or say resizes to a fixed size which potentially leads to loss of information.

SPPnet (He et al. (2014)) fastens up the process of training by almost 3X and test time by 10X. SPPnet computes a convolution feature map for the input image. Then, the region of interest is extracted out of this extracted feature vector instead of the original image.

3.1.2 Fast/Faster R-CNN

Fast R-CNN Girshick (2015) solves the problems in R-CNN by passing the complete image through a convolutional neural network and outputting a feature vector corresponding to the original image (similar to SPPnet). But still the ROIs are carved out and not learnt sequentially. Fast R-CNN is much faster than the classical R-CNN, so much so that the bottleneck now was the calculation of regions of interest which took more time then actually training the model. Faster R-CNN Ren et al. (2015)completely obsoletes the usage of algorithm based region of interest searching and proposes a pure neural network based ROI search (called the RPN).
RPN is a separate network which works in parallel and takes as input a raw image(normalised) and outputs rectangular potential regions with their objectness scores. In faster R-CNN, a sliding window approach is taken, but on the output feature map of a dense conv network instead of the original image as in RCNN. The RPN is capable of predicting what the implementation calls **anchors**. Anchors are nothing but regions of multiple sizes and aspect ratios carved out from the output feature vector. At any

given position of the sliding window, the network proposes 'k' different regions of proposals. Every region proposal has a size and aspect ratio associated with it. Ren et al. (2015) fix the number of possible values for size and aspect ratios to be 3 and 3. Thus for each sliding window, there are 'k'=9 anchors possible. For a feature map of size W X H, the number of anchors is given by W*H*k. Refer to 5 for a detailed view.

Loss Calculation

The loss function defined by Ren et al. (2015) includes the sum of two losses :

$$L = 1/N_{cls} \sum_{i=1}^{n} L_{cls} + 1/N_{reg} \sum_{i=1}^{n} p_i^* L_{reg}$$

$p_i - > Predicted\, probability$
$i - > Anchor\ Index$
$p_i^* - > ground\, truth\, probability(1\ for\ positive\ anchor\ and\ 0\ for\ negative\ anchor)$

The RPN here is taken to be a binary classifier in the sense that each anchor contains an object or it is does not. Consequently, for Faster RCNN consists of two separate SVM heads one for classification and other for regression. Faster R-CNN implemented what is called a "single step training pipeline". In this , the two two heads talked about above are trained simultaneously to improve on training time. Faster R-CNN is a 2 step implementation: The first stage is called the RPN (region proposal network). The output for this branch of CNN is a proposal on bounding boxes. The second stage extracts out the features using ROI pool for each candidate region proposal calculated in the previous step.

Result

Faster R-CNN is more of a speed improvement than an accuracy improvement. Fast R-CNN improves the training time for image for VGG16 by 140x and in terms of time, it brought the training time down to 9hrs as compared to 84hrs before.

3.1.3 Mask R-CNN

Mask R-CNN (He et al. (2017)) is an extension of Faster R-CNN. Mask R-CNN adds a fully connected layer for each Region Of Interest detected in the frame. This fully connected layer is termed as the segmentation mask, hence the name Mask R-CNN.

What is a Mask ?

He et al. (2017) define mask as an m X m matrix which encodes the spatial information of the mask. The requirement for keeping this as an mXm matrix instead of a vector comes from the fact that the spatial representation for every mask needs to be kept instead of just a bounding box. This, apart from keeping just the pixel values, mask representation is able to maintain the pixel to pixel relationship as well. ***Training Architecture***

Mask-RCNN is architected in the same vein as Fast/Faster R-CNN. In addition to the original output of Fast RCNN, Mask RCNN also outputs a binary mask for each ROI. Now, the multi-task Loss if given by :

$$L = L_{cls} + L_{box} + L_{mask}$$

The classification and box losses are similar to those calculated in faster R-CNNs. The mask loss is what is calculated additionally for each region. Since the mask loss is calculated for each class. This means if the class predicted in that of a dog, then the mask would be different from what it would be when the class predicted for ROI is a cat. This decouples the predictions of class and masks. 2 visualizes the architecture in greater detail.

ROI Align

ROI align is the standard method of extracting out regions from the output feature map. This requires a floating point quantization. Further, the extracted ROI is again divided into sub-parts requiring further quantization. This leads to 2 problems. 1. There is loss of information due to decreasing size of ROI. 2. This also sometimes leads to having addition of extra information which is not meant to be a part of the region of interest.Kemal (2020) explains ROI align in greater detain which is a technique of overcoming this loss of information through interpolating subregions of the ROI when picking out the ROI from extracted feature vector.

3.2 Backbone techniques

These techniques alter the backbone architecture of classic object detection techniques and improve the accuracy of object detection.

3.2.1 Path Aggregation Network

Liu et al. (2018) explores a further improvement on the state of the art Mask RCNN. They point out 2 potential scopes for improvements. 1. Low level features contain plenty of information on segmentation i.e. individual objects and well highlighted in the initial layer of the network. But the convolutional path from initial layers to the final output is long and this information is mostly lost. 2. Mask predictions are made based on a single view of the image. The key motivation behind this technique is that high layers contain features that correspond to overall frame whereas lower layers correspond to localized features important for segmentation. This idea is used too further improve the process of predicting masks. PANet is also the architecture employed in the famous YOLO technique's version 4 and is an advancement of already matured feature pyramid networks (FPNs) (Bochkovskiy et al. (2020))
Architecture Liu et al. (2018) augment a bottom up path to make bottom layer information propagation more feasible. They create an architecture which is able to access information from all levels and enhances prediction. This was done based on the fact that strong low level patterns are strong indicators of edges and boundaries and effectively localise features. The paper calls it a "shortcut" consisting of no more than 10 layers. PANet also uses the state of the art ROI align technique to fuse the output of all bottom up layers' outputs. 3 gives a better picture of this architecture.

3.2.2 CBNet

CBNet (Liu et al. (2020)) aims to improve the object detection accuracy together by altering the backbone of the whole detection pipeline. The idea implemented by Liu et al. (2020) is to inculcate the advantages of multiple pre-trained backbones (like ResNet (He et al. (2016)) and ResNext). This is done by using multiple backbone architectures in cascade. Thus the high level output of one backbone network is fed as the input the next-in-cascade backbone network. The motivation also comes from the fact that designing and pre-training a novel architecture is computationally expensive to achieve minor gains in the detection accuracy.
Architecture
CBNet preconditions the requirement of more than 2 cascaded backbone networks. The architecture divides the type of backbone network to 2 subcategories namely Assistant backbones and Lead Backbones. Assistant backbones are tasked to enhance the features of Lead Backbones Each backbone is further divided into 5 stages with each stage comprising of multiple convolutional layers. CBNet can be better understood governed by the flow equation :

$$x_k^j = F_k(x_k^j - 1 + g(x_{k-1}^l))$$

The input of j$^{\text{th}}$ stage of backbone takes input from j-1$^{\text{th}}$ stage and output of the parallel stage of the prvious backbone. The connection from previous parallel stages is termed as composite connection. 4 explains this concatenation in greater detail.

[1]

[1] Chahal and Dey (2018), `https://towardsdatascience.com/`, `https://medium.com/` were also referred for this literature review

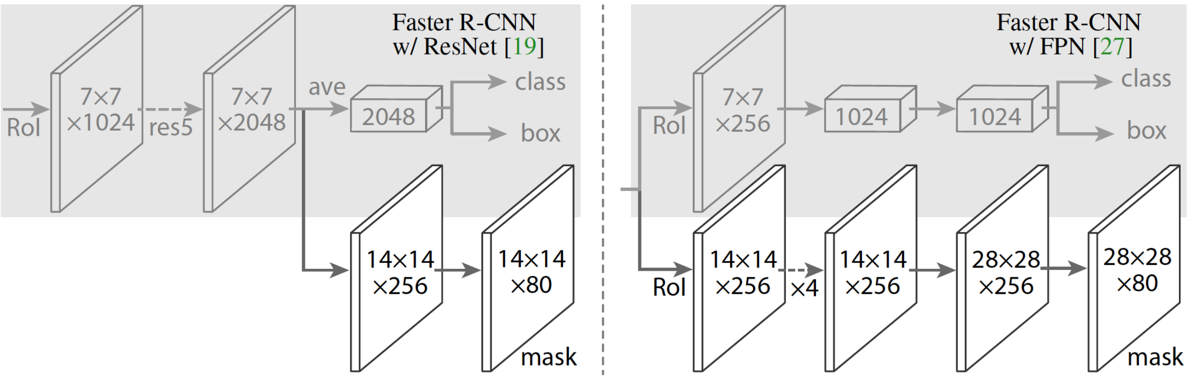

Figure 2: Mask RCNN architecture

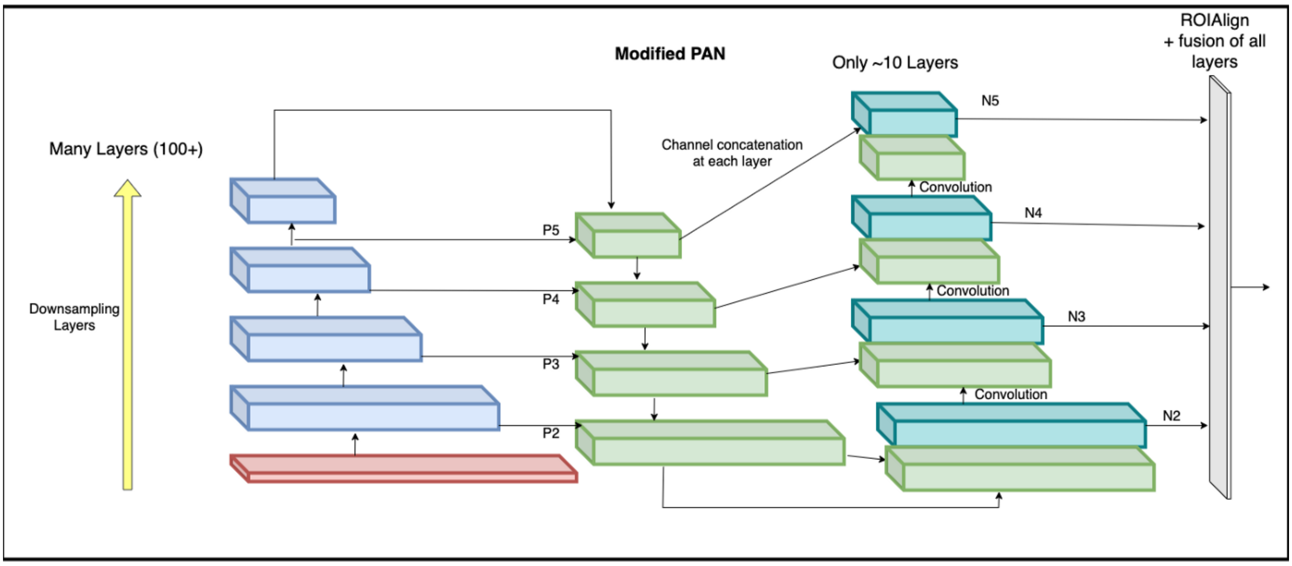

Figure 3: Path Aggregation Liu et al. (2018)

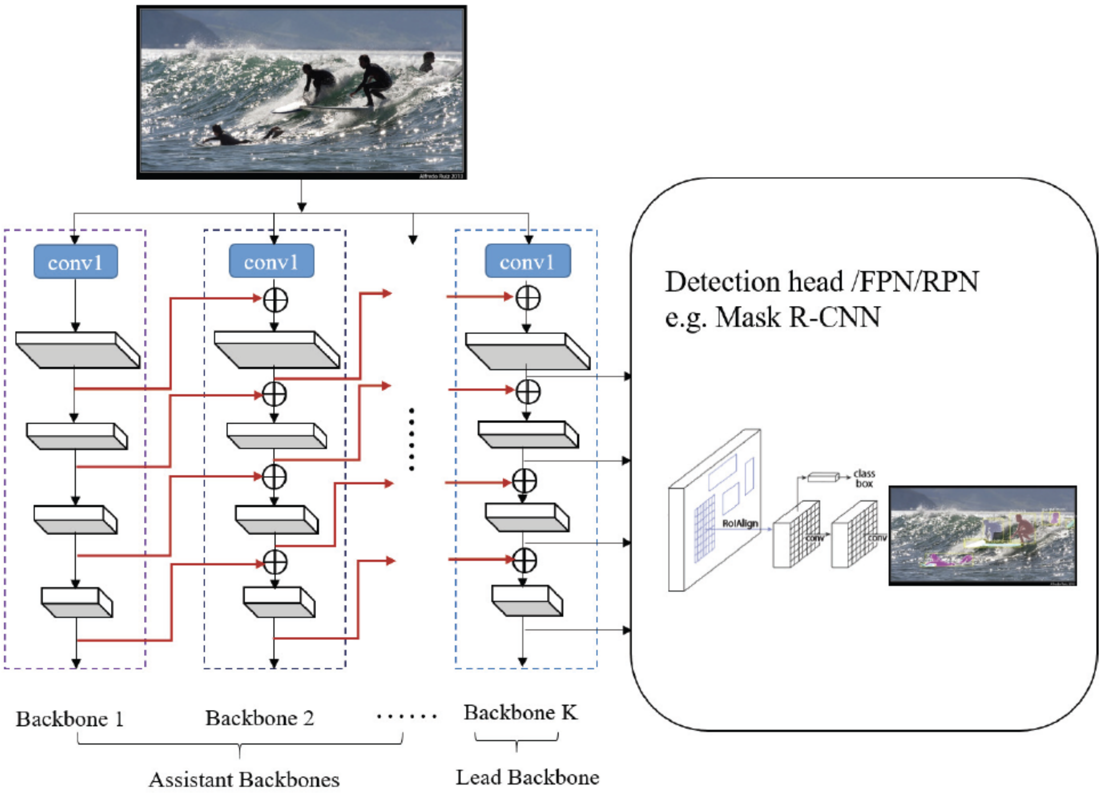

Figure 4: cascading of multiple backbones Liu et al. (2020)

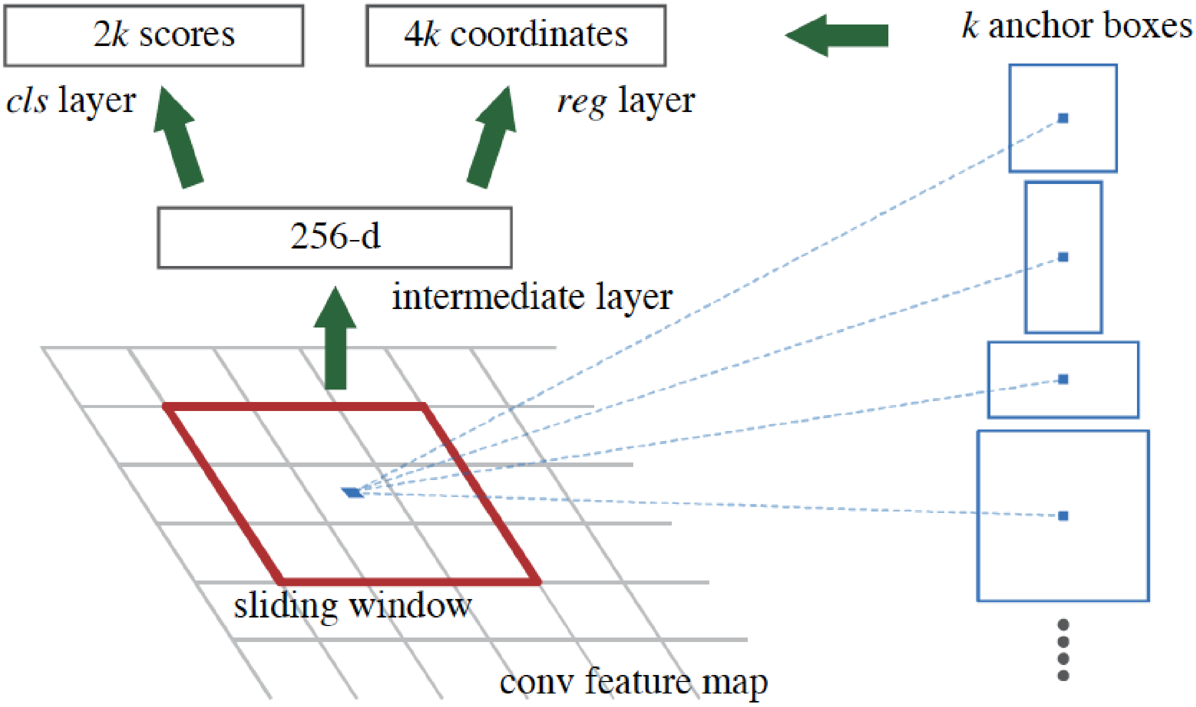

Figure 5: Faster RCNN sliding window approach

References

Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

Karanbir Chahal and Kuntal Dey. A survey of modern object detection literature using deep learning. 08 2018.

Ross Girshick. Fast r-cnn. 04 2015. doi: 10.1109/ICCV.2015.169.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 11 2013. doi: 10.1109/CVPR.2014.81.

Chunhui Gu, Joseph J. Lim, Pablo Arbelaez, and Jitendra Malik. Recognition using regions. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1030–1037, 2009. doi: 10.1109/CVPR.2009.5206727.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 346–361, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10578-9.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. pages 2980–2988, 10 2017. doi: 10.1109/ICCV.2017.322.

Erdem Kemal. Understanding region of interest — (roi align and roi warp). 2020.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018.

Yudong Liu, Yongtao Wang, Siwei Wang, Tingting Liang, Qijie Zhao, Zhi Tang, and Haibin Ling. Cbnet: A novel composite backbone network architecture for object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:11653–11660, 04 2020. doi: 10.1609/aaai.v34i07.6834.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf`.

Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann Lecun. Pedestrian detection with unsupervised multi-stage feature learning. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3626–3633, 2013. ISSN 1063-6919. doi: 10.1109/CVPR.2013.465. 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013 ; Conference date: 23-06-2013 Through 28-06-2013.

Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/file/f7cade80b7cc92b991cf4d2806d6bd78-Paper.pdf`.