Xavier Chun Hin, Chan

1. Template

Intelligence signal processing
Project Idea Title 1: Camera Surveillance System

2. Project Aim

The objective of the project is to design and implement an operational indoor home security camera system that can detect unauthorized entry into a room and promptly alert the user through email notifications. The system is designed to facilitate video and image transmission, along with relevant metadata, to the user without any intermediary involvement. Additionally, the system will feature a user-friendly web dashboard to enable easy access and browsing of historical data.

3. Literature Review

The scope of my research is centered around computer vision-related subjects, encompassing object detection models, computer vision libraries, and tools. Additionally, various metrics and evaluation techniques are pertinent to, and may have been utilized in, each research paper.

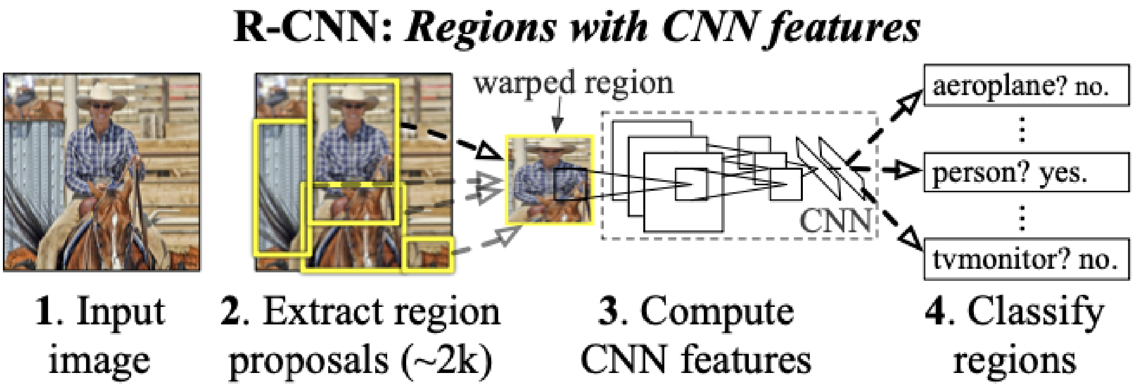
4. Algorithms

R-CNN

The paper[1] authored by Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik proposes a new object detection algorithm that significantly improves mean average precision (mAP)[2]. The authors report a remarkable improvement of more than 30% in mAP compared to the previous best result on the PASCAL VOC 2012 [3], achieving a mAP of 53.3%.

The paper highlights that this is the first study to demonstrate that a convolutional neural network (CNN) can lead to significantly higher object detection performance on the PASCAL VOC compared to systems based on simpler HOG-like features. The researchers in this study aimed to overcome two challenges in object detection.

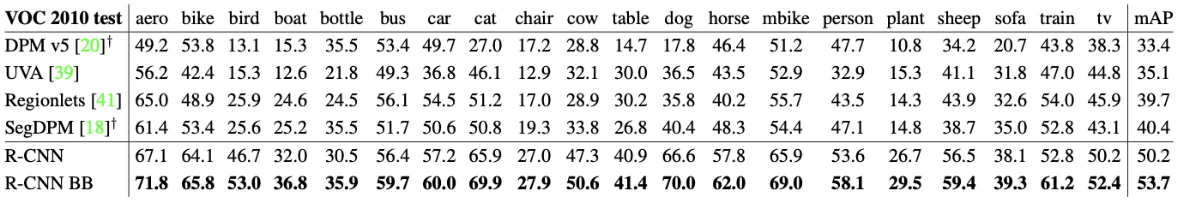
The first challenge was localizing objects using a deep network and training a high-capacity model with limited labeled detection data. They addressed the first challenge by utilizing the "recognition using regions" approach, which has been effective in both object detection and semantic segmentation tasks, to solve the CNN localization problem. During testing, their approach produced approximately 2000 category-agnostic region proposals for each input image, where a fixed-length feature vector was extracted from each submission utilizing a CNN. The method then classified each region using category-specific linear SVMs. A simple technique (affine image warping) was used to generate a fixed-size CNN input for each region proposal, regardless of the region's shape. Since the method merged region proposals with CNNs, it was dubbed R-CNN. (Regions with CNN features).



Object detection system overview. [1]

The authors of the study conducted a comparative analysis between their proposed R-CNN method and the OverFeat [4] detection system, which was recently introduced at the time the paper that was published. This was done by running R-CNN on the 200-class ILSVRC2013 detection dataset, where OverFeat currently held the top position for performance. The study revealed that R-CNN surpassed OverFeat, with a mAP of 31.4% as compared to 24.3%.

The scarcity of labeled data in detection presents a second challenge, where the current amount available is inadequate for training a large CNN. A conventional solution to this problem is to utilize unsupervised pre-training, followed by supervised fine-tuning. In their experiments, fine-tuning for detection enhanced the mAP performance by 8 percentage points. After fine-tuning, their system attained a mAP of 54% on VOC 2010, which is a significant improvement compared to the highly-tuned, HOG-based deformable part model that achieved a mAP of 33%.

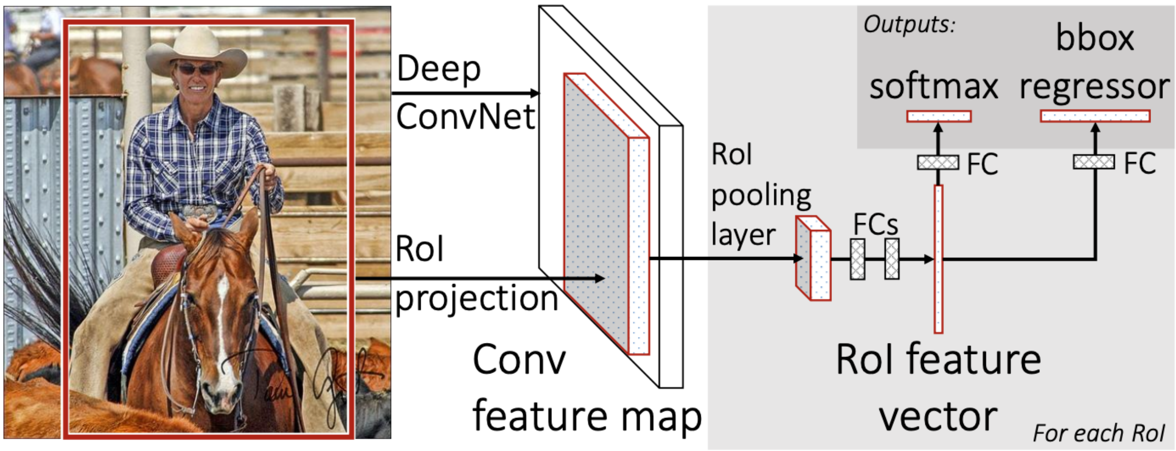


VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [39]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [41]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [18]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

Detection average precision (%) on VOC 2010 test. [1]

Fast R-CNN

The paper was authored by Ross Girshick, a researcher at Microsoft [6]. Fast R-CNN expands upon previous research to effectively classify object proposals via deep convolutional networks. This approach incorporates several innovative techniques to enhance both the training and testing speeds, while also improving detection accuracy when compared to prior methods. Notably, Fast R-CNN achieves a significant improvement in training times, achieving 9 times the speed of R-CNN, and is 213 times faster during test-time, while also obtaining a higher mean average precision (mAP) on PASCAL VOC 2012. Furthermore, compared to SPPnet [7], Fast R-CNN trains the VGG16 [8] network 3 times faster and tests 10 times faster, while also demonstrating superior accuracy.



Fast R-CNN system overview. [5]

The paper presents a training algorithm that concurrently learns to classify object proposals and refine their spatial locations in a single stage. As a result, this approach enables the training of a highly sophisticated detection network (VGG16) at a faster rate than both R-CNN and SPPnet. During runtime, this detection network can efficiently process images within 0.3 seconds while attaining superior accuracy on PASCAL VOC 2012, with a mean average precision (mAP) of 66%, which is 4% higher than the R-CNN method.

Additionally, they highlighted notable drawbacks of the R-CNN method, despite its capacity to attain high levels of accuracy, including:

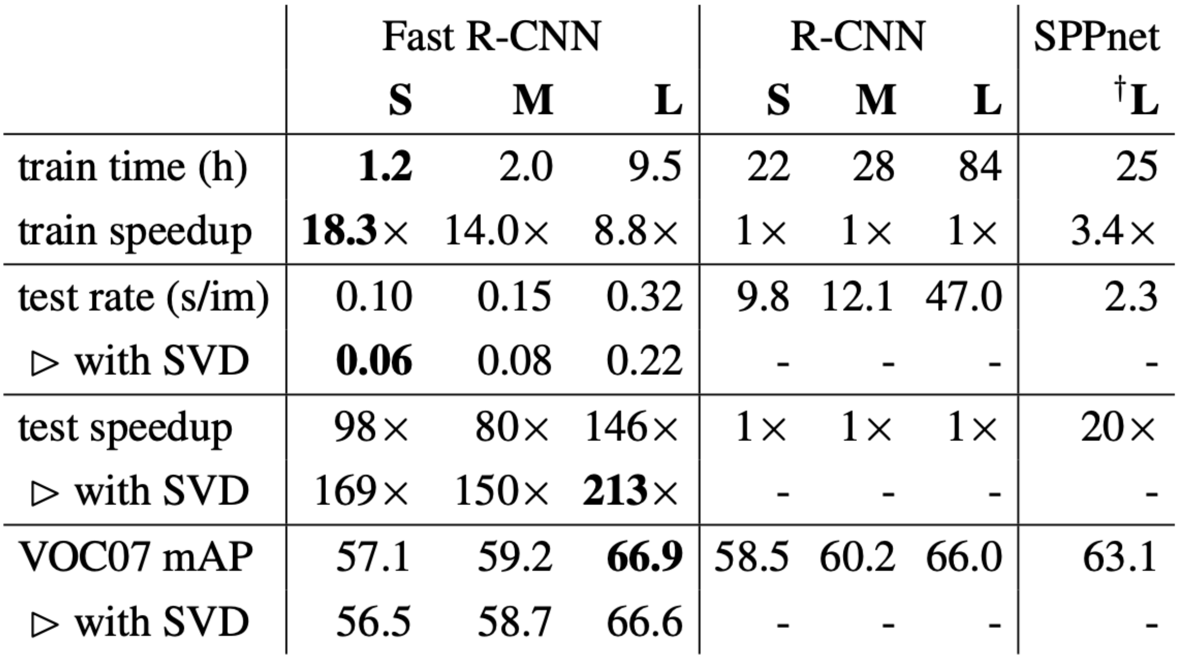
1. The training process is a multi-stage pipeline.
2. The training process demands significant space and time resources.
3. Object detection using this approach is comparatively slow.

Furthermore, the authors address the factors that hindered the speed of the R-CNN method, as well as their solutions. In particular, the R-CNN method was found to be slow due to its inability to share computation, which resulted in the performance of a ConvNet [8] forward pass for each object proposal. To address this challenge, SPPnets were introduced as a means to expedite R-CNN by sharing computation. This approach led to a remarkable 10 to 100 times acceleration of R-CNN during test-time, while also reducing training time by a factor of 3 through faster proposal feature extraction. Despite these gains, SPPnets have their own limitations. For instance, unlike R-CNN, the proposed fine-tuning algorithm cannot update the convolutional layers preceding the spatial pyramid pooling, which restricts the accuracy of very deep networks.

The new algorithm overcomes the limitations of the R-CNN and SPPnet approaches and enhances their speed and accuracy. The method is referred to as Fast R-CNN, owing to its relatively faster training and testing times. Additionally, Fast R-CNN has several advantages over the prior methods, including:

1. Superior detection accuracy (measured in mAP) compared to R-CNN and SPPnet.
2. The training is a single-stage process that utilizes a multi-task loss.
3. The training process can update all network layers.
4. There is no need for disk storage for feature caching.

The study's findings on Fast R-CNN's testing outcomes are as follows. Notably, the second significant outcome is the improved speed in both training and testing times. The figure below illustrates the comparison between training time (measured in hours), testing rate (measured in seconds per image), and mAP on VOC07 for Fast R-CNN, R-CNN, and SPPnet. Specifically, Fast R-CNN can process images using VGG16 146 times faster than R-CNN without truncated SVD and 213 times faster with it. Training time is reduced by a factor of 9, from 84 hours to 9.5. Compared to SPPnet, Fast R-CNN trains VGG16 2.7 times faster (in 9.5 vs. 25.5 hours) and tests 7 times faster without truncated SVD, or 10 times faster with it. Fast R-CNN eliminates the need for storing features, which would otherwise require hundreds of gigabytes of disk storage.

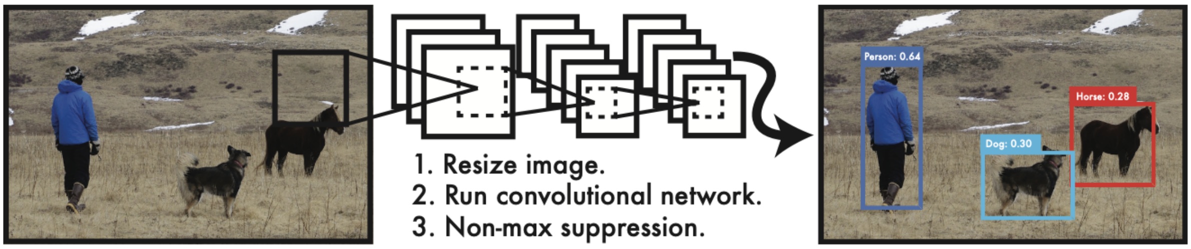


	Fast R-CNN			R-CNN			SPPnet
	S	M	L	S	M	L	†L
train time (h)	**1.2**	2.0	9.5	22	28	84	25
train speedup	**18.3×**	14.0×	8.8×	1×	1×	1×	3.4×
test rate (s/im)	0.10	0.15	0.32	9.8	12.1	47.0	2.3
▷ with SVD	**0.06**	0.08	0.22	-	-	-	-
test speedup	98×	80×	146×	1×	1×	1×	20×
▷ with SVD	169×	150×	**213×**	-	-	-	-
VOC07 mAP	57.1	59.2	**66.9**	58.5	60.2	66.0	63.1
▷ with SVD	56.5	58.7	66.6	-	-	-	-

The table shows a runtime comparison between the identical models implemented in Fast R-CNN, R-CNN, and SPPnet. Fast R-CNN employed a single-scale mode, whereas SPPnet utilized the five scales specified in the study. The measurements were taken on an Nvidia K40 GPU.

YOLO

The authors of this publication, Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi [9], introduce a novel methodology called YOLO for the purpose of object detection. Traditionally, object detection has relied on adapting classifiers for detection tasks. However, the authors propose a different approach by formulating object detection as a regression problem, specifically addressing the spatial separation of bounding boxes and their associated class probabilities. Through a singular neural network, YOLO enables the direct prediction of bounding boxes and class probabilities from complete images in a single evaluation. This unified network architecture allows for end-to-end optimization, directly enhancing the overall detection performance.



YOLO system overview. [9]

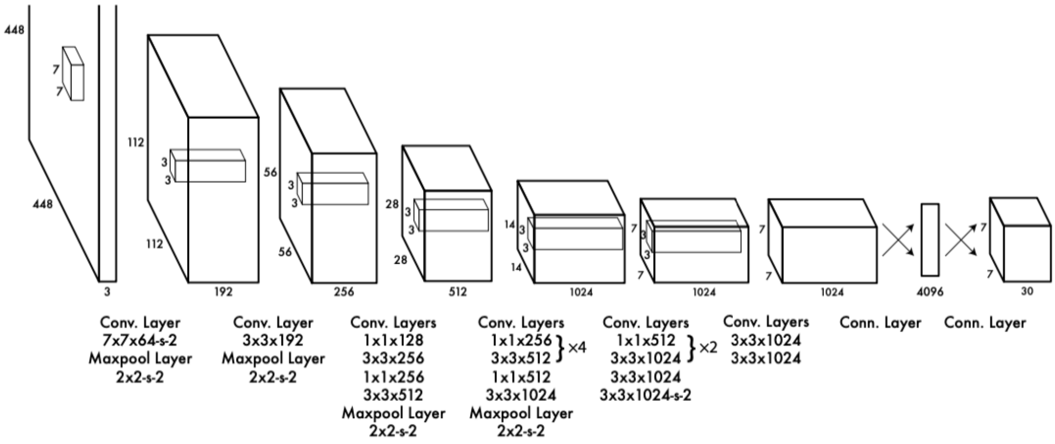
The authors acknowledge the limitations of deformable parts models (DPM), which rely on a sliding window approach, and the region proposal methods employed by R-CNN. These existing approaches are burdened by complex pipelines, which in turn contribute to their slow execution and challenging optimization process. A key factor contributing to this inefficiency is the requirement to train each individual component separately.

The authors address the limitations associated with deformable parts models (DPM), which rely on a sliding window approach, and the region proposal methods utilized by R-CNN. These methods exhibit complex pipelines, resulting in slower performance and difficulties in optimization due to the need to train individual components separately.

To overcome these challenges, the authors propose a unified solution, employing a single convolutional network that performs simultaneous predictions of multiple bounding boxes and class probabilities. This network is trained on full images and directly optimized for enhanced detection performance. The adoption of this unified model brings several advantages compared to traditional object detection methods. Furthermore, the authors highlight that YOLO achieved more than double the mean average precision (mAP) compared to other real-time systems [10].

The network architecture draws inspiration from the GoogLeNet model utilized for image classification [11]. The authors' network consists of 24 convolutional layers, followed by two fully connected layers. In contrast to GoogLeNet's employment of inception modules, the authors employ 1 × 1 reduction layers followed by 3 × 3 convolutional layers, similar to the approach by Lin et al.

Additionally, the authors train a fast variant of YOLO, specifically designed to achieve rapid object detection. Fast YOLO employs a neural network with fewer convolutional layers (9 instead of 24) and reduced filters within those layers. Apart from the network size, all training and testing parameters remain consistent between YOLO and Fast YOLO.



YOLO architecture overview. [9]

YOLO demonstrates its ability to acquire generalizable object representations, surpassing prominent detection methods like DPM and R-CNN by a substantial margin when tested on artworks after being trained on natural images. This high level of generalizability reduces the likelihood of failure when applied to novel domains or unexpected inputs.

However, the authors acknowledge that, at the time of writing, YOLO still falls behind state-of-the-art detection systems in terms of accuracy. While the model exhibits swift object identification capabilities, it encounters difficulties in precisely localizing certain objects, particularly smaller ones. Despite imposing robust spatial constraints on bounding box predictions, where each grid cell predicts only two boxes with a single class, there are inherent limitations. This spatial constraint restricts the model's ability to predict a larger number of nearby objects, leading to challenges when dealing with small objects appearing in groups, such as flocks of birds.

Furthermore, due to the model's reliance on learning to predict bounding boxes from data, it struggles to generalize well to objects with atypical aspect ratios or configurations. Additionally, the use of relatively coarse features for bounding box prediction arises from the architecture's incorporation of multiple downsampling layers from the input image.

Lastly, although the authors train the model using a loss function that approximates detection performance, the treatment of errors remains uniform for both small and large bounding boxes. While a small error in a larger box generally has a negligible impact, a small error in a smaller box significantly affects the intersection over union (IOU). The primary source of error lies in inaccurate localizations.

In addition, the authors conducted a comparative analysis between the YOLO detection system and several leading detection frameworks. The findings of this evaluation are summarized as follows:

1. DPM [12] - In contrast to the disjoint pipeline utilized by the deformable parts model (DPM), which involves extracting static features, classifying regions, and predicting bounding boxes separately, the authors propose a single convolutional neural network architecture. This unified approach replaces the individual components with a cohesive network that trains features in tandem, optimizing them specifically for the detection task. Consequently, the unified architecture of YOLO achieves superior speed and accuracy compared to DPM.
2. R-CNN [1] - Unlike the region proposal approach employed by R-CNN and its variants, which aim to identify objects in images, Redmon et al. introduced selective grasp detection. This method incorporates a grid-based bounding box prediction technique inspired by the MultiGrasp system [13], which focuses on regression for grasp detection. However, it is important to note that grasp detection represents a comparatively simpler task than object detection. MultiGrasp solely needs to predict a single graspable region in an image containing a single object, without requiring the estimation of size, location, boundaries, or class prediction. In contrast, YOLO excels by simultaneously predicting bounding boxes and class probabilities for multiple objects across various classes within an image.

5. Libraries/tools

Open CV

OpenCV [14] (Open Source Computer Vision) is a free and open-source computer vision and machine learning software library. It was originally developed by Intel in 1999 and later maintained by Willow Garage and now by Itseez. OpenCV provides various tools and libraries that can be used to develop real-time computer vision applications, including image and video processing, object detection, face recognition, feature detection and extraction, and more. It supports multiple programming languages, including C++, Python, Java, and MATLAB, making it a popular choice for developers and researchers in the field of computer vision.

Reference

[1] Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
https://arxiv.org/pdf/1311.2524.pdf
https://inst.eecs.berkeley.edu/~cs280/sp15/lectures/10.pdf

[2] Breaking Down Mean Average Precision (mAP), Ren Jie Tan, 25th Mar 2019, https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52

[3] Visual Object Classes Challenge 2012 (VOC2012), Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A., 2008-2012, http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

[4] OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks, Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, Yann LeCun, 24th Feb 2014, https://arxiv.org/pdf/1312.6229.pdf

[5] Fast R-CNN, Ross Girshick, 27 Sep 2015, https://arxiv.org/pdf/1504.08083.pdf

[6] Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, 23rd Apr 2015, https://arxiv.org/pdf/1406.4729.pdf

[7] VGG-16 | CNN model, pawangfg, 10th Jan 2023
https://www.geeksforgeeks.org/vgg-16-cnn-model/

[8] Convolutional neural network, Wikipedia, https://en.wikipedia.org/wiki/Convolutional_neural_network

[9] You Only Look Once: Unified, Real-Time Object Detection, Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, 9th May 2016, https://arxiv.org/pdf/1506.02640.pdf

[10] YOLO real-time on a webcam demo, YOLO, https://pjreddie.com/darknet/yolo/

[11] Deep Learning: GoogLeNet Explained, Richmond Alake, 23rd Dec 2020, https://towardsdatascience.com/deep-learning-googlenet-explained-de8861c82765

[12] Deformable Part Models are Convolutional Neural Networks, Ross Girshick, Forrest Iandola, Trevor Darrell, Jitendra Malik, 1st Oct 2014, https://arxiv.org/pdf/1409.5403.pdf

[13] Real-world Multi-object, Multi-grasp Detection, Fu-Jen Chu, Ruinian Xu, Patricio A. Vela, 20th Jul 2018, https://arxiv.org/pdf/1802.00520.pdf

[14] OpenCV, https://opencv.org/