Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

R&D Project

Localization of Objects Using Unsupervised Representation Learning and Object Proposal Techniques

Venkata Santosh Sai Ramireddy Muthireddy

Submitted to Hochschule Bonn-Rhein-Sieg,
Department of Computer Science
in partial fullfilment of the requirements for the degree
of Master of Science in Autonomous Systems

Supervised by

Prof. Dr. Paul G. Plöger
M.Sc. Deebul Nair

August 2020

I, the undersigned below, declare that this work has not previously been submitted to this or any other university and that it is, unless otherwise stated, entirely my own work.

Date	Venkata Santosh Sai Ramireddy Muthireddy

Abstract

This research and development project addresses the object localization problem i.e, predicting the locations of objects that belong to a particular category in a given scene. Current object localization techniques based on deep learning models require object-level annotations to train the model, but preparing annotated datasets is time-consuming and financially expensive.

In this work, a thorough investigation is carried out to address the problem of object localization in an unsupervised setting by using off-the-shelf generic object proposal techniques and unsupervised representation learning techniques. However, on further research, it is understood that this problem can be solved using a weakly supervised setting rather than unsupervised techniques.

In Weakly Supervised Object Localization (WSOL), object locations are estimated using the classification models which are trained only on image-level labels. A systematic literature review is done on available WSOL methods. Multiple approaches are investigated and the best approach to solve this problem is identified. The identification of best approaches is based on the frequency of benchmarking, datasets used to prove the technique and methods against which it is proved.

Extensive experimentation is carried out to evaluate the WSOL methods Class activation mapping (CAM) and Gradient class activation mapping (Grad-CAM) to illustrate the classification and localization performance of the chosen approaches. This experimentation is carried out on three datasets namely PASCAL Visual Object Classes (PASCAL VOC), Yale-CMU-Berkeley (YCB) dataset for robotic manipulation research, and RoboCup@Work datasets.

On experimentation, it is observed that the size of the activation map plays an important role in localization and classification performance. If the size of the activation map is increased or decreased the classification and localization performance is poor compared to the stock activation map size from VGG16. Performance summary of experiment with VGG16 as network (16x16 pixels activations) and using PASCAL VOC dataset is, Maximal Box Accuracy (MaxBoxAcc): 58%, Pixel Average Precision (PxAP): 46%, Accuracy: 72%, Precision: 82% and Recall: 78%. Whereas for reduced map size (8x8 pixels) performance summary is MaxBoxAcc: 46%, PxAP: 38%, Accuracy: 68%, Precision: 77% and Recall: 75% and increased map size (16x16 pixels) performance is MaxBoxAcc: 39%, PxAP: 27%, Accuracy: 58%, Precision: 68% and Recall: 66%. We hypothesize that the poor performance when activation map size is reduced is due to loss of spatial feature information. Whereas performance drop in the case of increased activation map size is due to the concentration of significant features at the center of the activation map.

Acknowledgements

Foremost, I would like to convey my wholehearted gratitude to my supervisors Prof. Dr. Paul G. Plöger and M. Sc. Deebul Nair for the continuous support of my Research and Development project, for there patience, enthusiasm, immense knowledge and motivation. Their guidance helped me in all the time of this project.

I thank my colleagues at Hochschule Bonn-Rhein-Sieg: Anirudh, Deepan, Devaiah, Jaswanth, Lokesh, Mihir, Ragith, Sasi Kiran and Swaroop for the continuous support and discussion. I am blessed to have the moral support provided by my parents and friends, which motivated me to push the limits.

Contents

List of Figures

List of Tables

1

Introduction

Object localization is predicting the location of the object along with its boundaries in the given image. Object localization is very crucial in understanding the image and the first step for object detection tasks. The basic challenge in object localization is preparing models to handle complex and cluttered scenarios. The localization of multiple objects along with their category is object detection. Many deep learning techniques like Single Shot Detection (SSD) [1] and You Only Look Once (YOLO) [2] require a large number of training images in a strongly supervised setting. The problem is obtaining the large dataset with strong supervision annotations in other words fine-grained instance-level annotation like bounding boxes. There is a need for methods that use coarse-grained labels like image-level labels (only class labels for each image) for object localization tasks. This chapter describes the motivation behind the work followed by challenges and difficulties and the problem statement.

1.1 Motivation

Deep Neural Networks have been proven to solve many computer vision applications like semantic segmentation [3], object detection [4], and image captioning [5]. Deep learning models are generally trained on large image datasets such as ImageNet [6], MS-COCO [7], PASCAL VOC [8]. These datasets are used to pre-train deep learning models and reduce the risk of overfitting on small-sized dataset during transfer learning. The performance of deep networks is also dependent on the amount of training data available. However, making an annotated dataset with a large number of images is financially expensive and time-consuming. The availability of annotated data is not the only limitation for supervised learning. Additionally, models trained on a particular dataset cannot classify or detect objects that have not been trained on the same dataset. Machine Learning enthusiasts have the common assumption that Artificial Intelligence (AI) is exclusively coding but in reality, this contributes to only 20 percent of the AI system which is illustrated in Figure 1.1. Most of the effort to build a deep learning or a machine learning model is devoted to collecting and annotating the data to train the model. Creating an annotated

dataset is time-consuming, for example precisely annotating one single image from Cityscapes dataset [9] requires no less than 1.5 hours and assuming 5000 images were annotated, so that results in an average of 7500 man-hours [10]. If it is taken as 10 dollars as base labor price, it costs approximately 75000 dollars to annotate just 5000 images for semantic segmentation. In the case of object-level annotation, the cost is approximately 10000 dollars for 5000 images. In Figure 1.2, time required per each type of annotation is given. As we observe the time required per instance increases as the annotation type progresses from coarse-grained image-level labels to fine-grained pixel-level labels. The above explanation can clearly show the impact of annotation type on time and cost factors.

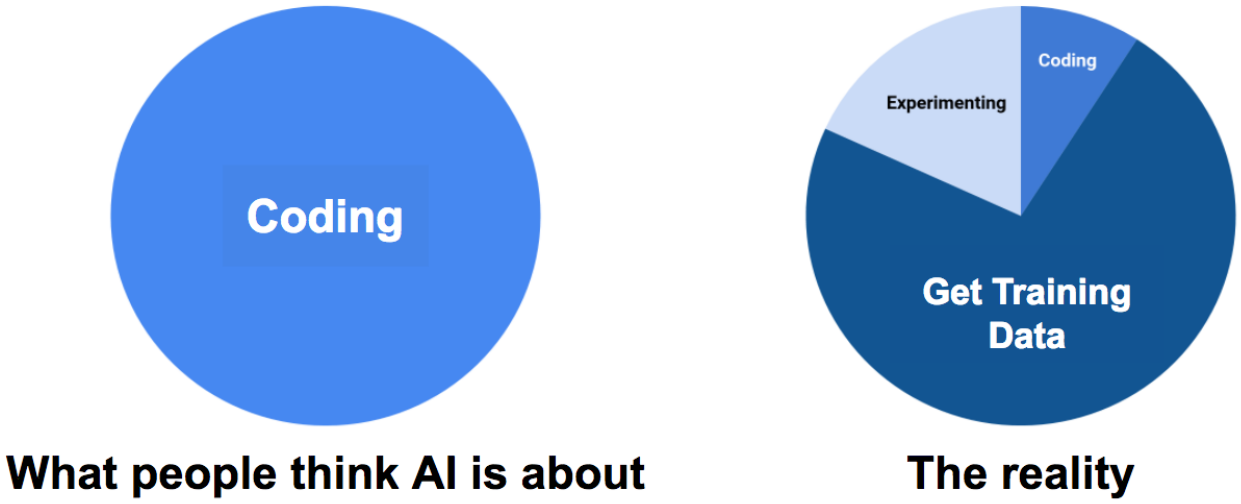


Figure 1.1: Reality of data requirement [10]

In order to tackle the problems that arise due to the unavailability of annotated datasets, unsupervised, semi-supervised, self-supervised, and weakly supervised learning (discussed in Chapter 2) approaches can be implemented which can use a large amount of unlabeled data. Unsupervised learning is widely practiced in machine learning techniques like clustering, density estimation, and dimensionality reduction. Deep neural networks trained in a supervised setting are not capable of detecting objects not in the trained classes. However, there is a probability that objects which do not belong to trained classes are present in the scene. Training a deep neural network with a new dataset that has unseen category objects is not a convenient option, rather it is helpful to train the network with a new dataset without any fine-grained annotations like bounding boxes. This project includes the implementation of a pipeline that can be trained with new data without any object-level annotations. This model can be later used to localize the new objects in the given scene. If the suggested approach is implemented, it will help to

solve the issues faced by the RoboCup teams at H-BRS (b-it-bots team) during the competitions where new objects are introduced every year.

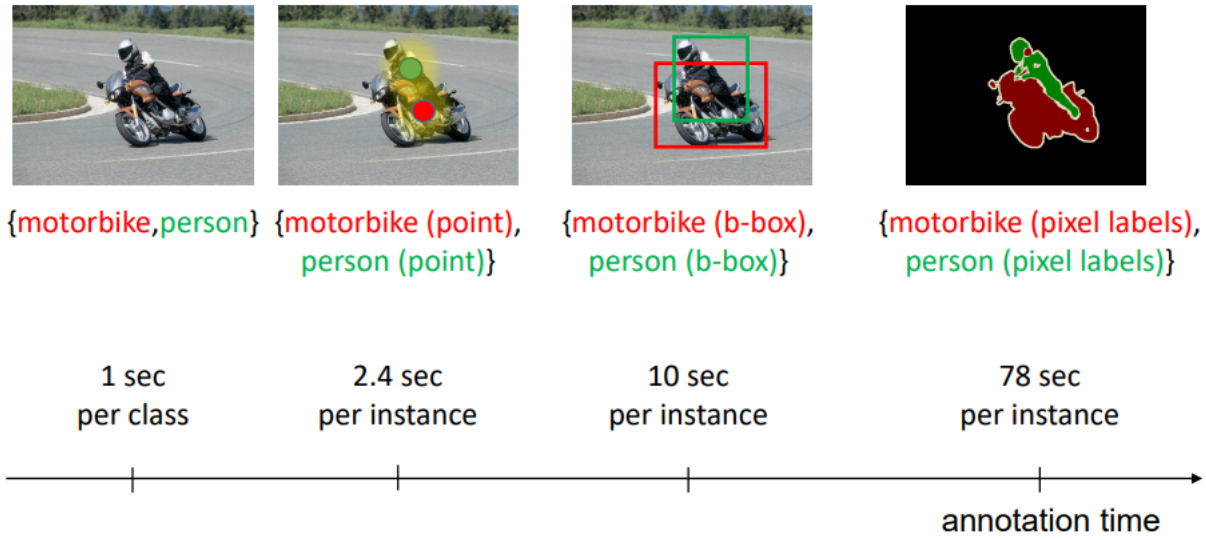


Figure 1.2: Time taken for each type of annotation [11]

To summarize the problem, the dataset will consist of curated data that is image-level labels are available for all the images in the dataset. Using this dataset, the aim to identify the location of objects is to be achieved. Based on the requirements and exhaustive literature search, it is understood that Weakly Supervised Object Localization(WSOL) [12, 13] is the approach to exactly fit the experiment setting and dataset constraints.

1.2 Challenges and Difficulties

The challenges and difficulties in the project progress are described below:

- Multi-label classification with SqueezeNet [14]: Training the SqueezeNet [14] (fully convolutional architecture with no dense layers) for multi-label classification (described in Section 2.2 is a difficulty as there is no prior work done to classify multi-label problems with SqueezeNet architecture.

- Adapting WSOL Techniques: Adapting Class Activation Mapping (CAM) [13] and Gradient Class Activation Mapping (Grad-CAM) [15] WSOL methods for all the deep learning architectures which include VGG16 [16], ResNet18 [17], and SqueezeNet [14]. This is a challenge as three selected architectures are very different with the main roadblock being SqueezeNet as it fully convolutional network with no fully connected dense layers.

- Yale-CMU-Berkeley (YCB) dataset handling: Handling the YCB dataset [18] was pretty tough as the dataset contains high-resolution images and only segmentation masks were given in annotations. These masks are in a new format which became a challenge to read and extract tight fit bounding boxes from masks.

1.3 Problem Statement

This R&D project will provide a literature study on State-Of-The-Art (SOTA) unsupervised deep learning methods, generic object proposal algorithms, and WSOL methods. One of the main focuses is to provide a software pipeline, which is capable of localizing the objects that belong to a particular category in a given scene. The main constraint is training the deep neural network with unlabeled data (no object-level annotations). Which is training the deep neural network model as a classification model which is described in Figure 1.3. This trained classification should be used to obtain the locations of objects which is shown in Figure 1.4. The furthermore complex situation would be using the data which is not curated (similar class images are not grouped). The identified method will be able to localize the objects in a scene and should be bound to the following metrics: there should be no false negatives, lower latency (real-time localization), and false positives are allowed. In the experimentation phase, statistics for specified metrics will be collected for the identified models. The best model to solve the problem will then be proposed. The research questions (RQ) that are answered in this R&D project are as follows:

RQ1 What are available methods to find locations of objects by using only image-level labels?

RQ2 Which metric/s to be chosen to satisfy constrains mentioned in proposal?

RQ3 What is the performance of selected WSOL method/s on smaller backbones like ResNet18 and SqueezeNet1_1?

RQ4 How selected methods are performing based on defined metrics on three datasets namely, PASCAL VOC, YCB and RoboCup@Work?

RQ5 What is the impact of activation map size at last convolution layer on the performance of WSOL methods?

RQ6 How to select the optimal threshold for generating binary mask from class activation map?

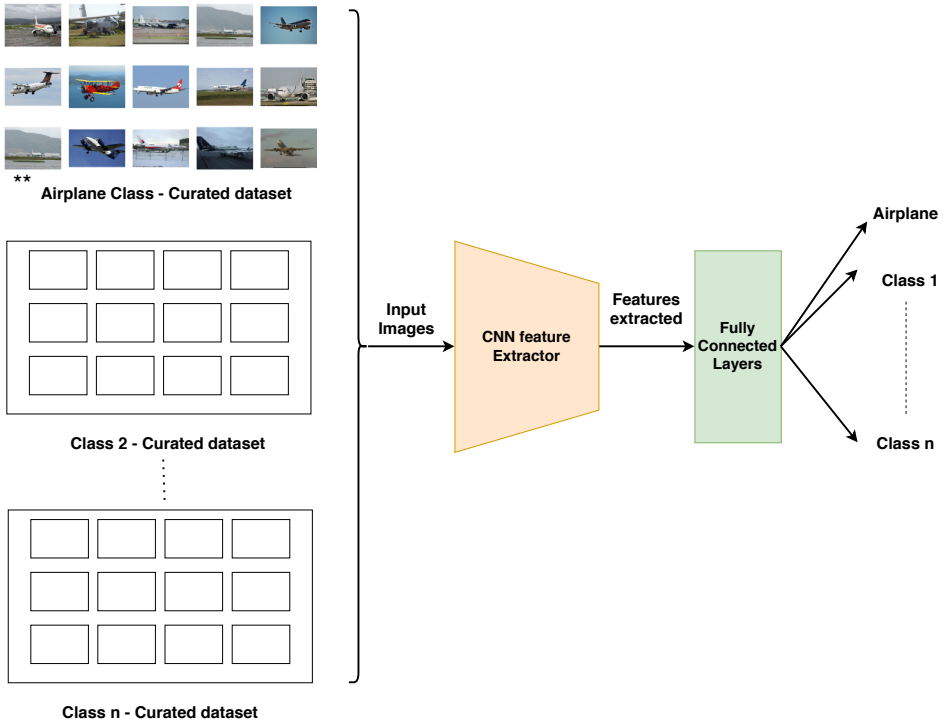


Figure 1.3: Training a deep neural network as classification model to predict the class for a given input image.

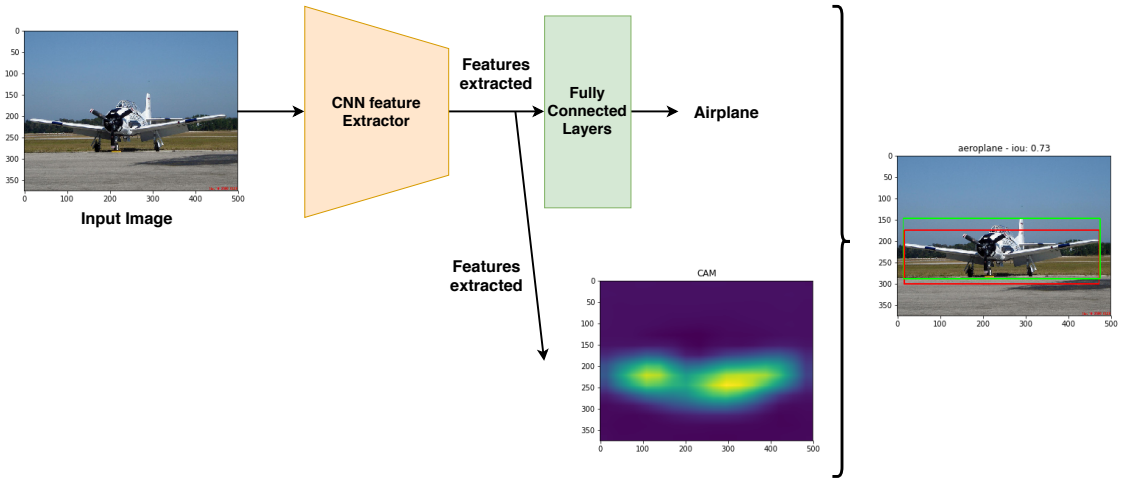


Figure 1.4: Using classification model for localization of objects where activation map is used to localize.

1.4 Report Outline

In Chapter 2, an overview of types of learning, types of classification, and applications are discussed. Types of learning include supervised, unsupervised, semi-supervised, self-supervised, and weakly supervised learning which intends to give information for absolute beginners. Chapter 3 provides search criteria and literature for general object proposal techniques, unsupervised representation techniques, and weakly supervised object localization techniques. This literature contains SOTA work presented in the respective areas and statistics for generic object proposal techniques. In the following Chapter 4, requirements for the dataset are mentioned along with discussing the available datasets that suit the requirements. In Section 4.4, detailed description of selected datasets is presented and in Section 4.5, dataset analysis is carried out to provide more insights on selected datasets. In Chapter 5, a brief discussion on method selection criteria is presented. Along with the method selection criteria, two selected methods are discussed in detail. Besides, the outcome of this research is presented in Chapter 6 for the methods selected based on the selection criteria mentioned in Chapter 5. Finally, contributions, lessons learned, and future research direction are portrayed in Chapter 7.

2

Background

In this chapter, a brief introduction to types of learning is provided to give an overview of the difference between types of learning which is helpful in the coming chapters. Alongside, types of classification are discussed for better understanding of multi-label classification.

2.1 Types of Learning

In this section, an overview of types of learning is presented. Type of learning includes supervised learning, unsupervised learning, semi-supervised learning, self-supervised learning, and weakly supervised learning.

2.1.1 Supervised Learning

In supervised learning, the model will have prior knowledge of how output values look for given input values. The objective in supervised learning is finding a function that maps a relation between input values and desired output values, given ground truth or target values during training. The workflow of supervised learning is explained in Figure 2.1. In training, data to the supervised learning model is always given in multiple pairs of input and respective ground truth labels. The mathematical interpretation of supervised learning is given. Input to the model is given by S=[(x_1, y_1), (x_2, y_2),....., (x_n, y_n)] where y_i is the label corresponding to x_i. The expected output from the supervised learning technique is a learnt function which maps from input to output. So learnt function is given by $h : x_i \rightarrow y_i$, where h is the learnt function.

Applications

Types of problems where supervised learning can be applied are as follows:

1. Classification: In classification, entities are sorted into respective categories. Sample applications of classification are:

- Image classification
- Diagnostics

2. Regression: In regression, real values are predicted for given input rather than discrete categories as given in classification. Few applications of regression are:

   - Population growth prediction
   - Weather forecast

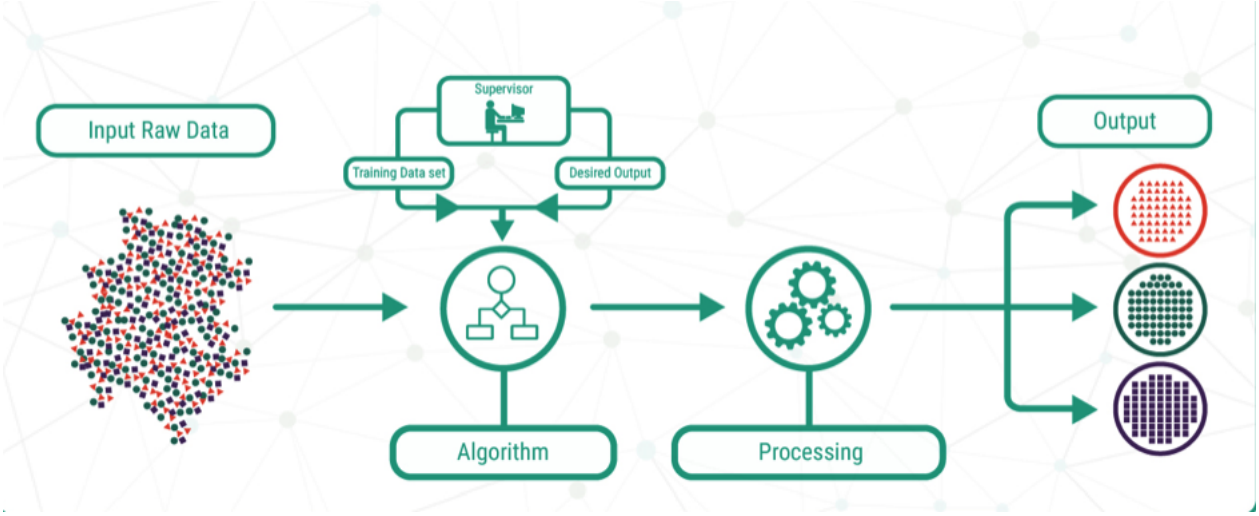


Figure 2.1: Block diagram explaining workflow of supervised learning [19]. A supervisor control is available during training that try to reduce difference between predicted and ground truth value.

2.1.2 Unsupervised Learning

In unsupervised learning, the model will not have prior knowledge of how output values look for the given input values. The objective of unsupervised learning is finding an underlying structure for the given input data. The workflow of supervised learning is explained in Figure 2.2. In training data only set of input values are provided without any ground truth. In this learning technique mapping function from input to output is found by extracting useful features and analyzing them.

Mathematical interpretation of unsupervised learning is given. Let the input to the model is given by S=[(x_1), (x_2),....., (x_n)] where x_i is the training data with m number of features. The expected output from the unsupervised learning technique is a learned function which maps from input to output. So learned function is given by $h : x_i \rightarrow y_i$, where h is the learned function and

y_i is the cluster or group or number of significant components in given input data. Training data is partially labeled i.e., a small amount is labeled and a large amount of data is unlabeled.

Applications

Types of problems where unsupervised learning can be used to solve the problem are as follows:

1. Clustering: In clustering, inputs are organized into different groups or clusters by finding the underlying structure of the data. Some applications of clustering are:

   - Customer Segmentation
   - Recommended Systems

2. Dimensionality reduction: In dimensionality reduction, the aim is to decrease the number of features or columns in given input data to get the most significant data and remove the redundant noisy data. Applications of dimensionality reduction are:

   - Big data visualization
   - Meaningful compression

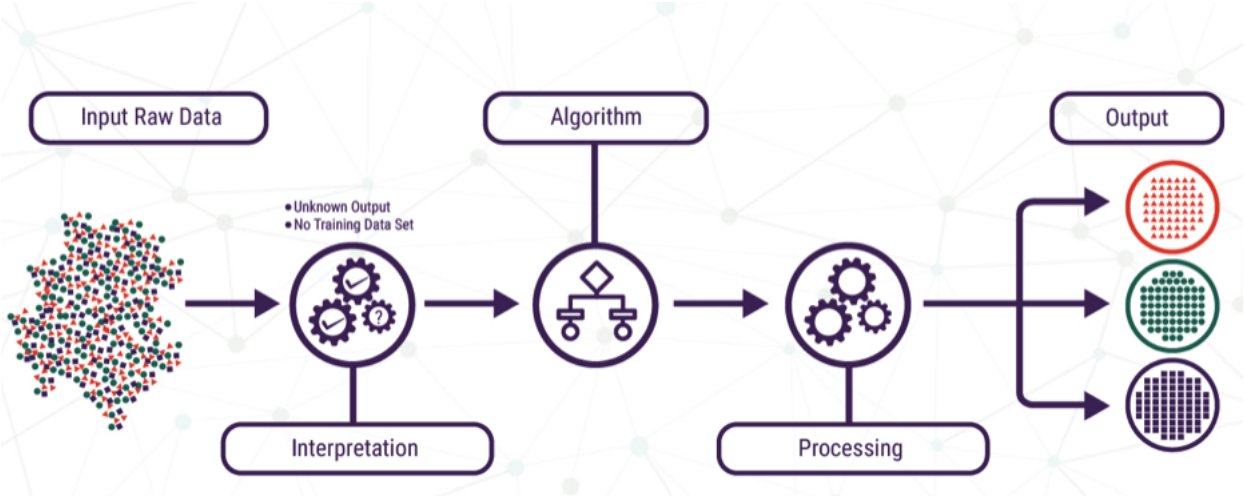


Figure 2.2: Block diagram explaining workflow of unsupervised learning [20]. In this, no supervisory control is available which indicates no ground truth labels are available during training.

2.1.3 Semi Supervised Learning

In semi-supervised learning, data with annotations and without annotations is used as training data during the model training stage. Train model with labeled data and predict labels for unlabeled data with this model and these data are used to extend the dataset. The workflow of semi-supervised learning is illustrated in Figure 2.3. Mathematical interpretation [21] of semi-supervised learning is given as below:

- Consider two datasets D_l and D_u, where D_l is labeled data set $D_l = (X_l, Y_l)$ where $X_l = (x_1, x_2, ...x_n)$ and $Y_l = (y_1, y_2, ...y_n)$. and $D_u = (X_u)$ is unlabeled dataset where $X_u = (x_{1+n}.....x_{m+n})$ and $m >> n$
- Find $h : X_l \rightarrow Y_l$, where h is learnt model.
- Find Y_u using the mapping function learnt from labelled dataset $h(X_u) = Y_u$. Now use $D_u = (X_u, Y_u)$ as labeled training dataset to generalize $h : X \rightarrow Y$

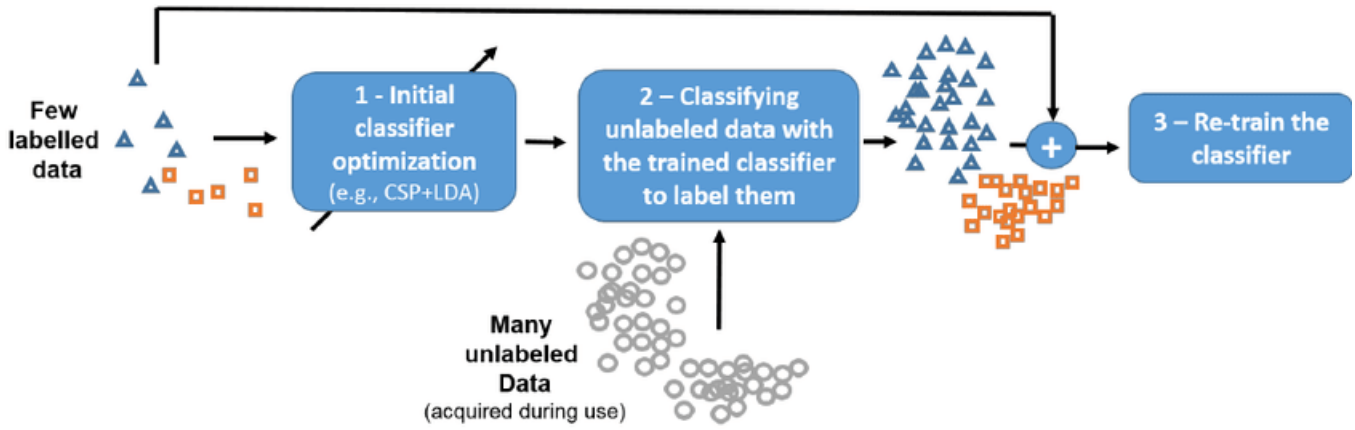


Figure 2.3: Principle of semi-supervised learning [22]. As shown in figure ground truth labels are available for few samples and large chunk of unlabeled data is available during training.

Applications

1. Text Classification
2. Document Clustering
3. Speech Analysis

2.1.4 Self Supervised Learning

Self-supervised learning is part of unsupervised learning. In self-supervised learning, during training phase labels are obtained for free i.e., labels for input data are automatically generated. In other words, this technique is known as pseudo labeling. In self-supervised learning algorithms are learned in a fully supervised manner but no manual labeling is required. It consists of two types of tasks, they are pretext task which is used in pretraining the network and downstream task which is used for fine-tuning the model. The general pipeline for self-supervised learning is shown in Figure 2.4.

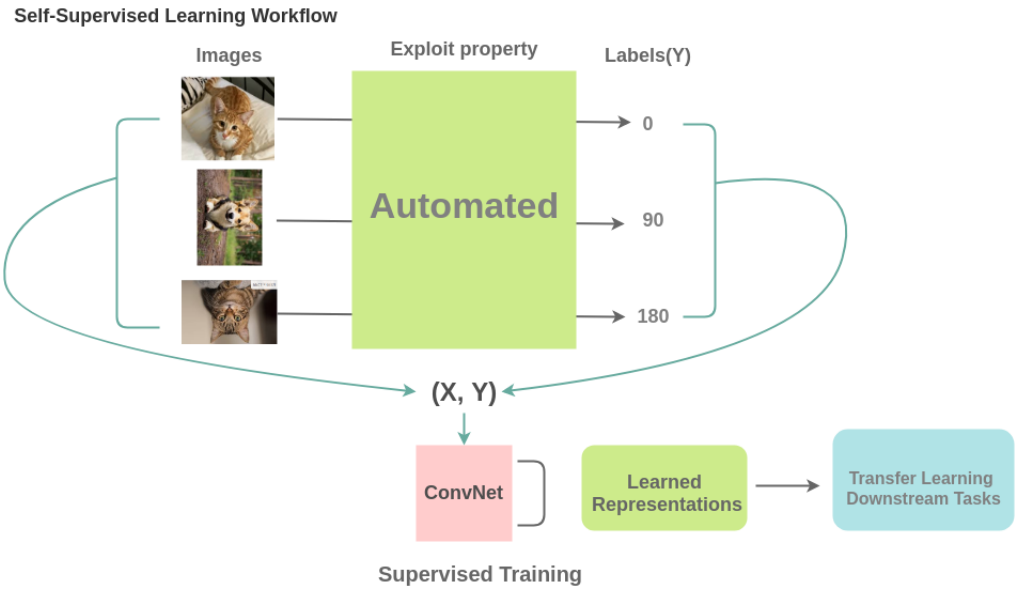


Figure 2.4: General pipeline for self-supervised learning [23]. In this data is labeled automatically by exploiting significant property of data.

Mathematical interpretation of self-supervised learning is given as below:

- Let $X_i = (x_1, x_2, ...x_n)$ is the input.
- Function $g : X_i \rightarrow Y_i$ generates Y_i where $Y_i = (y_1, y_2, ...y_n)$ are pseudo labels.
- Input fed to training model is (X_i, Y_i)
- Mapping function $f : X_j \rightarrow Y_j$ s learnt using the knowledge transferred from above.

Applications

Few applications of self-supervised learning in context of computer vision tasks are:

1. Image Colorization (grayscale, colorized)
2. Image Superresolution (small, upscaled)
3. Context Prediction (image-patch, neighbor)

2.1.5 Weakly Supervised Learning

In weakly supervised learning, lower degree annotation is available at the training time i.e., coarse-grained annotations like image-level, curated datasets, and a higher level of annotation are expected during evaluation or testing phase i.e., fine-grained annotations like bounding boxes, semantic labels. An example pipeline for weakly supervised learning is shown in Figure 2.5.

Applications

1. Object Localization
2. Object Detection
3. Semantic Segmentation

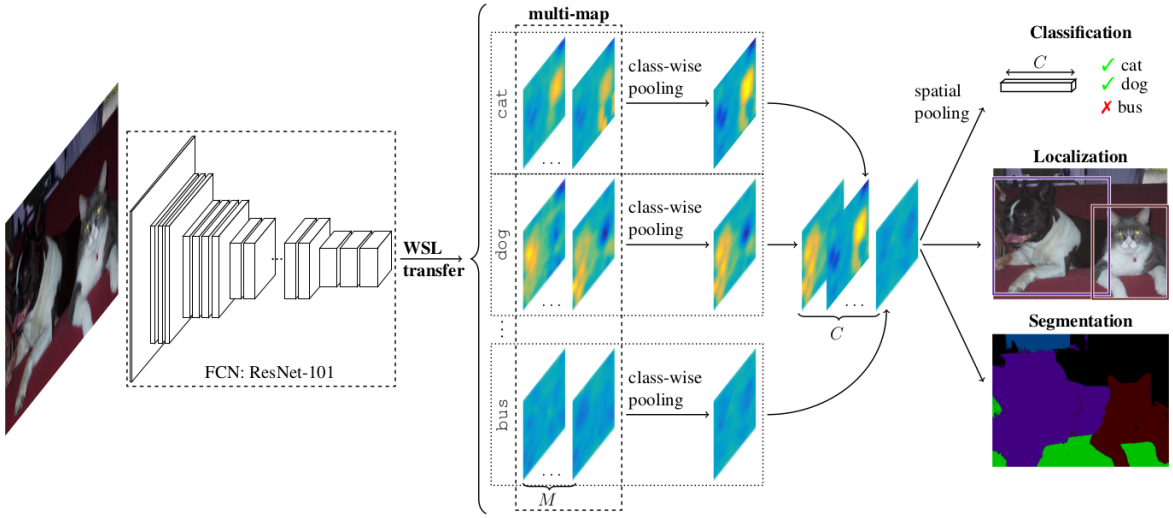


Figure 2.5: WILDCAT architecture, a weakly supervised learning pipeline for three tasks [24]. A deep learning technique that tries to align image regions to gain spatial invarience and thus learning strongly localized features in weakly supervised setting.

The difference between strong supervision and weak supervision is illustrated in Figure 2.6. In this, the trade-off line is specified with regular/strong supervision. The annotations during

training are on X-axis and labels obtained during testing are on Y-axis. If the annotation pair is below trade-off it is strongly supervised and if the pair is above the trade-off line it means weakly supervised. This can also be explained as coarse annotations are available during training and rich labels are obtained during testing.

2.2 Types of Classification

In deep learning or machine learning, classification is the task of assigning class labels to given input from the problem domain [26]. For example, a classical task would be given an email classifying it as "spam" or "ham" is a classification task. According to [26], there are different types of classification tasks based on the input dataset.

2.2.1 Binary Classification

Binary classification directs to those tasks that have only two category labels. Generally, binary classification tasks consist of one class as a normal state and another class as an abnormal state. In this type of classification commonly model predicts a Bernoulli probability distribution [26] for a given input. As we know Bernoulli distribution is a discrete distribution that will assign a binary outcome for the event.

Examples include

- Email spam detection
- Churn prediction
- Conversion prediction

2.2.2 Multi-class Classification

Multi-class classification directs to those tasks that have three or more categories labels. In this, there is a range of states unlike normal and abnormal states in binary classification. A good example would be a face recognition system where the model has to predict a person from the number of faces learned by the system. In multi-class classification tasks model predicts Multinoulli probability distribution [26]. Which is a discrete distribution that covers a case where the outcome for a given event is categorical [26].

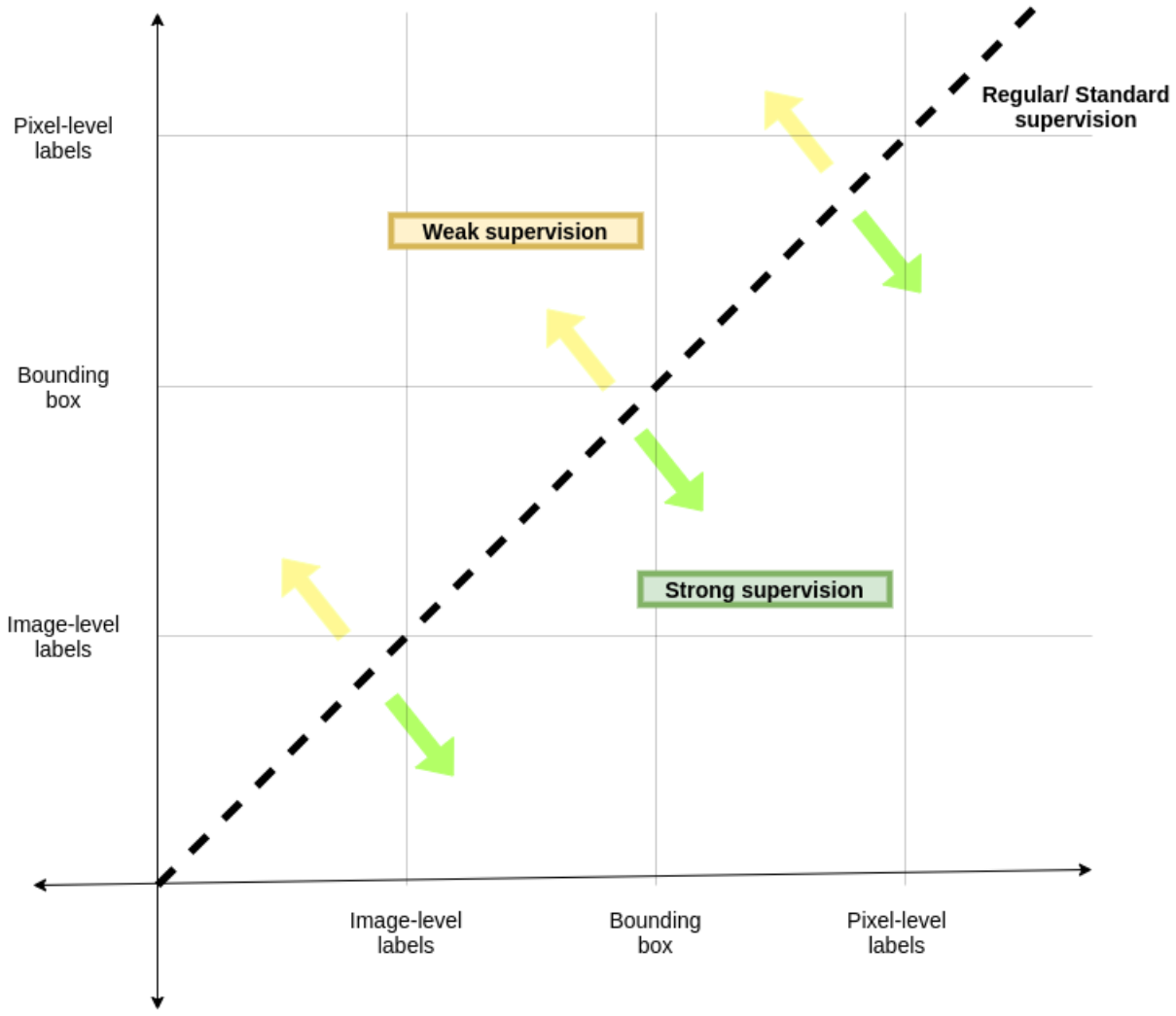


Figure 2.6: Weak supervision versus strong supervision, inspired from [25]. In this the trade-off line is specified with regular/strong supervision. The annotations during training are on X-axis and labels obtained during testing is on Y-axis. If the annotation pair is below trade-off it is strongly supervised and if the pair is above trade-off line it means weakly supervised.

Examples include

- Face classification
- Plant species classification
- Sequence generation (Special case)

2.2.3 Multi-label Classification

Multi-label classification is an extension for Multi-class classification there should be three or more categories labels and additionally one or more class labels will be predicted for a given input. In this type of classification tasks model predicts multiple outputs for a given input in which each outcome as predicted as Bernoulli probability distribution.

Examples include

- Selection of mobile base-station
- Semantic scene classification

3

Literature

In this chapter, the literature overview on Generic object proposal techniques, Unsupervised representation learning techniques, and Weakly Supervised Object Localization (WSOL) techniques are presented. A systematic literature survey is done based on the guidelines provided in [27].

3.1 Search Criteria

Search for literature is formulated based on PICOC (Population, Intervention, Comparison, Outcome, and Context) suggested by [27].

1. **Population:** In the research context population refers to an application area or a target group. In our case population are scoping studies on object localization using unlabeled data.

2. **Intervention:** Similarly intervention includes methodology, technology, tools, and so on. In our case generic object proposal techniques, unsupervised representation learning techniques weakly supervised object localization techniques

3. **Comparison:** In this study, we compare various techniques that are subsets of the above-mentioned interventions. This comparison is based on the defined metric which shows performance of methods based on recall and latency in particular.

4. **Outcome:** The measurable outcome of the study is identifying the appropriate method that solves the object localization problem in a defined experimental setting which is explained in Section 1.3.

5. **Context:** Only deep learning-based methods are expected for object localization.

   The identified keywords are shown in Figure 3.1 which were used to develop search strings. These keywords are directly derived from definitions of PICOC. In Figure 3.1, the color

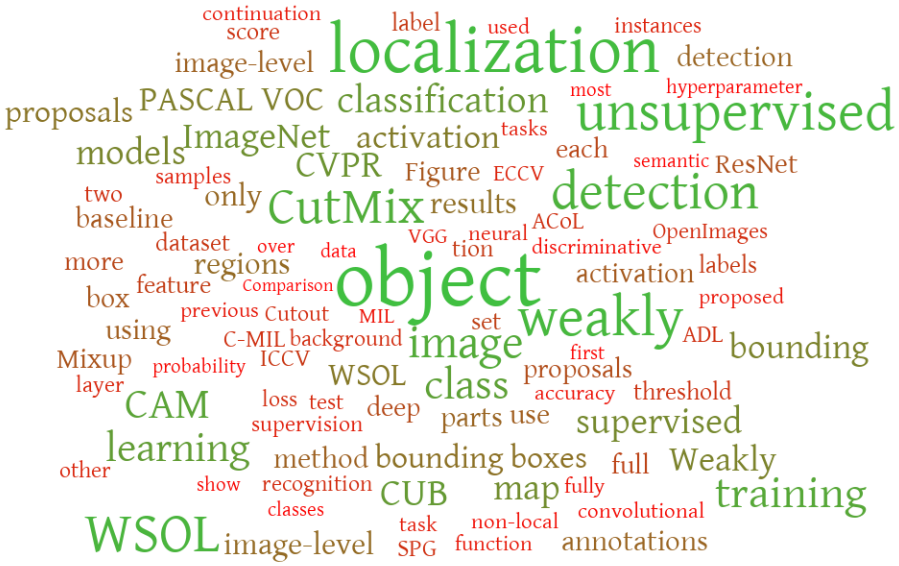


Figure 3.1: Keywords used for developing search string, color and size mapping from green to red and large to small are based on the rank of the keyword.

map from green to red and size large to small is based on the rank of the keyword. The rank of the keyword is based on the frequency of usage in a search string.

3.2 Generic Object Proposal Techniques

Generic object proposals are very important for the preprocessing of given images to use them in computer vision tasks like object classification and detection. Generic object proposals mean finding whether any object present in the current search window irrespective of category. Few prominent works for objectness detection in window are [28] [29]. Apart from objectness measure, methods like Binarized Normed Gradients (BING) [30] proposes an object by taking the norm of gradients in a binary state, and localization is performed using Multi Thresholding Straddling Expansion (MTSE) post-processing. The improvement of BING is another novel algorithm BING++ [31] which retains the proposal efficiency and improves the localization accuracy of the model by precisely capturing the object boundaries. The above methods use a single technique to propose objects. In the novel approach, selective search [32] they proposed using diverse image parts to handle all the possible scenarios.

General object proposal techniques can be divided into three types based on the technique used to find the objects in an image. Types of generic object proposal techniques are presented in the following sections.

3.2.1 Segment Grouping Methods

In segment grouping methods, multiple segments are generated from an image. These segments have a probability for the existence of objects. These segments are generated by the hierarchical segmentation of an image. Thus obtained segments are grouped or merged based on similarity scores between the segments. Results from this grouping method highly depend on the choice of the segmentation algorithm. Some examples of grouping methods are given below:

- Selective Search [32]: It exploits the combination of two approaches exhaustive search and segmentation. As in exhaustive search, this method attempts to find all the possible objects using multiple techniques instead of utilizing a single method. This approach introduces different possible conditions in images by partitioning image. Segmentation helps in guided samplings by exploiting the image structure.

- Constrained Parametric Min-Cut (CPMC) [33]: This method develops and rates possible objects in a given image using bottom-up algorithmic techniques and intermediate selection cues.

- Endres and Hoiem [34]: This method generates a group of segmentation outputs using graph cut methods. These graph cut techniques get a random initial region to start with and learns a affinity function. Thus obtained regions are ranked by learning structure from numerous cues.

- Randomized Prim [35]: In Randomized Prim, connectivity between superpixels in an image are used to model the weights of edges in spanning trees. These weights are learned based on the probability of superpixels belonging to the same object. Thus building a large random spanning tree with a high expected sum value of edge weights will generate the object proposals.

- Rantalankila (global and local search) [36]: This is a class independent search based on a group of superpixels. This replaces the exhaustive search and as it is done class independent manner computation cost to generate object proposal reduces. This combines the local and global search techniques where local search exploits the relation of connectivity between superpixels. While global search performs graph cut segmentation on the superpixel clusters obtained from local search to provide final sets of region proposals.

- Multiscale Combinatorial Grouping (MCG) [37]: This approach is a bottom-up technique that utilizes the hierarchical image segmentation and developed on normalized cuts technique. In this method, a powerful image segmenter is implemented that fuses information

at multi-scale levels. Finally, a grouping scheme generates accurate object proposals by combining multiscale regions.

- Geodesic Object Proposals [38]: Object proposals are generated by diagnosing important level sets in geodesic distance transforms (GDT) [38] that are obtained from anchor points placed in the image. These anchor points are positioned by classification models that are optimized to find out objects.

3.2.2 Window Scoring Methods

In window scoring methods, windows of different sizes are generated and moved across the whole image. Each window is assigned a score based on the probability of containing an object. This approach is faster compared to grouping methods but comes with a downfall on accuracy being lesser accurate. Below are some of the object proposal methods which are based on the window scoring approach.

- BING [30]: This technique uses the fact that objects candidates can be found when they have well-defined boundaries. Each object can be discriminated by finding the norm of gradients at the boundaries. This method uses a 8 x 8 sliding window to generate the norm of gradients. To improve the performance this method used binarized version of the norm of gradients which is Binarized Normed Gradients (BING) [30]. Thus extracted features from BING are used to generate generic object candidates.

- EdgeBoxes [39]: This explores the sparse information that is carried in the edges. These edges are responsible for separating blobs in contour detection. As this is a window-based method, the presence of an object in the window can be given as a likelihood estimate of the number of contours in the window. A simple objectness score is assigned as a function of the number of edges in a window box and members of contours that overhang the boundary of the given box.

- Objectness [28]: In this, measuring the objectness score of a given image window is done using a Bayesian framework. This Bayesian framework quantifies the measure of a few characteristics of objects compared to the background given various image cues. The measured score from the characteristics will be the objectness of the image windows.

- Rahtu [40]: The cascades are a proven framework to boost object detection efficiency. In this method, initial layers object detection cascade are taken and a large number of windows are sampled from objectness prior. Thus found objectness is used to order or rank the image windows to generate final object candidates.

- Using Cascaded Ranking SVMs [41]: It is a cascaded implementation of ranking SVM that generates a set of proposals within the given window that contains object instances. The top-ranking proposals are the final object candidates.

- Structured Forest for Fast Edge Detection [42]: This method capitalizes on the advantage of the underlying structure in local image patches. Local edge masks are predicted using a structured learning algorithm applied to random decision trees. Thus learned decision trees are used to quantify the objectness of local patches.

3.2.3 CNN based Methods

In these methods, a fully Convolution Neural Network (CNN) accepts images as inputs and set of bounding boxes are given as output with an objectness score. These bounding boxes i.e, object proposals are generated by sliding a small network over the activation maps, output calculated by the last convolutional layer in general region proposal networks. However, this may vary from method to method, below are few CNN based object proposal techniques.

- Overfeat [43]: This network is used to all three vision tasks classification, localization, and detection. In the localization pipeline network learns to generate locations of objects using a regression network at the final layers. Thus locations are predicted at multiple scales and multiple locations across the image. Then bounding boxes are accumulated over multiple scales to generate bounding boxes with higher confidence.

- Boosting Convolutional Features [44]: This method is an alternative to selective search [32] algorithm which gives proposals based on superpixels. But there are drawbacks to these kinds of methods as reproducibility with superpixels is unstable. This novel boosting approach will make use of hierarchical features extracted from CNNs for localizing regions of interest.

- Learning to Segment [45]: In this approach, discriminative convolutional networks are used to generate object candidates. The model is trained to achieve two objectives given an input image patch: the first one being generating class-agnostic segmentation patch and second part output likelihood of the given patch contains total object at the center. During testing, a full image is passed through the trained network which generates segmentation masks and each is assigned an object likelihood score.

- Deepbox [46]: Proposals are generated from the bottom-up method and ranked according to the image cues. DeepBox uses CNNs to rank generated proposals again. A four-layer network is used to quantify the objectness and improves the generalization across categories that the network has never seen.

- DeepProposal [47]: Generates hypotheses by sliding window moving over different activation layers of CNNs. In this method, the inverse cascade pipeline is implemented by moving the sliding window setup from final layers to initial layers to combine the best of activation maps. This achieves refining bounding box proposals from coarse to fine grain. Thus object proposals are generated very efficiently as it avoids dense evaluation of the object candidates due to inverse cascading.

- Faster RCNN [48]: Cost-free region proposals are generated using Region Proposal Networks (RPN) that shares input images with existing detection networks. An RPN is a convolutional network that extracts object bounds along with the objectness score. This RPN is trained end-to-end which is capable of generating accurate region proposals.

- HyperNet [49]: This method overcomes the drawbacks of having several thousands of proposals which decreases the efficiency of the detection network by using Hyper Feature [49]. Hyper Feature is aggregate generated by combining hierarchical activation maps and then compressing them to a uniform space. Hyper Features incorporate significant semantic and high-resolution features of an input image. Thus, enables to implementation HyperNet by sharing the Hyper Features to generate proposals and detecting objects by doing end-to-end training.

	Pascal VOC	ImageNet	BSDS 500	MS COCO	NYU Depth
Selective Search [32]	✓	✓	✗	✗	✗
CPMC [33]	✓	✗	✗	✗	✗
Endres and Hoiem [34]	✓	✗	✓	✗	✗
Randomized Prim [35]	✓	✗	✗	✗	✗
Rantalankila (Global and local search) [36]	✓	✗	✗	✗	✗
Multiscale Combinatorial Grouping [37]	✓	✗	✓	✗	✗
Geodesic Object Proposals [38]	✓	✗	✗	✗	✗
BING [30]	✓	✗	✗	✓	✗
EdgeBoxes [39]	✓	✗	✗	✗	✗
Objectness [28]	✓	✗	✗	✗	✗
Rahtu [40]	✓	✗	✗	✗	✗
Using cascaded ranking SVMs [41]	✓	✗	✗	✗	✗
Structured forest for fast edge detetion [42]	✗	✗	✓	✗	✓
Overfeat [43]	✗	✓	✗	✗	✗
Boosting conv features [44]	✗	✓	✗	✗	✗
Learning to segment [45]	✓	✗	✗	✓	✗
Deepbox [46]	✓	✗	✗	✓	✗
DeepProposal [47]	✓	✗	✗	✗	✗
Faster RCNN [48]	✓	✗	✗	✓	✗
HyperNet [49]	✓	✗	✗	✗	✗
Total	**17**	3	3	4	1

Table 3.1: Datasets used for various generic object proposal techniques.

3.2.4 Comparison

In this section, a comparison of all the above-stated methods is presented. In Table 3.1, datasets used in experimentation and evaluation of all the methods are listed. These statistics will help in selecting the dataset for the experiment. In Table 3.2, Recall, and Time metrics are compared as the study focus is on knowing the latency present in method and there is no provision for false negatives which can be concluded using recall metric. The presented metric is directly taken for **respective publications**.

	Recall (%) (IoU≥0.7)	Time (seconds)
Selective Search [32]	87	10
CPMC [33]	65	250
Endres and Hoiem [34]	67.7	100
Randomized Prim [35]	80	1
Global and local search [36]	68	10
Multiscale Combinatorial Grouping [37]	83	34
Geodesic Object Proposals [38]	-*	1
BING [30]	29	0.2
EdgeBoxes [39]	87	0.25
Objectness [28]	39	3
Rahtu [40]	70	3
Using cascaded ranking SVMs [41]	-*	-*
Structured forest for fast edge detetion [42]	-*	-*
Overfeat [43]	-*	-*
Boosting conv features [44]	38.7	2
Learning to segment [45]	-*	1.2
Deepbox [46]	87	2.5
DeepProposal [47]	82	0.75
Faster RCNN [48]	-*	0.2
HyperNet [49]	95	1.14

Table 3.2: Comparison of various object proposal techniques which are evaluated on PASCAL VOC dataset. Recall (%) at IoU $\geq$ 0.7 and Time in seconds are compared among all the methods. **-*:** Method not evaluated on either Recall and Time metrics or PASCAL VOC dataset

3.3 Unsupervised Representation Learning Techniques

The availability of labeled data for custom datasets would be really small whereas collecting the unlabeled data is relatively inexpensive and not a hectic task. This unlabeled data can be used for learning representations in an unsupervised manner. Various unsupervised approaches are capable of learning representations that are generalized enough which can be used in various tasks that have a learning component involved. Unsupervised learning methods can be grouped into three types, they are Transformation Equivariant Representations, generative models, and autoregressive models.

3.3.1 Transformation Equivariant Representations Methods

In unsupervised methods emerging principles like Transformation Equivariant Representations (TERs) [50] in this approach, scene structure in a given image can be compactly represented by applying different transformations on the scene. In TERs, notable works are Group Equivariant Convolutions (GEC) [51], Streerable CNN [52] and Group Equivariant Capsule Networks [53].

-*: Method not evaluated on either Recall and Time metrics or PASCAL VOC dataset

3.3.2 Generative Methods

Another widely studied field in unsupervised learning tasks is generative models like Auto-Encoders (AEs) and Generative Adversarial Networks (GANs) [54]. For example, BigGAN [55] and ALI [56] rely on encoding generated from the encoder network to learn the representation from the data. Other notable GANs that are state-of-the-art in unsupervised representation learning are IntroAVE [57] and VEEGAN [58]. As mentioned before in addition to GANs, many variants of Auto-Encoders are being utilized in unsupervised learning architectures. Variational AE [59], Denoising AE [60] and Contractive AE [61] are generative approaches that learn the representation by reconstructing the input data, in AE we train a pair of encoder and decoder which are inference and reconstructor components respectively.

3.3.3 Autoregressive Methods

One more dimension in unsupervised learning is self-supervised learning methods which are autoregressive models, representations are learned based on predicting missing data, future data, and the context in the data. Few models are PixelRNN [62], PixelCNN [63] and Transformer [64]. In addition to autoregressive models, in Context Encoder [65] pseudo labels from data can be derived from the context of two different parts of data, colorization of the image can also be used to derive labels in unsupervised manner [66]. Very recent developments in the other approaches include Exampler-CNN [67] that generate surrogate classes, clustering e.g, DeepCluster [68], these methods allocate a pseudo label initially and will train the network in back-propagation and then labels are updated. This kind of approach is under unsupervised learning because no manual effort is being used in labeling the data. Evaluations of unsupervised methods are done on the benchmark datasets such as ImageNet, STL10, Places, CIFAR10.

3.4 Weakly Supervised Object Localization (WSOL) Techniques

According to researchers the levels of supervision in object localization task can be defined at image-level [69], points [70], bounding boxes [71], gaze [72], scribbles [73] or combination of multiple types [74]. WSOL [75] is an object localization technique in which models trained on image-level labels learn to localize the objects. Nowadays WSOL became attractive, as image-level annotations can be obtained with less human effort, less time, and at a much cheaper cost than instance-level or pixel-level labels. In this section, the literature of WSOL is presented and tried to present the research in the field.

The work presented in [76] illustrates the ability of CNNs to have a global average pooling (GAP) layer before classifiers to localize objects explicitly although they are trained on image-level

labels. In [76], Class Activation Mapping (CAM) is proposed which generates a score map from a fully-convolutional classifier by manipulating activations before the GAP layer. However, the initial CAM approach is criticized for just using small discriminative parts in the images for localizing the objects. So techniques like Hide-and-Seek (HaS) [77] and Cutmix [78] are proposed where few patches in input images are randomly dropped to diversify the cues. In HaS [77], hide some patches in training images randomly and force the network to learn other less important parts when most discriminative parts randomly are hidden. Whereas in Cutmix [78] authors stated "patches are cut and pasted among training images where the ground truth labels are also mixed proportionally to the area of the patches." Apart from these, adversarial techniques [79], [80] are also proposed which dynamically drops the most significant patch in the given image. In [79], Adversarial Complementary Learning (ACoL) activation maps from penultimate convolution layer are used to enerate class localization maps. Mid-level features are extracted and passed into two parallel-classifiers for finding interdependent object regions. In this method, activation maps from two classifiers A and B are combined to obtain object maps thus locating the objects. Similarly in [80], Attention-Based Dropout Layer (ADL) based localization tries to hide the most discriminative parts in the image and highlight informative regions to improve WSOL accuracy. This is a lightweight and powerful method based on the self-attention mechanism is used to remove the most discriminative part.

Self-produced guidance (SPG) [81] utilizes local correlation between pixels to generate SPG masks. SPG maks are capable of separating foreground and background. These masks are used to provide supervision during training. Mid-level features are extracted from inputs and fed into SPG for classification. CAMs generated from classification networks are used to by SPG to gradually learn the refined CAMs. Then another SPG net uses these refined CAMs as supervision to further improve the final output. In recent developments [13] it has been proven that methods mentioned above are not performing better than more primitive CAM [76] approach. So, approaches like Gradient Class Activation Mapping (Grad-CAM) [15] uses gradients unlike weights of classifier in CAM [76]. These generate more accurate class activation maps and removes the dependency of having a GAP layer in the architecture to generate a CAM. Grad-CAM is the generalization of CAM which can be applied to any CNN based architectures without doing any modifications. A further improvement over Grad-CAM is Grad-CAM++ [82], which provides better network visualization than Grad-CAM and improves the localization of objects and finding all occurrences of multiple objects that belong to the same category. Reformulated the structure of weights in Grad-CAM++ which uses a weighted sum of positive partial derivatives from activation maps of last convolutional layer.

	ImageNet	CUB-200	OpenImages	PASCAL VOC	MS COCO
[13]	✓	✓	✓	✗	✗
[83]	✓	✓	✗	✗	✗
[75]	✓	✓	✗	✗	✗
[78]	✓	✓	✗	✗	✗
[80]	✓	✓	✗	✗	✗
[84]	✓	✓	✗	✗	✗
[85]	✓	✓	✗	✓	✗
[86]	✓	✓	✗	✗	✗
[12]	✓	✓	✗	✗	✗
[87]	✓	✓	✗	✓	✗
[88]	✗	✗	✗	✓	✗
[89]	✓	✓	✗	✗	✗
[90]	✓	✓	✗	✗	✗
[91]	✗	✓	✗	✗	✗
[92]	✗	✗	✗	✓	✗
[93]	✓	✗	✗	✗	✗
[81]	✓	✓	✗	✗	✗
[79]	✓	✓	✗	✗	✗
[94]	✗	✗	✗	✓	✓
[95]	✗	✗	✗	✓	✗
[96]	✗	✗	✗	✓	✗
[97]	✗	✓	✗	✗	✗
[98]	✓	✗	✗	✓	✓
[77]	✓	✗	✗	✗	✗
[99]	✓	✗	✗	✓	✓
[24]	✗	✗	✗	✓	✓
[100]	✗	✗	✗	✓	✗
[76]	✓	✓	✗	✗	✗
[101]	✗	✗	✗	✓	✗
[102]	✗	✗	✗	✓	✓
Total	**19**	17	1	13	5

Table 3.3: Datasets used in various WSOL techniques that are included in final literature.

In this research, experimentation is performed on CAM [76] and Grad-CAM [15], these are evaluated on different datasets against the metric proposed in [13] which are discussed in Chapter 5. In Table 3.3, datasets used in various WSOL techniques are presented, this statistics are used to choose datasets is discussed more in Chapter 4. A brief summary of the timeline of WSOL is presented using tables in Appendix A which include a few interesting methods that are not discussed in this Chapter. Each table consists of a glance about the paper, proposed methodology or technique, metric used for evaluation, and SOTA in which they are evaluated.

4

Datasets

Image based datasets are required for evaluation of selected methods and rank the selected methods based on evaluation metrics. In this chapter, discussion on dataset requirements, available datasets, the selection process of the dataset and dataset analysis are presented.

4.1 Dataset Requirements

- Dataset should contain images.
- Image level annotations available for training i.e, the dataset should support classification tasks.
- Object-level annotations required for evaluation i.e., bounding boxes should be available for calculation of evaluation metrics.
- Multi-class and multi-label datasets are encouraged.

4.2 Available Datasets

In this section, an overview of different benchmarking datasets used in object classification, detection, segmentation tasks are presented in Table 4.1.

Dataset	**Number of classes**	**Number of images**	**Annotation types**	**Multiple instances**	**Objects per image**	**Image size** (pixels X pixels)	**Data format** (Image, Annotation)
PASCAL VOC [8]	20	11540	Classification, Detection, Segmentation	Yes	2.4	470x380	jpeg, xml
MS-COCO [103]	80	328000+	Object and keypoint detection,Stuff and Panoptic segmentation,Image Captioning	Yes	7.3	640x480	jpeg, json
ImageNet [104]	22000	14 million+	Detection, Attribute learning	Yes	1.5	500x400	jpeg, xml
STL10 [105]	10	13000 labeled 100000 unlabeled	Detection, Attribute learning	Yes	-	500x400	jpeg, xml
Tiny ImageNet [106]	2000	120000	Detection, Attribute learning	Yes	1.5	500x400	jpeg, xml
Open Images [107]	600	9.2 million	Classification, Detection, Visual relation detection	Yes	8.3	varied	URL, csv
Places [108]	434	10 million+	Scene recognition	No	-	256x256	jpeg, array
Caltech-UCSD Birds 200 (CUB-200) [109]	200	6033	Classification and detection	No	1	varied	-
CIFAR10 [110]	10	60000	Classification	No	1	32x32	png, csv
CIFAR100 [110]	100	60000	Classification	No	1	32x32	png, csv
Yale-CMU-Berkeley (YCB) [111]	77	46200	Classification, detection, segmentation	No	1	4272x3848	jpeg, pbm (mask)
Robocup @Work*	12	720	Classification, detection, segmentation	No	1	640x480	jpeg, xml

Table 4.1: Comparison of various datasets

*: Custom collected dataset

4.3 Dataset Selection

Dataset for further experimentation and evaluation is selected based on the statistics derived from the literature study. PASCAL VOC [8] dataset is selected because we can conclude from Table 3.1 and Table 3.3 that PASCAL VOC is mostly benchmarked dataset in evaluation of methods. Apart from this fact, more reasons for selecting PASCAL VOC dataset are as follows:

- Just 20 classes and smaller datasets easy to handle.
- Other datasets like MS COCO, ImageNet are not considered as they have more number of images which makes training slower.
- All 20 classes are common objects in everyday life.
- Significantly large intra class variations.
- Contains many difficult examples.
- Mostly used in object localization and detection tasks.
- Object-level ground truth is available, in case of evaluation.
- Multiple instances in the same image, so that multiple objects can be localized in the same image.
- Collected data in realistic scenes.

Along with PASCAL VOC, YCB dataset and RoboCup@Work* datasets are also selected as we are also interested in solving the issues faced by b-it-bots @Home and @Work teams which are mentioned in Section 1.3.

4.4 Dataset Description

In this Section, detailed description of each selected dataset is presented.

4.4.1 PASCAL VOC

The PASCAL VOC dataset consists of four splits of data which can be for tasks mentioned below:

- Classification/Detection Task

*: Custom collected dataset

- Segmentation Task
- Action Detection Task
- Person Layout Taster Task

	Train		Validation	
	Images	**Object Instances**	**Images**	**Object Instances**
Aeroplane	327	432	343	433
Bicycle	268	353	284	358
Bird	395	560	370	559
Boat	260	426	248	424
Bottle	365	629	341	630
Bus	213	292	208	301
Car	590	1013	571	1004
Cat	539	605	541	612
Chair	566	1178	553	1176
Cow	151	290	152	298
Diningtable	269	304	269	305
Dog	632	756	654	759
Horse	237	350	245	360
Motorbike	265	357	261	356
Person	1194	4194	2093	4372
Pottedplant	269	484	258	489
Sheep	171	400	154	413
Sofa	257	281	250	285
Train	273	313	271	315
Tvmonitor	290	392	285	392
Total	**5717**	**13609**	**5823**	**13841**

Table 4.2: Statistics of the classification/detection image sets.
Note: All the statistics are taken from The Pascal Visual Object Classes (VOC) Challenge [8].

In this research, the classification/detection task dataset is used for training classification and evaluating metrics accuracy, precision, and recall. Whereas segmentation task dataset is used to evaluate against the metrics MaxBoxAccV2 [13] and PxAP [13]. Each set contains three data splits *train*: training data, *val*: validation data and *trainval*: the union of train and val data splits. Only *val* data split is used from segmentation dataset. The dataset contains 20

object categories they are Aeroplane, Bicycle, Bird, Boat, Bottle, Bus, Car, Cat, Chair, Cow, Diningtable, Dog, Horse, Motorbike, Person, Pottedplant, Sheep, Sofa, Train, Tvmonitorwhich. These are collected from Flickr. The average image size is 470x380 pixels and average objects per image are 2.4. Table 4.2 summarizes the number of images and instances per class present in classification/detection dataset and similarly Table 4.3 provide summary for segmentation set.

	Train		**Validation**	
	Images	**Object Instances**	**Images**	**Object Instances**
Aeroplane	88	108	90	110
Bicycle	65	94	79	103
Bird	105	137	103	140
Boat	78	124	72	108
Bottle	87	195	96	162
Bus	78	121	74	116
Car	128	209	127	249
Cat	131	154	119	132
Chair	148	303	123	245
Cow	64	152	71	132
Diningtable	82	86	75	82
Dog	121	149	128	150
Horse	68	100	79	104
Motorbike	81	101	76	103
Person	442	868	445	865
Pottleplant	82	151	85	171
Sheep	63	155	57	153
Sofa	93	103	90	106
Train	83	96	84	93
Tvmonitor	84	101	74	98
Total	**1464**	**3507**	**1449**	**3422**

Table 4.3: Statistics of the segmentation image sets.
Note: All the statistics are taken from The Pascal Visual Object Classes (VOC) Challenge [8].

4.4.2 Yale-CMU-Berkeley (YCB)

Yale-CMU-Berkeley (YCB) is a benchmarking dataset for robotic grasping and manipulation tasks. A total of 77 objects are categorized into 5 types: Food items, Kitchen items, Tool items,

Shape items, Task items. In a given image only one instance of the object is present. In total, for each object, the dataset contains the following:

- 600 RGB-D images
- High resolution RGB images
- Each image have semantic mask
- Camera calibration data for every image
- Texture-mapped 3-D mesh models

Images used in experimentation are 12.2 megapixel resolution. Each category of objects is given as follows. Food items in the YCB object set: Chips can, Coffee can, Crackerbox, Box of sugar, Tomato soup can, Mustard container, Tuna fish can, Chocolate pudding box, Gelatin box, Potted meat can, Plastic fruit (lemon, apple, pear, orange, banana, peach, strawberries, plum). Kitchen items: Pitcher, Bleach cleaner, Glass cleaner, Plastic wine glass, Enamel-coated metal bowl, Metal mug, Abrasive sponge, Cooking skillet with glass lid, Metal plate, Eating utensils (knife, spoon, fork), Spatula, White table cloth. Tool items: Power drill and Wood block, Scissors, Padlock and keys, Markers (two sizes), Adjustable wrench, Phillips- and flat-head screwdrivers, Wood screws, Nails (two sizes), Plastic bolts and nuts, and Hammer, Spring clamps (four sizes). Shape items: Mini soccer ball, Softball, Baseball, Tennis ball, Racquetball, Golf ball, Plastic chain, Washers (seven sizes), Foam brick, Dice, Marbles, Rope, Stacking blocks (set of 10), Credit card blank. Task items: Box and blocks test, 9 hole peg test, Timer, Lego Dublo, Magazine, Rubick's cube, T-shirt, Airplane toy.

4.4.3 RoboCup@Work

RoboCup@Work dataset consists of a total of 12 classes which are Axis, Bearing, Bearing box, F20 20 black, F20 20 gray, M20, M20 100, M30, Motor, R20, S40 40 black and S40 40 gray. This dataset is collected in-house using the automatic data collection and augmentation tool *Easy Augment* [112]. A brief discussion on *Easy Augment* tool and how it use it is presented in Appendix B. In this dataset, each image consists of only a single object instance. Each category contains the following:

- 60 RGB images with 640x480 resolution
- Corresponding depth frames
- Segmentation masks for each image

- Point cloud of a segmented object
- Bounding box annotations in PASCAL VOC format

4.5 Dataset Analysis

In this section, statistical analysis is performed on selected datasets. This analysis will help us understand the distribution of datasets i.e., either a dataset is balanced or imbalanced. With the help of this inference, we can comment on the results obtained in the experimentation phase on how the distribution of the dataset affects the model.

According to [113], imbalance in dataset can be measured using Shannon entropy. Shannon entropy can be defined with Equations 4.1 and 4.2, where n is total number of instances, k is number of classes and c_i is count of instances for i^{th} class.

$$H = -\sum_{i=1}^{k} \frac{c_i}{n} \log \frac{c_i}{n} \tag{4.1}$$

$$Balance = \frac{H}{\log k} \tag{4.2}$$

The measure of balance is quantified using Equation 4.2 which tends to 0 for unbalanced dataset and tends to 1 for a balanced dataset. From the data analysis shown in Figure 4.1 we can see that PASCAL VOC dataset is relatively imbalanced and other two datasets are balanced. Further analysis is done at class for PASCAL VOC and illustrated in Figure 4.2 where distribution of number of images per class is presented. From the Figure 4.2, it is clearly understood that *Person* class is dominant. The distribution of dataset will have an impact on training of classification model. The impact of this dataset imbalance is discussed more in Chapter 6.

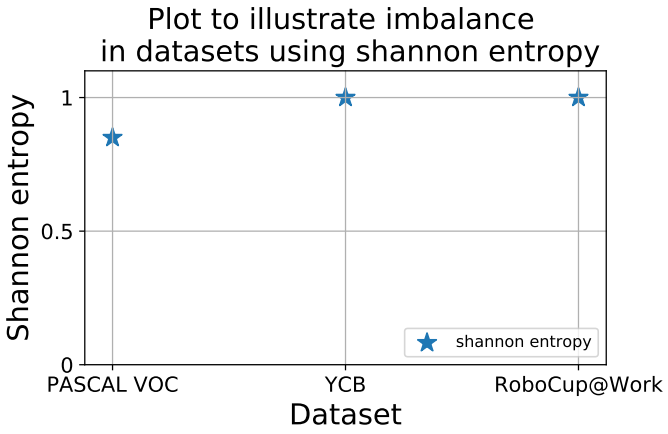


Figure 4.1: Measure of dataset balance is illustrated using Shannon entropy.

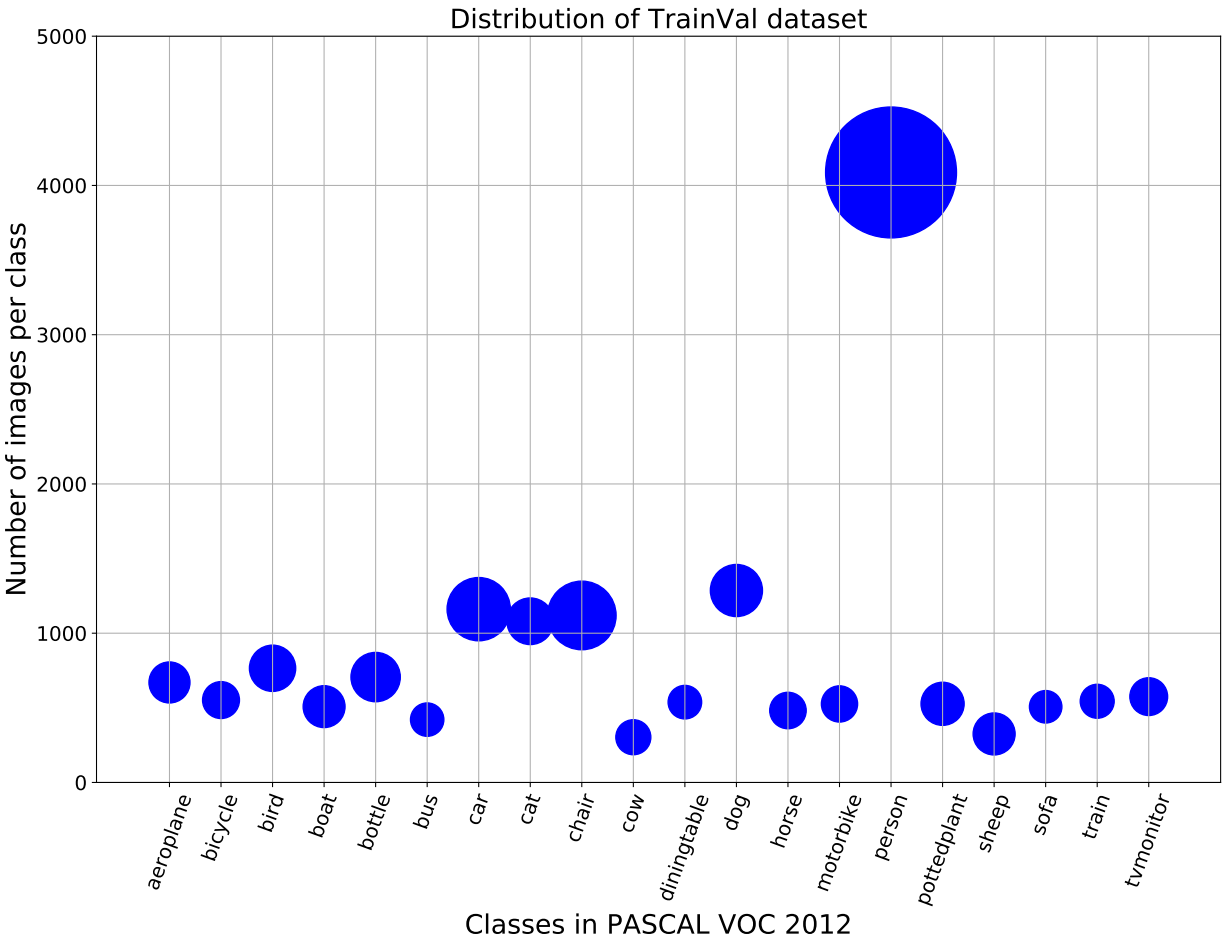


Figure 4.2: Distribution of PASCAL VOC 2012 dataset, size of marker specifies Number of instances per class.

5

Methodology

In this chapter, method selection based on guidelines of [27] is discussed. As well as experimentation pipeline and evaluation metrics for all the selected methods are described in the following sections.

5.1 Method Selection

Baseline method and other SOTA are selected from the literature presented in Chapter 3. Methods are selected based on the data extracted and synthesized from the literature. The following selection criteria are set to select a method. Methods are selected based on:

- Frequency of benchmarking
- Metrics used for proving it
- SOTA methods on which it is evaluated against
- Availability of source code
- Backbone and dataset used
- Abstract and approach

5.2 Pipeline

The experimentation pipeline implemented for training a classification model is shown in Figure 5.1. In this, a multi-label classification is trained for the PASCAL VOC, YCB and RoboCup@Work datasets described in Chapter 4 using three different backbones namely VGG16, ResNet18, and Squeezenet1_1. Similarly evaluation pipeline for calculating classification metric and localization metrics is illustrated in Figure 5.2. Methods used to generate heatmaps in evaluation is discussed in Sections 5.3 and 5.4. A brief discussion on measured metrics is done in Section 5.5.

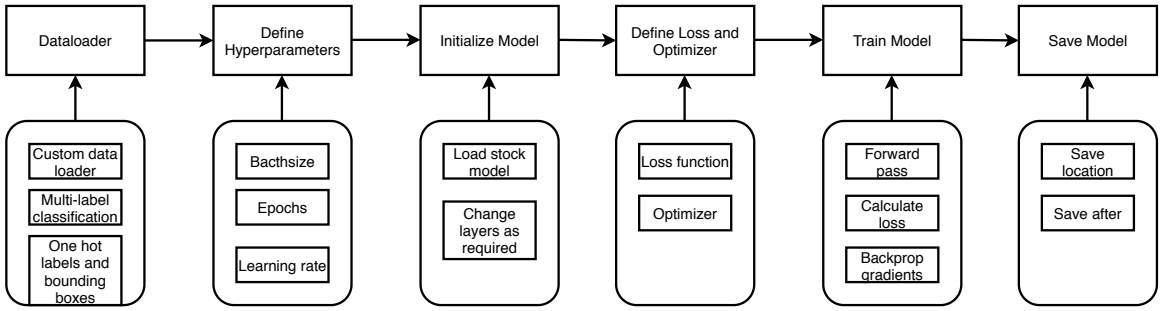


Figure 5.1: General pipeline for training the classification model.

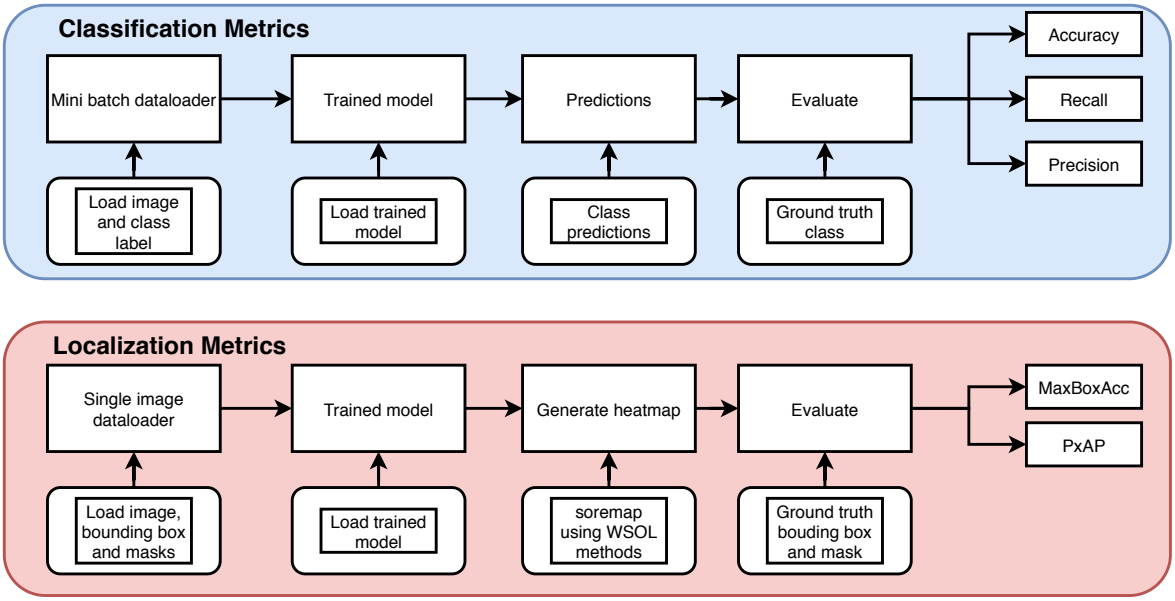


Figure 5.2: General pipeline for evaluating the classification and localization metrics.

5.3 Class Activation Mapping

The method that is used as a basic approach that satisfies the problem statement is known as the baseline method. This baseline method is used to compare other SOTA methods selected. The baseline method is selected based on the above-mentioned selection criteria and Figure 5.3 shows the frequency of benchmarking of different methods. From Figure 5.3, Class Activation Mapping (CAM) [76] is selected as baseline method. For experimentation and evaluation code implementation is adapted from [13] and GitHub repository **link***.

*https://github.com/clovaai/wsolevaluation/

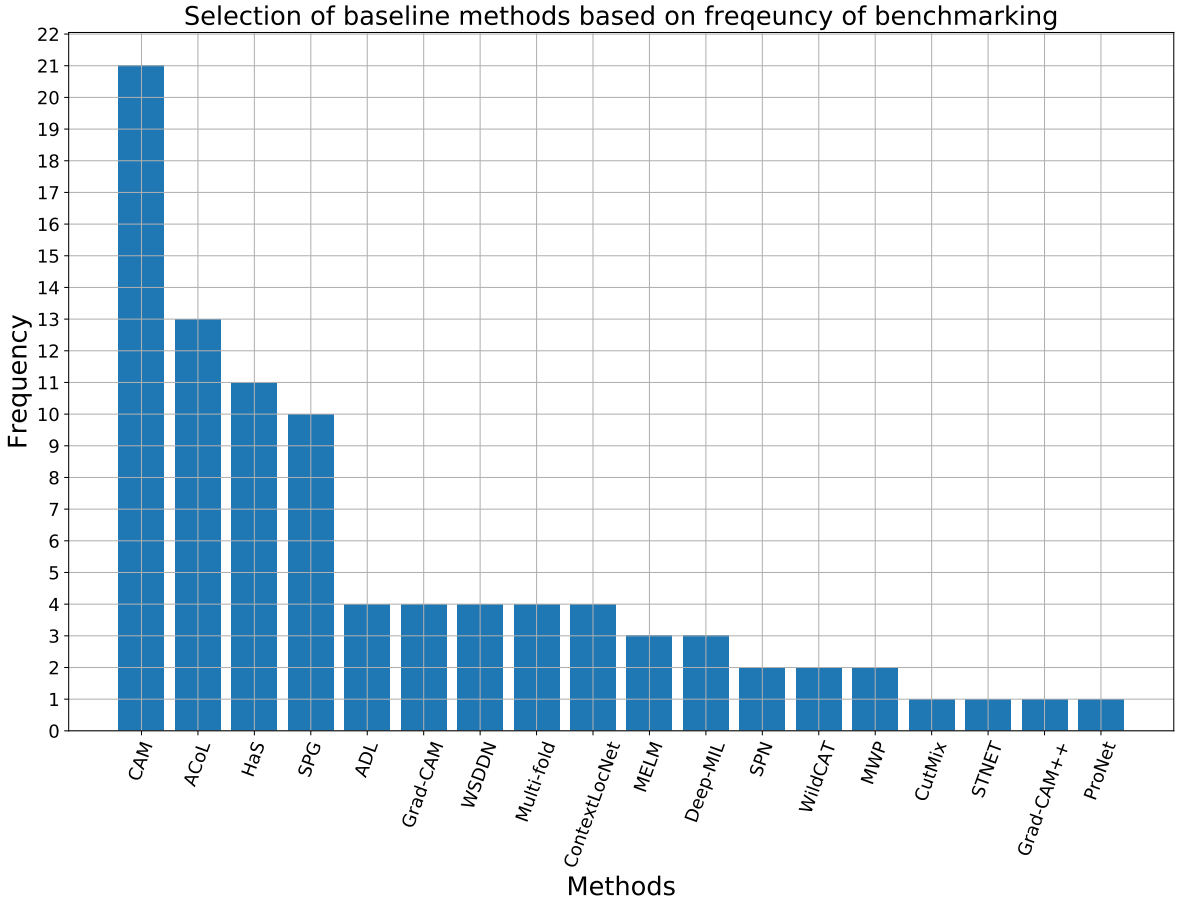


Figure 5.3: Selection of baseline method based on benchmark count taken from papers that are selected using defined inclusion and exclusion criteria.

In [76], authors showcased that CNNs have a very good capability to predict the location of objects although they are trained for classification tasks. CAM for a selected category describes the distinctive regions of images, in which CNNs used to classify the selected category object. To achieve this, an average pooling layer is added in between the last convolution layer and a fully connected classification network. These discriminative regions are identified by projecting weights of the classification output layer that corresponds to the selected category on to the feature maps from the end convolution layer. This is called Class Activation Mapping and the weighted sum of the activation maps and output layer weights will result in Class Activation Map for the selected category which is illustrated in Figure 5.4.

In the Class Activation Mapping technique proposed by [76], a drawback is that architectures that are performing global average pooling (GAP) over activation maps immediately before

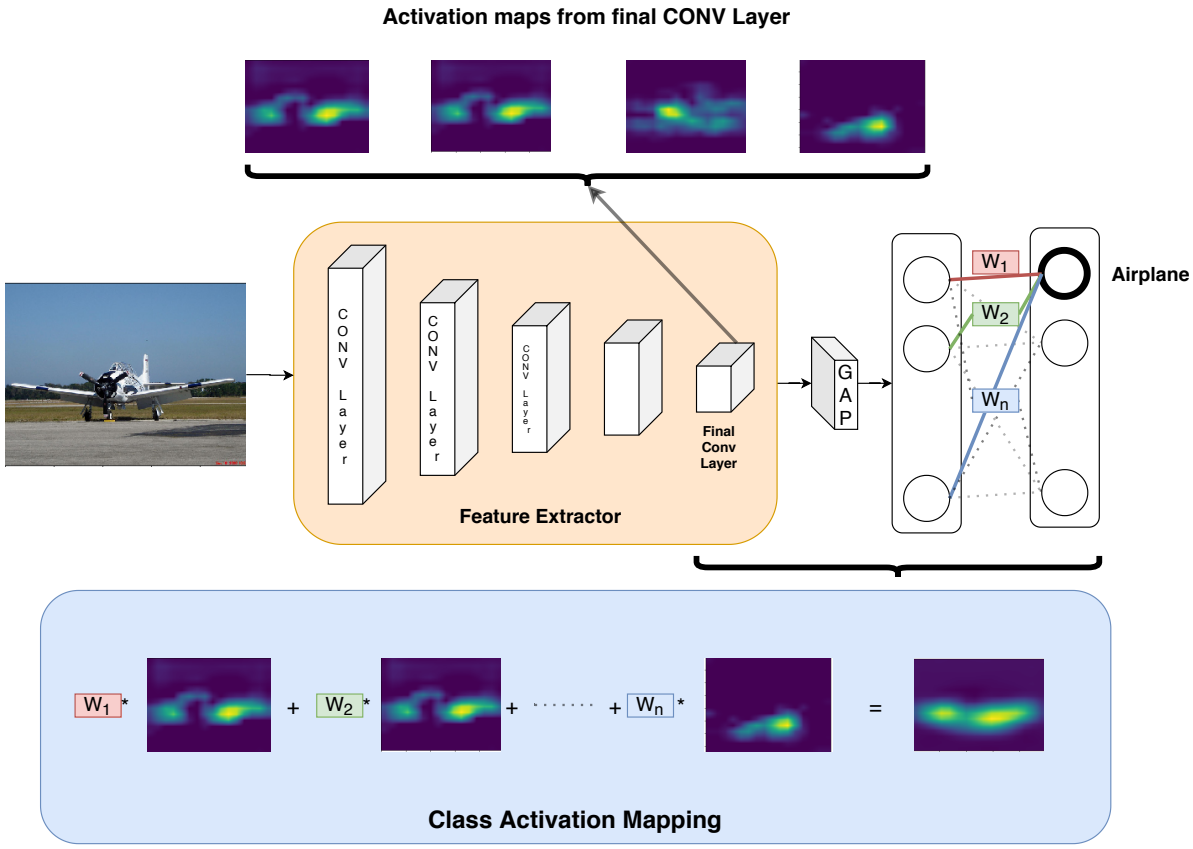


Figure 5.4: Class Activation Mapping: the confidence score of predicted class is projected back to the preceding convolution layer to generate the class activation maps (CAMs). The CAM highlights the category-specific significant fields. Inspired from [76]

classifier layers are only applicable. Such architectures may perform poorly relative to the general network on some tasks. So, next method to perform WSOL is chosen to overcome the drawback in CAM which is Grad-CAM [15] which is discussed in Section 5.4. This is not the only reason to select Grad-CAM as another method to evaluate. According to [13], other SOTA WSOL methods like HaS [77], ACoL [79], SPG [81], ADL [80] and CutMix [78] have either very slight improved or no improvement over vanilla CAM [76] when evaluated against metrics which are discussed in Section 5.5.

5.4 Gradient Class Activation Mapping

Gradients in learning techniques are vectors whose value is a partial derivative of the function and this gradient will be flowing towards the steepest rate of increase of that function. Gradient Class Activation Mapping (Gard-CAM) exploits this information along with class specifics to produce score maps of significant regions in the given image. Grad-CAM combines pixel space gradient information with class discriminative property to generate score maps. The architecture of Grad-CAM is shown in Figure 5.5. In Grad-CAM, spatial location data of an object is preserved so last convolution layers are utilized in the generation as neurons from that layer are capable of identifying significant parts to the given class.

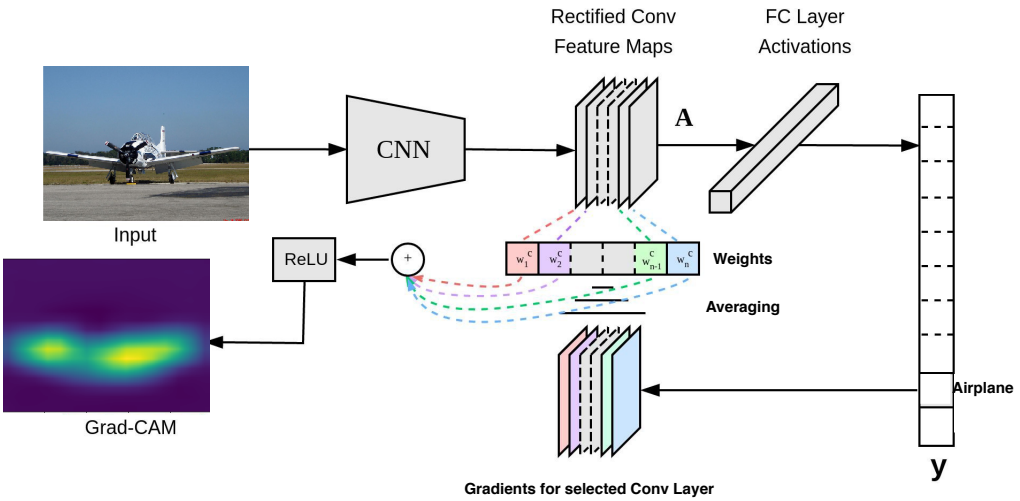


Figure 5.5: Gradient Class Activation Mapping architecture. Grad-CAM uses gradient flow passing through last convolutional layer to generate weights. Then perform weighted sum of activation using these weights to predict locations. Inspired from [15]

To generate a Grad-CAM for a given image and predicted class, gradients of the score for the predicted class (let's say c) y_c concerning feature maps A_k of the selected convolutional layer which is given by $\frac{\partial y_c}{\partial A_k}$. Once gradients are obtained weight that specifies the significance of feature map is calculated using Equation 5.1 where the summation over i and j specify GAP and partial derivatives are the gradients from backpropagation. As shown in Figure 5.5, the weighted sum of activation maps passed through the Rectified Linear Unit (RELU) is performed to get the final score map for predicted class. This summation is described in Equation 5.2. RELU is used to combine the feature maps as it highlights the significant features having a positive affect on predicted class. For experimentation and evaluation code is adapted from GitHub repository **link***.

$$a_c^k = \frac{1}{Z} \sum_i \sum_j \frac{\partial y_c}{\partial A_{ij}^k} \tag{5.1}$$

$$Scoremap_c = RELU \sum_k a_k^c A_k \tag{5.2}$$

5.5 Evaluation Metrics

In this experimentation phase evaluation metrics are chosen such that they will support answering the research questions RQ3, RQ4, RQ5, and RQ6 mentioned in Section 1.3.

5.5.1 Classification Metrics

PASCAL VOC dataset contains multi-label data that is given image can have instances of multiple categories. In this regular metrics cannot be used as performance can be quantified inappropriately. So metrics prescribed in the lecture [114] are used for evaluating classification performance of models.

Classification Accuracy

Classification accuracy is defined as the rate of correct predictions for a given test or validation set. Classification accuracy for multi-label classification is calculated using Equation 5.3, where $\hat{y}^i$ is prediction, y^i is ground truth and N is number of samples.

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} \frac{|\hat{y}^i \wedge y^i|}{|\hat{y}^i \vee y^i|} \tag{5.3}$$

*https://github.com/jacobgil/pytorch-grad-cam

Precision and Recall

Precision and Recall are calculated using scikit learn metrics. Precision is the ratio given by Equation 5.4 and recall is the ratio given by Equation 5.5, where tp is number of true positives, fp is the number of false positives and fn is the number of false negatives. The precision score indicates the ability of trained classification model that will not label negative sample as positive. The recall score measures the ability of trained classification model to find all positive samples.

$$Precision = \frac{tp}{tp + fp} \tag{5.4}$$

$$Recall = \frac{tp}{tp + fn} \tag{5.5}$$

5.5.2 Localization Metrics

The localization accuracy is the measurement of localization performance of trained model. In general Top-1 localization accuracy is used to quantify the localization performance in WSOL methods which uses the normalized scoremap. After normalizing scoremap, WSOL techniques threshold the scoremap at t. This thesholded binary mask is used to generate tight fit bounding box. Generally t is fixed value which gives wrong performance measure for localization as t depends on data and model architecture. So new evaluation metrics are proposed in [13] to overcome the dependency of threshold t. Proposed metrics are discussed below:

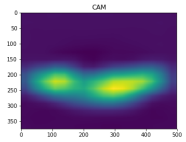
Maximal Box Accuracy (MaxBoxAcc)

This metric can be calculated when bounding boxes are available as ground truth. The box accuracy for heatmap at the given threshold t is defined by Equation 5.6. It is quantified by the proportion of images where the box spawned from the scoremap intersects with the ground truth at multiple IoU thresholds. Each score map is thresholded at range of values to obtain threshold independence in generating binary mask, masks generated at different thresholds are shown in Figure 5.6 and bounding boxes are extracted which is illustrated using Figure 5.7. Final performance metric is mean of MaxBoxAcc across all IoU thresholds given in Equation 5.7. To get the threshold independence of IoU, MaxBoxAccV2 is proposed where mean across different IoU thresholds is calculated given in Equation 5.8, where $box(s(X^{(n)}, t))$ is tightest bounding box, $B^{(n)}$ ground truth bounding box, t is threshold to generate binary mask and t_{IoU} is threshold for IoU.

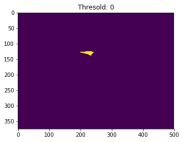
$$BoxAcc(t) = \frac{1}{N} \sum_{n} l_{IoU}(box(s(X^{(n)}, t)), B^{(n)}) \geq t_{IoU} \tag{5.6}$$

$$MaxBoxAcc = max_t BoxAcc(t) \tag{5.7}$$

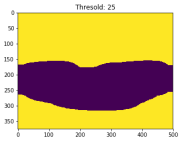
$$MaxBoxAccV2 = mean_{t_{IOU}}(max_t BoxAcc(t)) \tag{5.8}$$



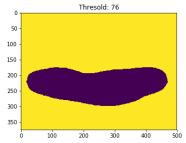
(a) CAM output



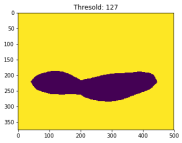
(b) CAM binary image at t=0



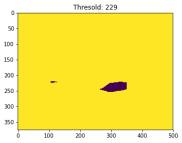
(c) CAM binary image at t=25



(d) CAM binary image at t=76



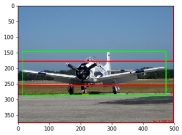
(e) CAM binary image at t=127



(f) CAM binary image at t=229

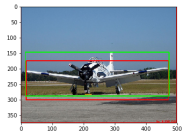
Figure 5.6: CAM output thresholds at different values to get tight fit max IoU bounding boxes.

(a) Bounding box at t=0

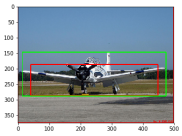


(b) Bounding box at t=25

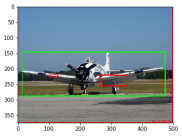
(c) Bounding box at t=51



(d) Bounding box at t=76



(e) CAM binary image at t=127



(f) CAM binary image at t=229

Figure 5.7: Bounding boxes drawn at different thresholds.

Pixel Average Precision (PxAP)

This metric can be calculated when semantic masks are available as ground truth. A precision recall curve if generated at pixel level G and area under the curve is calculated as a measure. Pixel precision and recall at specific threshold is given by Equations 5.9 and 5.10. Threshold independence is achieved by calculating pixel average precision (PxAP) using Equation 5.11.

$$PxPrec(t) = \frac{|\{s_{ij}^{(n)} \geq t\} \cap \{T_{ij}^{(n)} = 1\}|}{|\{s_{ij}^{(n)} \geq t\}|} \tag{5.9}$$

$$PxRec(t) = \frac{|\{s_{ij}^{(n)} \geq t\} \cap \{T_{ij}^{(n)} = 1\}|}{|\{T_{ij}^{(n)} = 1\}|} \tag{5.10}$$

$$PxAP = \sum_{l} PxPrec(t_l)(PxRec(t_l) - PxRec(t_{l-1}) \tag{5.11}$$

Where s_{ij} is value of pixel at (i, j) in scoremap, T_{ij} is value of pixel at (i, j) in ground truth mask, t is the threshold and n is image id.

6

Results

6.1 Impact of Activation Map Size on Model Performance

In this section, results for impact of activation map size on model performance which answers RQ5 mentioned in Section 1.3 are presented. The pipeline for extracting 16x16 pixels activation map size from VGG16 architecture is shown in Figure 6.1. In the original pipeline, a 16x16 pixels activation map is generated for an input image of size 256x256 pixels. Few example stages on how reduction in activation sizes is carried out is presented in Figure 6.1. The experimentation setup is as follows:

- Dataset: PASCAL VOC
- WSOL method: Class activation mapping (CAM)
- Network architecture: VGG16
- Localization metrics: Maximal Box Accuracy (MaxBoxAcc) [13] and Pixel Average Precision (PxAP) [13]
- Classification metric: Accuracy, Precision and Recall

6.1.1 Objective

The objective of this experiment is to assess the model performance on localization and classification metrics when there is a change in the size of the activation maps at the last convolution layer of a deep neural network. This is important as activation maps are interpolated to the size of the input image to predict the location of objects.

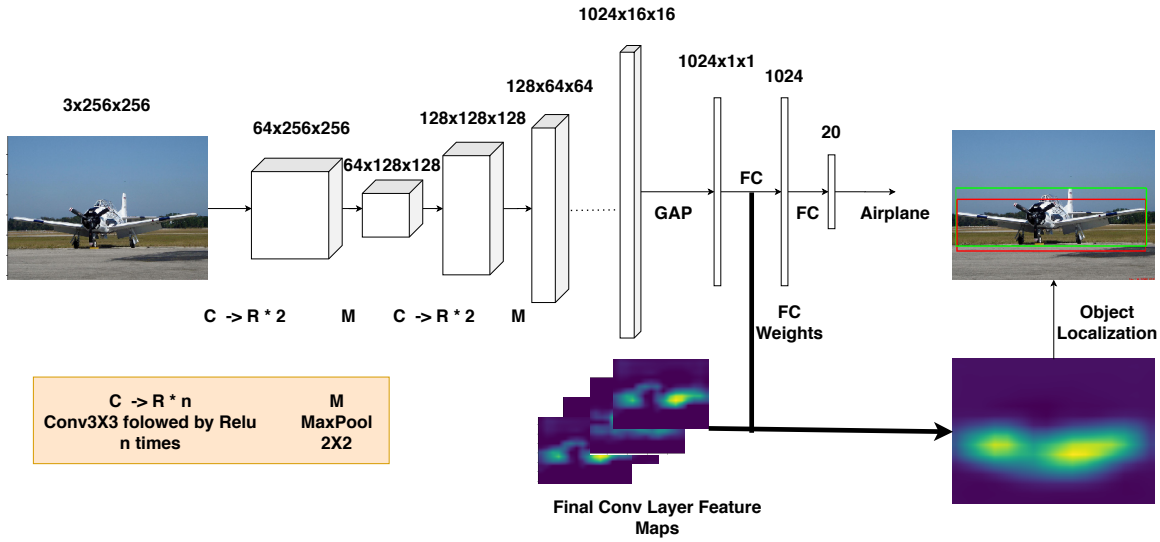


Figure 6.1: The stock pipeline based on VGG16 network architecture to generate 16x16 pixels activation map.

6.1.2 Hypothesis

If the size of the activation map is decreased from the original size it results in poor performance on localization and classification metrics. On the other hand, if the size of the activation map is increased, it should result in better performance than the original size of the activation map.

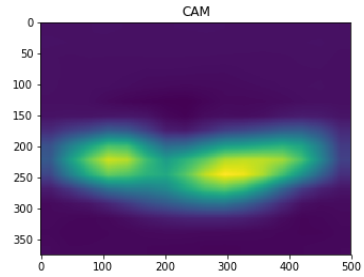
6.1.3 Observation

In this section, two approaches are presented that are implemented to access the impact of the activation map size on model performance. The size of the activation map from stock architecture is 16x16 pixels. Increased activation map size is 32x32 pixels and the reduced activation map size is 8x8 pixels.

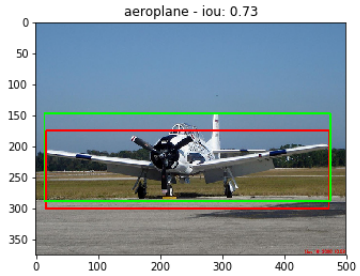
Approach 1

In this approach, changes in network architecture are done in order to change the size of activation map and size of input is kept constant at 256x256 pixels. Stock architecture is given Figure 6.3. The architecture implemented to extract 8x8 pixels reduced activation map is given in Figure 6.4. In Figure 6.4, convolution layers 26 and 28 (highlighted in blue) are changed in order to reduce the size of activation map. Similarly in Figure 6.5, convolution layers 21, 24, 26 and 28 (highlighted in red) are altered to increase the activation map size to 32x32 pixels. Class

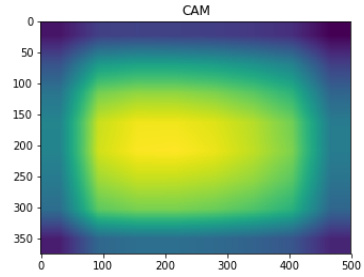
activation maps generated using this approach are illustrated in Figure 6.2. In Figure 6.2, it is seen that localization ability of altered activation map size (32x32 and 8x8 pixels) is reduced compared to the original size of activation maps (16x16 pixels) and it is the initial impression that change in activation map size only worsen the performance which is discussed in detail in Section 6.1.4.



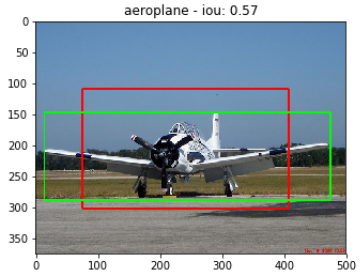
(a) CAM generated from 16x16 pixels activation maps



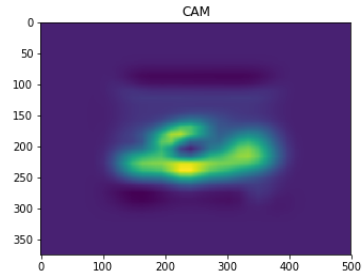
(b) Bounding box with maximum IoU for object



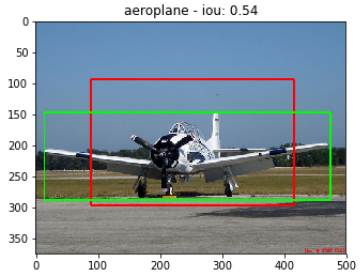
(c) CAM generated from 8x8 pixels activation maps



(d) Bounding box with maximum IoU for object



(e) CAM generated from 32x32 pixels activation maps



(f) Bounding box with maximum IoU for object

Figure 6.2: Visualization for effect of change in activation map size using approach 1. From the class activation maps generated from respective final layer activation maps sizes are on left hand side. It is observed that localization ability is decreasing when there is alteration in activation map size.

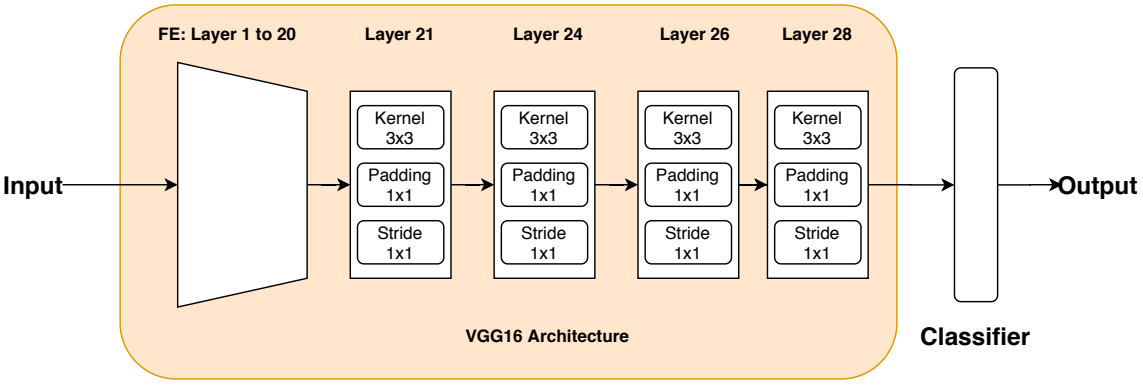


Figure 6.3: Architecture of VGG16 for original activation map size 16x16 pixels.

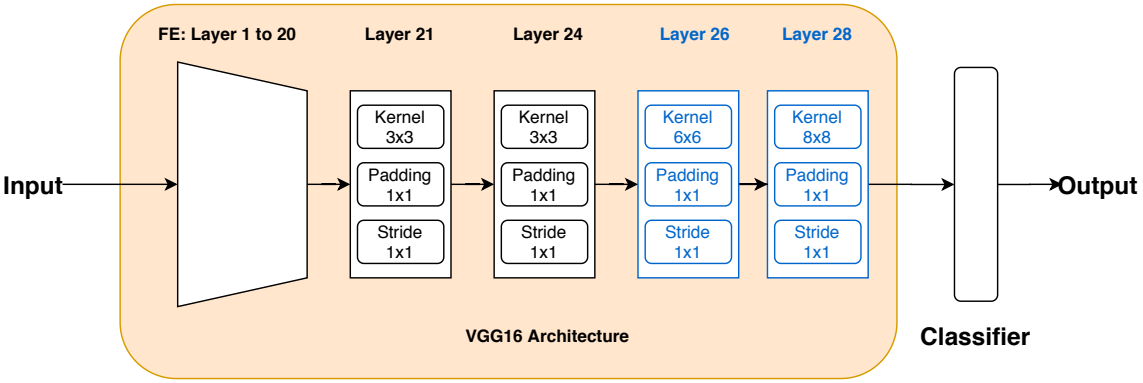


Figure 6.4: Architecture of VGG16 for reduced activation map size 8x8 pixels. The blocks coloured in blue are the modifications made to generate reduced activation map size.

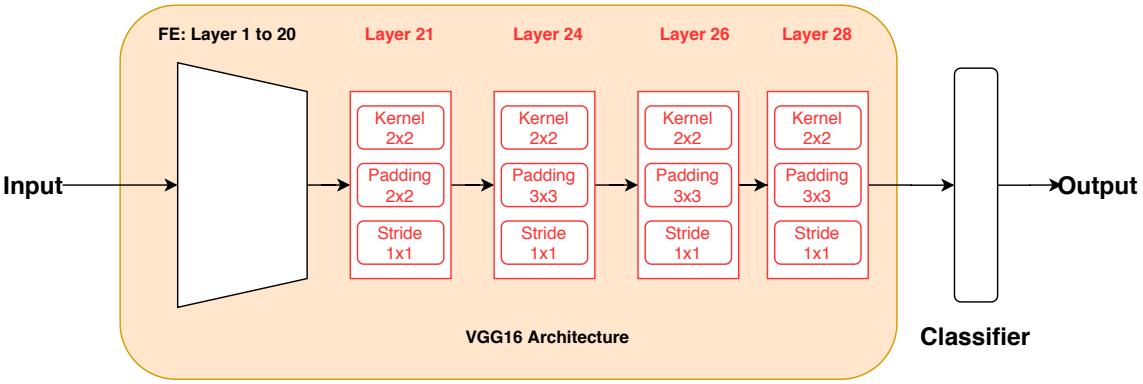
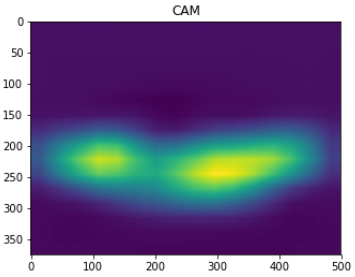


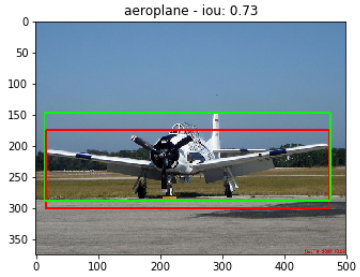
Figure 6.5: Architecture of VGG16 for increased activation map size 32x32 pixels. The blocks coloured in red are the modifications made to generate increased activation map size.

Approach 2

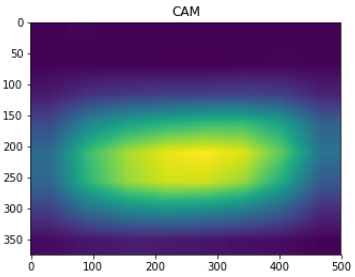
In this approach, the network architecture is not changed and Figure 6.3 is used for all the experiments that are performed. The naive approach followed to change the activation map size is changing the size of the input image. In the original experiment setup, the image size used is 256x256 pixels which result in 16x16 pixels heatmap. So, changing the input image size to 512x512 pixels results in 32x32 pixels heatmap, and altering the input image size to 128x128 pixels will change the activation map size to 8x8 pixels. Even in this approach, it is observed that generated activation maps in altered scenarios will reduce the model's classification and localization performance which can be visualized in Figure 6.6.



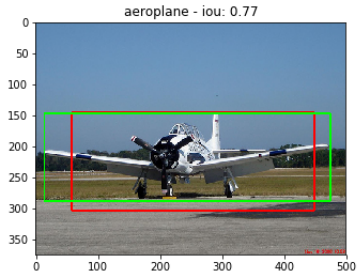
(a) CAM generated for 256x256 pixels image



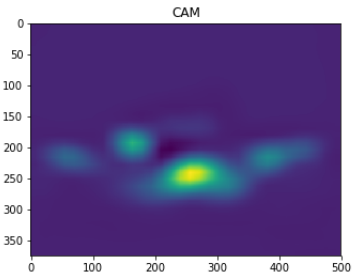
(b) Bounding box with maximum IoU for object



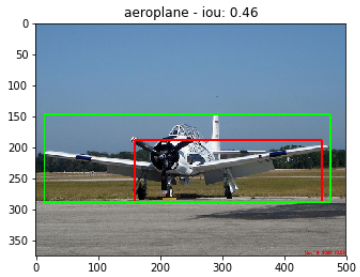
(c) CAM generated for 128x128 pixels image



(d) Bounding box with maximum IoU for object



(e) CAM generated for 512x512 pixels image



(f) Bounding box with maximum IoU for object

Figure 6.6: Visualization for effect of change in activation map size using approach 2. From the class activation maps generated from different input image sizes are on left hand side. It is observed that localization ability is decreasing when there is alteration in input image size.

6.1.4 Verdict

In Figure 6.7, it is understood that altering the network architecture to change the size of the activation map leads to a reduction in the classification and localization performance of the model. In Figure 6.8, there is clearly reduction in localization performance while classification performance in the case of 32x32 pixels activation map is similar to 16x16 pixels map size. So, from the conclusion drawn from Figures 6.7 and 6.8, **assumed hypothesis is rejected**. The

hypothesis is rejected as the reduction in classification and localization performances are observed in both approaches. This drop in performance is attributed to the loss of spatial features in the case of reduced activation map size. The loss of spatial information is due to interpolating the smaller activation map size to actual image size. In the case of increased activation map size, spatial features are concentrated at the center of the activation map due to the extra padding added to specified layers in the architecture.

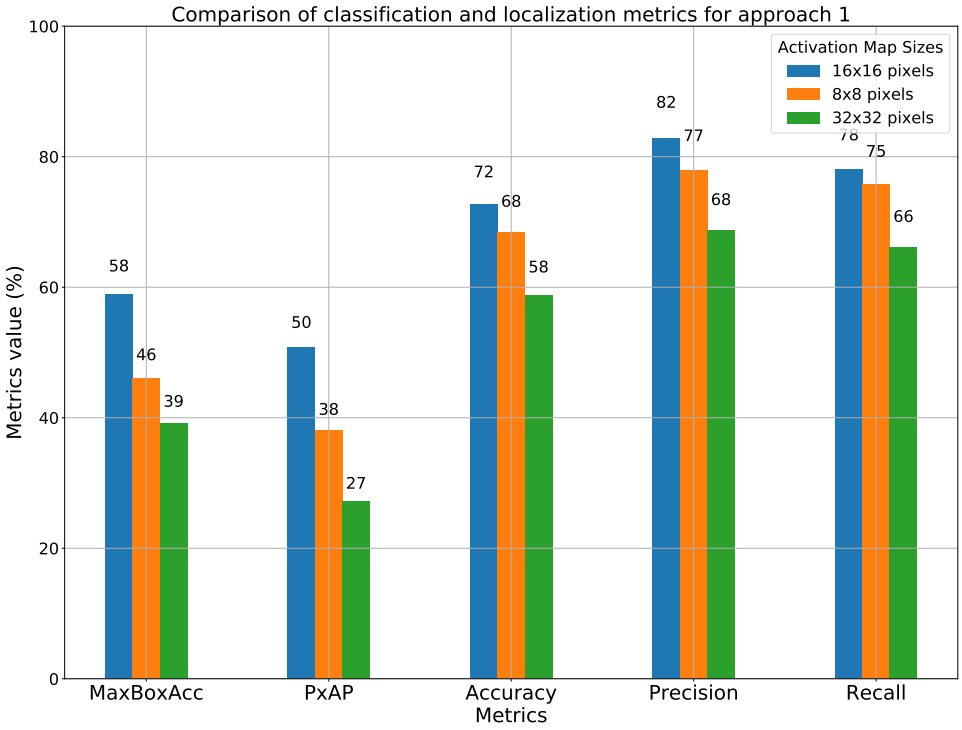


Figure 6.7: Comparison of model performance on localization and classification metrics for change in activation map size using approach 1. From the plot, it can be inferred that classification and localization performance is dropped on modifying the activation map size.

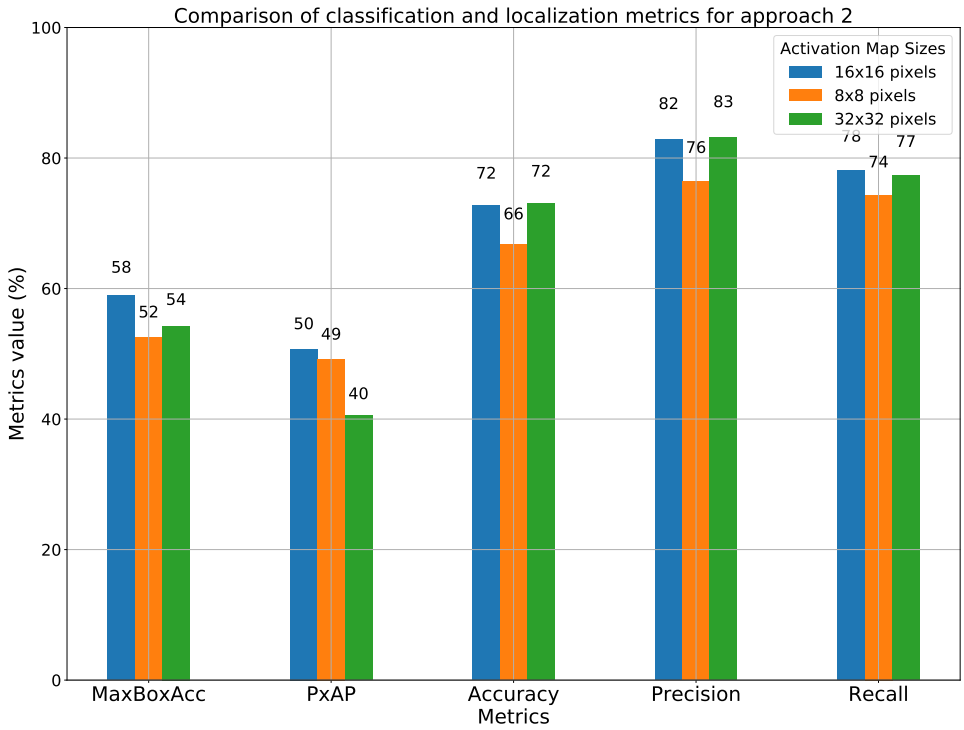


Figure 6.8: Comparison of model performance on localization and classification metrics for change in activation map size using approach 2. From the plot, it can be inferred that classification and localization performance is dropped on modifying the activation map size.

6.2 Analysis on Smaller Network Architectures

Evaluation of two WSOL methods is done on the PASCAL VOC dataset. This evaluation is carried out to support the conclusion for RQ3 mentioned in Section 1.3. The experimentation setup is as follows:

- Dataset: PASCAL VOC
- WSOL method: Class activation mapping (CAM), Gradient class activation mapping (Gard-CAM)
- Network architectures: VGG16 (baseline), ResNet18, SqueezeNet1_1
- Localization metrics: Maximal Box Accuracy (MaxBoxAcc) [13] and Pixel Average Precision (PxAP) [13]

- Classification metric: Accuracy, Precision and Recall

6.2.1 Objective

The objective is to evaluate the performance of CAM [76] and GradCAM [15] on smaller network architectures. The motivation for this experiment is to observe the trade-off between network size and performance of the WSOL method. The size of the network is crucial as smaller networks will have lower latency and can be deployed on machines with lower computational capability. In particular, evaluation on smaller networks will provide a direction toward the annotation free localization model for RoboCup teams at H-BRS.

6.2.2 Hypothesis

The hypothesis for this experiment is that smaller network architectures like ResNet18 and SqueezeNet1_1 will perform poorly in localization and classification tasks compared to larger network architectures like VGG16.

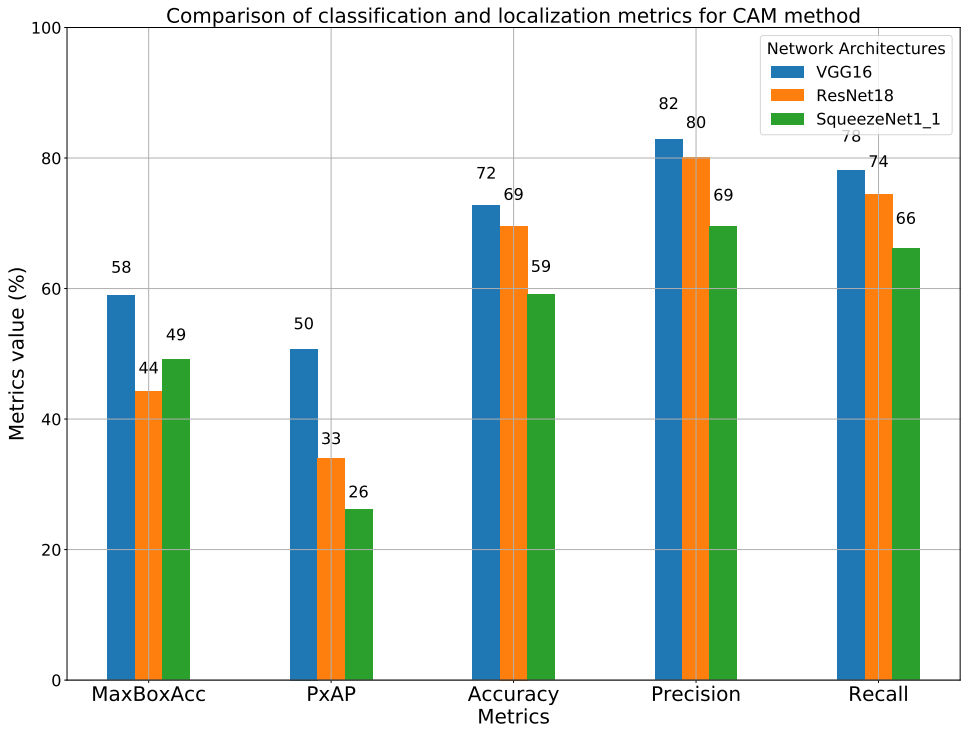


Figure 6.9: Comparison of smaller network architectures performance on localization and classification metrics using CAM WSOL technique. From the plot, it can be inferred that classification and localization performance is dropped on using the smaller networks.

6.2.3 Observation

From Figures 6.9 and 6.10, it is seen that as the network architecture become smaller the performance in localization as well as classification tasks is following a decreasing trend. This reduction in performance is observed across both the WSOL techniques CAM [76] and Grad-CAM [15]. One outlier is seen in experiment of CAM [76] method, where SqueezeNet1_1 is out performing ResNet18 in MaxBoxAcc [13] metric which indicates SqueezNet1_1 is performing better in terms of generating tight fit bounding boxes. But It is the same decreasing trend in pixel-wise localization as PxAP [13] is decreased. In Figure 6.10, localization metrics MaxBoxAcc and PxAP for SqueezeNet1_1 are not presented as there are some implementation challenges occurred during experimentation.

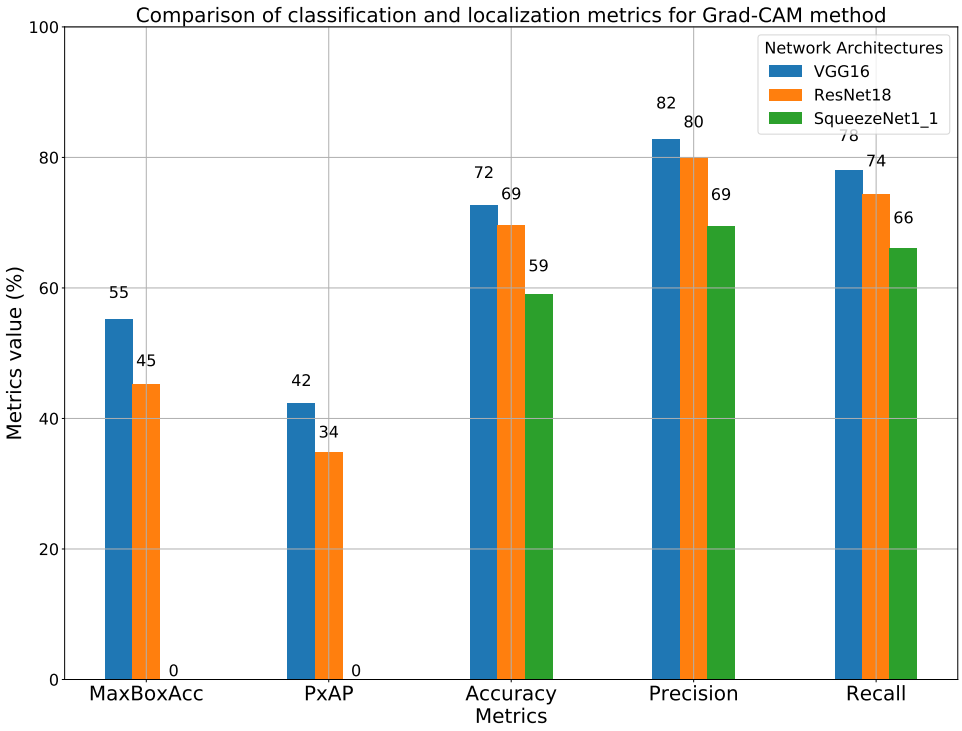


Figure 6.10: Comparison of smaller network architectures performance on localization and classification metrics using Grad-CAM WSOL technique. From the plot, it can be inferred that classification and localization performance is dropped on using the smaller networks.

6.2.4 Verdict

Experimentation results adhere to the assumption in the hypothesis mentioned in Section 6.3.2. Hence, **hypothesis is accepted** and it can be concluded that smaller architectures like ResNet18 and SqueezeNet1_1 perform poorly when compared to larger network architectures like VGG16. This drop in performance can be attributed to reduced quality of activation map in smaller networks.

6.3 Analysis on Different Datasets

The evaluation of two WSOL methods is done on PASCAL VOC, YCB, and RoboCup@Work datasets. This evaluation is carried out to support a conclusion for RQ4 mentioned in Section 1.3. The experimentation setup is as follows:

- Dataset: PASCAL VOC, YCB and RoboCup@Work
- WSOL method: Class activation mapping (CAM), Gradient class activation mapping (Gard-CAM)
- Network architectures: VGG16
- Localization metrics: Maximal Box Accuracy (MaxBoxAcc) [13] and Pixel Average Precision (PxAP) [13]
- Classification metric: Accuracy, Precision and Recall

6.3.1 Objective

The objective is to evaluate the performance of CAM [76] and GradCAM [15] on different datasets namely PASCAL VOC, YCB and RoboCup@Work. These experiments are done to answer the question, whether an imbalanced dataset and smaller datasets affect localization and classification performance. This evaluation helps in deciding on how to collect custom datasets for RoboCup teams at H-BRS.

6.3.2 Hypothesis

The hypothesis for this experiment is that balanced datasets will perform better and more samples in the dataset will give better results.

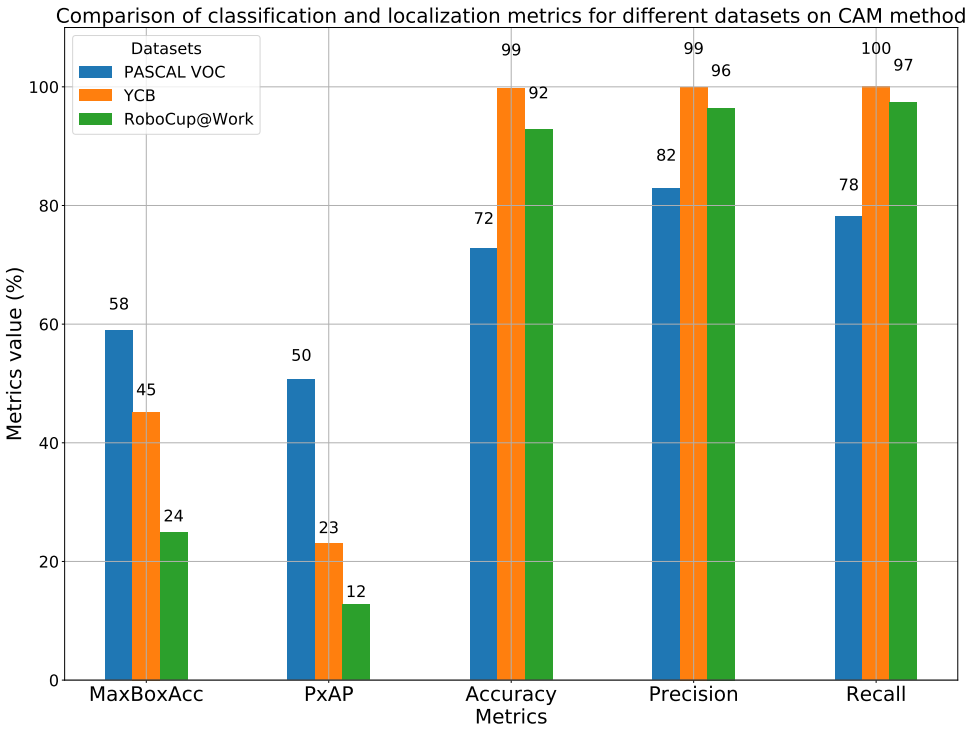


Figure 6.11: Comparison of different datasets performance on localization and classification metrics using CAM WSOL technique. From the plot, it can be inferred that localization performance is poor relatively and classification performance is improved.

6.3.3 Observation

In the CAM method, quite different results are observed in Figure 6.11, where balanced datasets YCB and RoboCup@Work are performing better than PASCAL VOC dataset in terms of the classification task. But, it is also seen that it did not hold for localization tasks, where the PASCAL VOC dataset is performing better. This might be due to the more difficult samples present in the PASCAL VOC dataset, unlike other datasets where the background is pretty much constant. This enforces the network to learn more significant features which will help in better localization. Similar outcomes can be seen in Figure 6.12 for the Grad-CAM method.

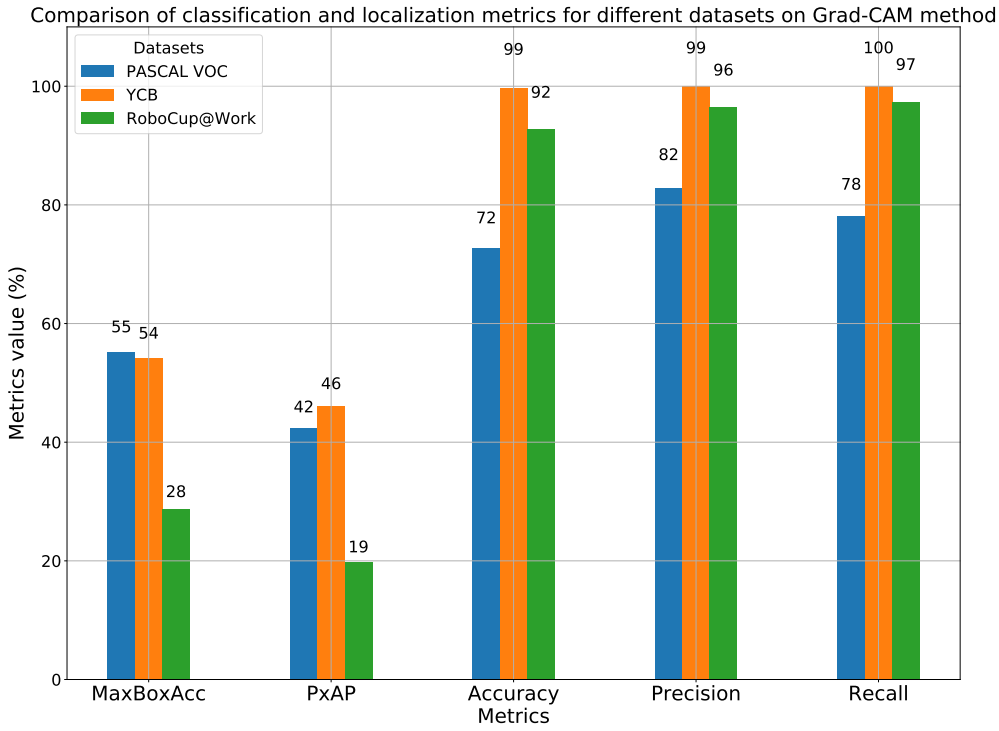


Figure 6.12: Comparison of different datasets performance on localization and classification metrics using Grad-CAM WSOL technique. From the plot, it can be inferred that localization performance is poor relatively and classification performance is improved.

6.3.4 Verdict

In this case, **hypothesis can neither be rejected not accepted** as mixed results were observed in localization and classification tasks. Balanced and easy datasets YCB, RoboCup@Work are performing better in a classification whereas imbalanced and more difficult PASCAL VOC dataset is performing better in localization task. We attribute this change in performance is due to intra-class variance present in the PASCAL VOC dataset, unlike YCB and RoboCup@Work. In YCB and RoboCup@Work datasets, objects captured per class are the same always while they are captured from different angles. In PASCAL VOC, samples consist of different objects captured with different backgrounds.

6.4 Selection of Optimal Threshold

In this section, a novel approach to the select optimal threshold is presented. The optimal threshold value for generating binary masks from class activation maps can be selected from the distribution of thresholds accumulated during the validation phase. The accumulated thresholds are generating good Intersection over Union (IoU) score between generated and ground truth bounding boxes. In Figure 6.13, it can be seen that the threshold value can be selected as 51.0 for the PASCAL VOC dataset. In another approach, thresholds are accumulated at the class level which might give better results during testing as the optimal threshold is defined at the class level. From Figures 6.14 and 6.15, optimal threshold values to generate binary mask for individual classes can be defined. Optimal threshold value for classes are aeroplane: 51.0, bicycle: 51.0, bird: 25.5, boat: 76.5, bottle: 51.0, bus: 76.5, car: 76.5, cat: 25.5, chair: 51.0, cow: 76.5, diningtable: 153.0, dog: 25.5 and so on for the remaining 14 classes.

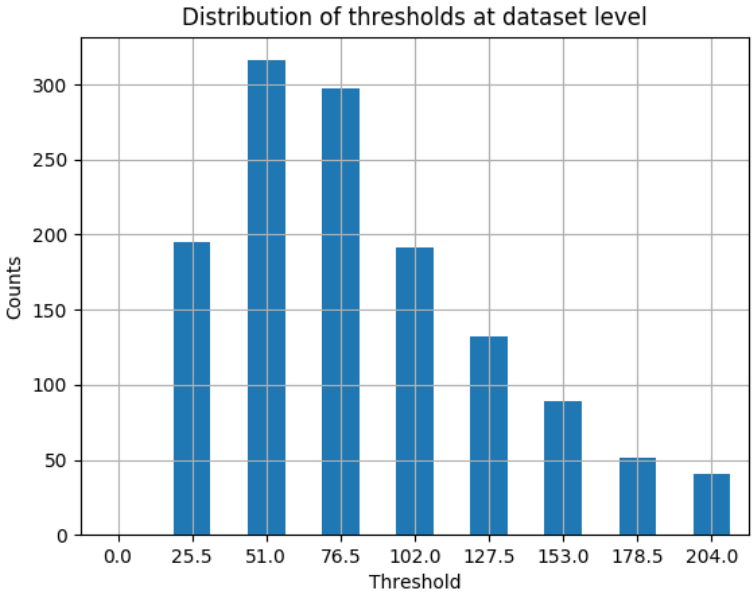
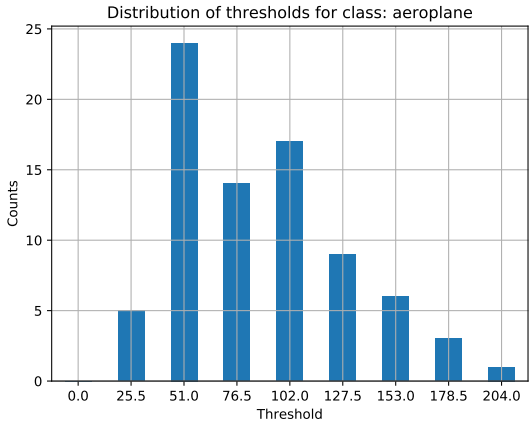
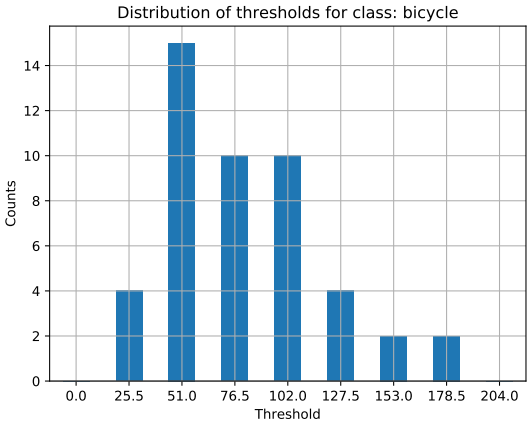


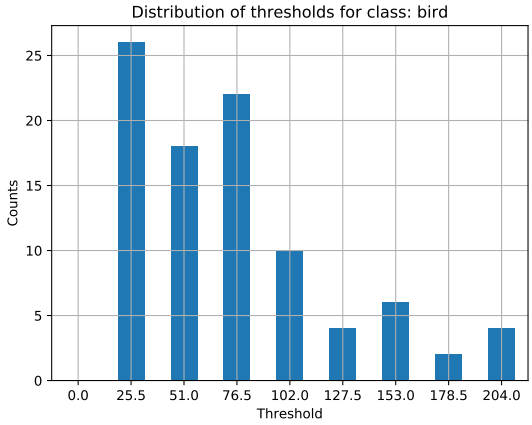
Figure 6.13: Distribution of thresholds taken at dataset level during validation which is used to select optimal threshold.



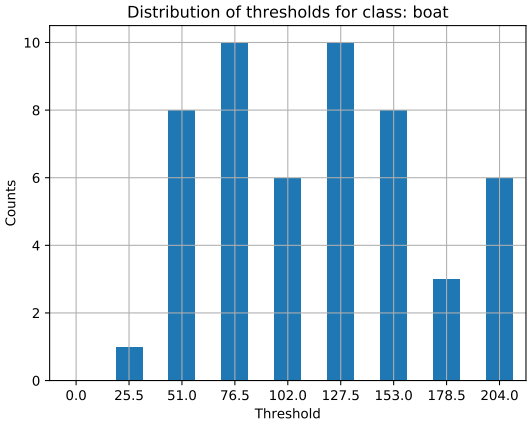
(a) Distribution of threshold for aeroplane class



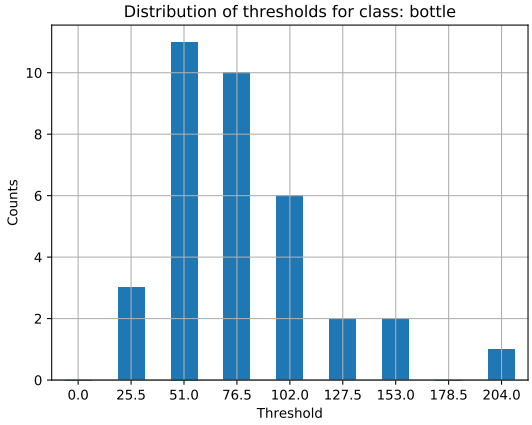
(b) Distribution of threshold for bicycle class



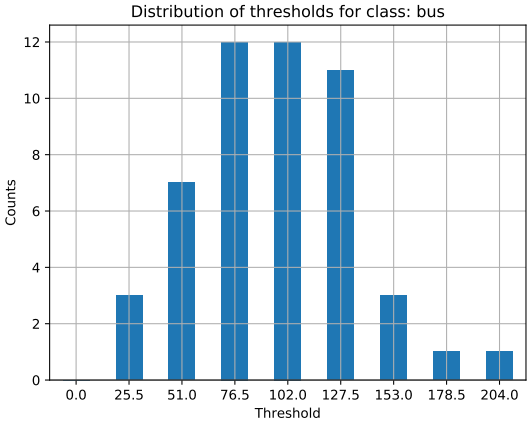
(c) Distribution of threshold for bird class



(d) Distribution of threshold for boat class

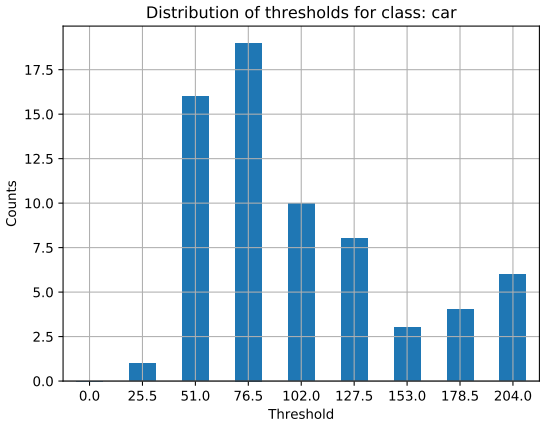


(e) Distribution of threshold for bottle class

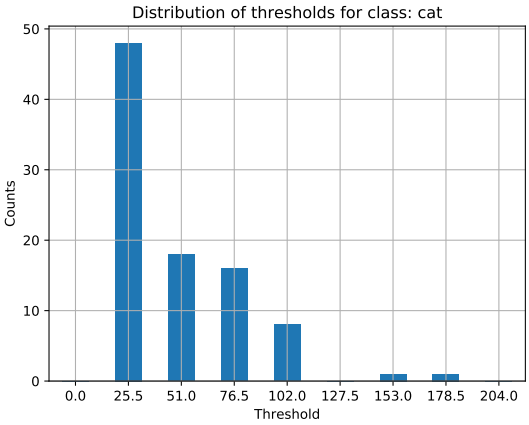


(f) Distribution of threshold for bus class

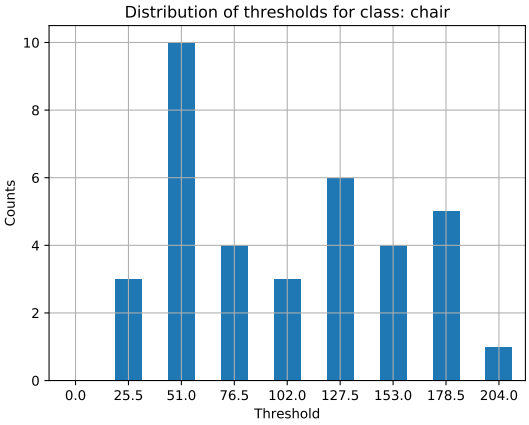
Figure 6.14: Distribution of thresholds taken at class level during validation which is used to select optimal threshold.



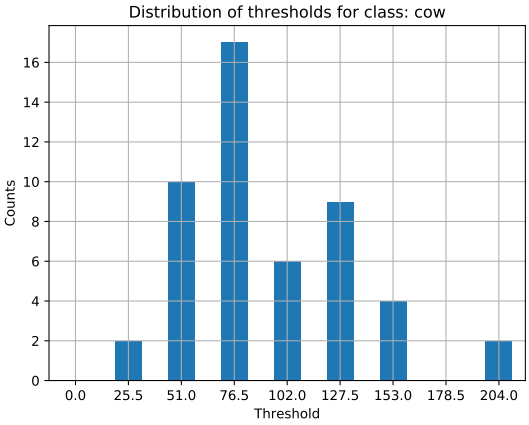
(a) Distribution of threshold for car class



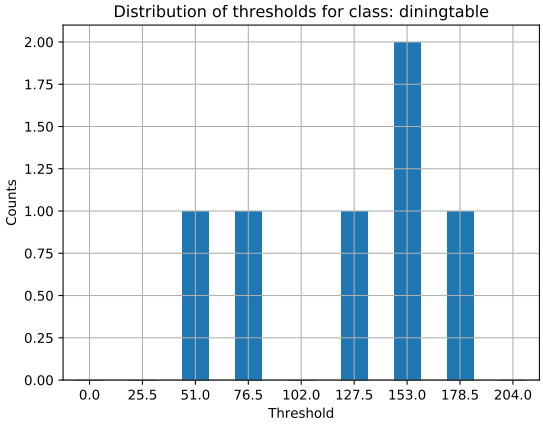
(b) Distribution of threshold for cat class



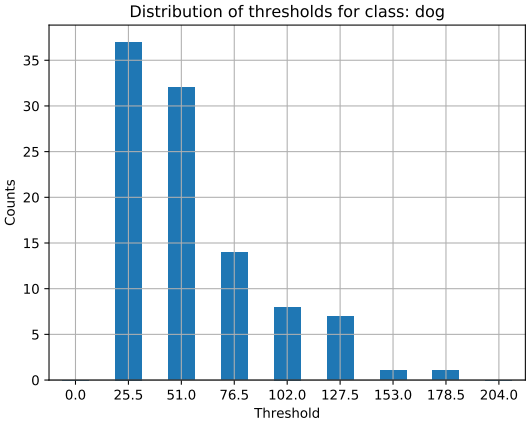
(c) Distribution of threshold for chair class



(d) Distribution of threshold for cow class



(e) Distribution of threshold for diningtable class



(f) Distribution of threshold for dog class

Figure 6.15: Distribution of thresholds taken at class level during validation which is used to select optimal threshold.

6.5 Summary

In this Section, summary of all the experiments are illustrated in Table 6.1. Results highlighted in green are top performance statistics across all the experiment setting. In Table 6.1, **-** indicates experiments that are unsuccessful and will be carried out in future work.

Method	Backbone	Dataset	Shape		Localization Metrics		Classification Metrics		
			Input	CAM	MaxBoxV2@ (0.3,0.5,0.7)	PxAP	Accuracy	Precision	Recall
CAM	VGG16	PASCAL VOC	256*256*3	16*16	**58.97**	**50.73**	72.7	82.85	78.1
CAM	VGG16	YCB	256*256*3	16*16	45.2	23.06	**99.76**	**99.99**	**100**
CAM	VGG16	@Work	256*256*3	16*16	24.94	12.8	92.86	96.43	97.32
CAM	ResNet18	PASCAL VOC	256*256*3	16*16	44.27	33.96	69.6	80.1	74.4
CAM	ResNet18	YCB	256*256*3	16*16	0.05	0.01	91.31	91.39	91.59
CAM	ResNet18	@Work	256*256*3	16*16	38.88	31.98	95.31	98.88	99.11
CAM	SqueezeNet1_1	PASCAL VOC	256*256*3	16*16	49.12	26.18	59.1	69.5	66.1
CAM	SqueezeNet1_1	YCB	256*256*3	16*16	29.83	0.03	95.68	95.91	96.91
CAM	SqueezeNet1_1	@Work	256*256*3	16*16	33.33	0.007	83.26	86.16	87.8
gradCAM	VGG16	PASCAL VOC	256*256*3	16*16	55.21	42.35	72.7	82.85	78.1
gradCAM	VGG16	YCB	256*256*3	16*16	54.19	46.09	99.76	99.99	100
gradCAM	VGG16	@Work	256*256*3	16*16	28.75	19.88	92.86	96.43	97.32
gradCAM	ResNet18	PASCAL VOC	256*256*3	16*16	45.25	34.83	69.6	80.1	74.4
gradCAM	ResNet18	YCB	256*256*3	16*16	0.04	0.01	91.31	91.39	91.59
gradCAM	ResNet18	@Work	256*256*3	16*16	37.04	42.87	95.31	98.88	99.11
gradCAM	SqueezeNet1_1	PASCAL VOC	256*256*3	16*16	-	-	59.1	69.5	66.1
gradCAM	SqueezeNet1_1	YCB	256*256*3	16*16	-	-	95.68	95.91	96.91
gradCAM	SqueezeNet1_1	@Work	256*256*3	16*16	-	-	85.04	88.61	90.63
CAM	VGG16	PASCAL VOC	256*256*3	32*32	39.21	27.25	58.82	68.65	66.19
CAM	VGG16	PASCAL VOC	256*256*3	8*8	46.01	38.04	68.4	77.97	75.7
CAM	VGG16	PASCAL VOC	512*512*3	32*32	54.27	40.52	72.99	83.13	77.32
CAM	VGG16	PASCAL VOC	128*128*3	8*8	52.44	49.21	66.76	76.45	74.34

Table 6.1: Summary of all the experiments conducted.

7

Conclusions

In computer vision, object localization is predicting the location of objects in the given image as well as identifying the object. Some of the applications of object localization include robot grasping[115], where the robot should be able to find the location of the object even before proceeding to grasp tasks. Another application in the field of robotics is obstacle avoidance systems[116] where robots should predict the location of obstacles. The availability of dataset with object-level annotations is a major drawback in achieving object localization models. The SOTA object detection methods like Faster-RCNN[48], You only look once (YOLO)[2] and Single shot detection (SSD)[1] require object-level annotations during training. Generating dataset with object-level annotations is a financially expensive and time-consuming task. This study focuses on finding and implementing object localization techniques which use very coarse annotations like image-level or no annotations. This research also includes the investigation done on available datasets and the analysis of dataset distribution. This analysis of dataset distribution helps to understand and defining rules when a custom dataset needs to be collected.

The techniques that adhere to the conditions like only image-level labels are used for localization of objects are known as Weakly supervised object localization (WSOL) techniques. WSOL techniques evaluated in this study are Class activation mapping (CAM)[76] and Gradient class activation mapping (Grad-CAM)[15]. These techniques are evaluated on three different datasets PASCAL VOC[8], Yale-CMU-Berkeley (YCB)[111] and RoboCup@Work datasets for different experiment setups. These experiments are set up to support the hypothesis assumed to answer research questions described in Section 1.3. The experiments conducted are briefed in Section 7.1 and outcome of these experiments are presented in Section 7.2. In Section 7.3, further work required to explore and understand the object localization techniques with only image-level labels better is given.

7.1 Contributions

The contributions of the research work are:

- **Dataset collection:** RoboCup@Work dataset is collected using Easy Augment[112] tool developed by myself which is a automatic data collection and augmentation tool. Annotations collected from Easy Augment[112] are inspected and cleaned (if required).
- **Literature:** A systematic literature review is carried out according to guidelines provided in [27]. The literature on Generic object proposal techniques, Unsupervised representation techniques, and Weakly supervised object localization techniques is presented.
- **Dataset analysis:** A analysis on selected datasets PASCAL VOC, YCB, and RoboCup@Work is done to show the measure of imbalance using Shannon entropy.
- **Illustration of localization metrics:** A detailed explanation of MaxBoxAcc[13] metric is done using images. The explanation of MaxBoxAcc using images will give better intuition of the metric.
- **Affect of activation map size:** A study is done to evaluate and compare the classification and localization performance of the model when different activation map sizes are produced at the last convolution layer of the model. Three different activation maps sizes examined are 32x32 pixels, 16x16 pixels, and 8x8 pixels generated using VGG16 architecture. This comparison study is done in a weakly supervised setting that is CAM and Grad-CAM WSOL techniques are evaluated on localization metrics.
- **Affect of network architecture:** Evaluated the performance of selected WSOL techniques on smaller network architectures ResNet18 and SqueezeNet1_1. This evaluation will help in understanding whether WSOL techniques can be deployed on less computationally powerful machines and whether RoboCup teams can use these techniques to bypass annotations.
- **Affect of dataset distribution:** This study is done in order to understand the role of dataset distribution on model classification and localization performance. A comparison is done between balanced YCB and RoboCup@Work datasets, and relatively imbalanced PASCAL VOC dataset.
- **Selection of optimal threshold:** A novel approach to select an optimal threshold for generating binary masks from generated class activation mapping is proposed. This selection process involves extracting the optimal threshold at two levels, the first one is getting the optimal threshold for the whole dataset from the distribution of the threshold observed during validation. The second approach suggests the selection of optimal threshold at the class level by observing the distribution of threshold during validation.

7.2 Lessons Learned

The following lessons are learned during the course of research:

- Convolutional neural networks have an explicit ability to localize objects despite being trained as a classification model that is trained on image-levels labels.
- Vanilla SqueezeNet1_1 architecture cannot be used for multi-label classification problem. It is solved by adding a fully connected layer after the global average pooling layer. Although the addition of a fully connected layer solved the training classification model, it increased the latency in the model.
- WSOL techniques CAM and Grad-CAM are not capable of localizing multiple occurrences of the same category object in a given image. Using CAM and Grad-CAM techniques multiple instance learning cannot be achieved in object localization tasks.
- YCB and RoboCup@Work datasets are balanced but only perform better in classification tasks but not in localization tasks. Whereas the PASCAL VOC dataset being relatively imbalanced performs better in localization tasks which is important in this study. This performance is observed as the background in YCB and RoboCup@Work datasets are the same throughout and there is very little variance between classes. In PASCAL VOC more difficult examples are present with varying backgrounds which forces the model to learn more significant parts of the object. This results in better localization performance using the PASCAL VOC dataset.
- Smaller network architectures will have a reduction in localization performance.
- Any alteration done is activation maps size will only result in decreasing the localization capability of convolutional neural network.
- There is a need for varying optimal threshold either across datasets or classes in order to better quantify the localization performance of WSOL techniques.

7.3 Future Work

The direction of possible future research is given in this section.

- A deeper investigation on why SqueezeNet1_1 architecture is failing to learn in multi-label classification tasks.
- Exploring the available WSOL techniques for multiple instance learning, where the model should be capable of localizing multiple instances of objects that belong to the same category in a weakly supervised setting.

- Evaluating Generalized gradient-based visual explanations for deep convolutional networks (Grad-CAM++)[82] against CAM and Grad-CAM methods for all the experiment settings.
- Evaluating the WSOL methods for measuring uncertainty in object localization performance.
- Work towards proposing a more concrete approach for selecting optimal threshold values for generating binary masks from class activation map.
- Generalize the hypothesis drawn from experiment "impact of activation map size" Currently it is evaluated only on VGG16 architecture. A concrete conclusion can be drawn if it is evaluated with some other network architectures.

A

Timeline of Weakly Supervised Object Localization

The most popular work in the field of weakly supervised object localization (WSOL) is presented in this chapter. Each table consists of a glance about the paper, proposed methodology or technique, metric used for evaluation, and SOTA in which they are evaluated. In summary *** indicates method cannot be explained orally without mathematical explanation. So these methods are not summarized in below tables.

Paper	Glance	Method	Proof	Evaluated Against
[13]	Argue that WSOL is ill-posed with only class level annotations and they come up with novel evaluation method where regular supervision is held over small set not matches with test set.	WSOL task as an image patch classification and demonstrate ill-posedness of WSOL	MaxBoxAcc: bounding box accuracies and PxAP: pixel average precision	CAM[76], HaS[77], ACoL[79], SPG[81], ADL[80], CutMix[78]
[83]	CAMs only highlight activations of class with highest probability. CCAM combines maps from highest to lowest probability linearly to suppress background regions.	Adapted the backbone network from[76] and linear combination of various activation maps is performed.	Classification acc, Localization acc and GT-Known localization accuracy	CAM[76], SPG[81], ACoL[79]
[75]	Divided WSOL into class-agnostic object localization and object classification. Proposed PSOL in which they generate psuedo annotations and perform bbox regression.	Psuedo supervised Object Localization (PSOL) which generate pseudo annotation i.e., noisy bbox as ground truth while training.	Top-1, Top-5 localization accuracy and GT-known localization accuracy	CAM[76], HaS[77], ACoL[79], SPG[81], ADL[80]

Table A.1: Summary of WSOL literature - 2020

Paper	**Glance**	**Method**	**Proof**	**Evaluated Against**
[78]	CutMix augmentation strategy: Authors stated that "patches are cut and pasted among training images where the ground truth labels are also mixed proportionally to the area of the patches."	***	Top-1 and Top-5 Localization accuracy percent	CAM[76], HaS[77], ACoL[79], SPG[81], ADL[80]
[80]	ADL based localization tries to hide the most significant regions in image and highlight informative region to improve WSOL accuracy.	ADL, a lightweight and powerful method based on self-attention mechanism is used to remove most discriminative part.	Top-1 classification and localization accuracy, GT-known localization acc,	CAM[76], HaS[77], ACoL[79], SPG[81]
[85]	Introduced novel instance filling approach. Pseudo labels are collected from noisy segment proposals. These pseudo supervision is used to predict class-agonist activation map.	This method is built on PRMs[94]. Then instance activation maps are generated from incomplete PRMs.	mean average precision and average best overlap	CAM[76], SPN[99], MELM[117]
[86]	Generally most discriminative object parts will conceal other parts of object. So to solve this Dual Attention Focused Module(DFM) is proposed. Two branches with information are fused to improve the object localization performance.	***	Top-1 classification and localization accuracy	CAM[76], HaS[77], ACoL[79], ADL[80]

Table A.2: Summary of WSOL literature - 2019 - 1

Paper	Glance	Method	Proof	Evaluated Against
[12]	Use internal relations between various CAMs and proposed novel intra-sample strategy, this restricts two CAMs of the sample. Developed a inter-sample criterion module to refine generated CAMs of each sample.	***	Top-1 classification and localization accuracy, GT-known localization acc	CAM[76], ACoL[79], SPG[81], STNet[118]
[87]	CAMs often display different weak response when image contains multiple instances of same class or small objects. This propose a varying scale discriminative region discovery method (DRDM) to find location of more objects. Gradient weights flowing into different conv layers are used to do this.	They firstly compute the gradient score maps by taking partial derivative of particular category confidence score to the activation maps from existing networks. Then proposed DRDM is applied to generate corresponding localization maps from different conv layers and these multi scale maps are fused to obtain final result.	Top-1 and Top-5 error, Localization acc	CAM[76], HaS[77], ACoL[79], SPG[81], Grad-CAM[15], Grad-CAM++[82]
[89]	Novel end-to-end model to enhance CAMs which result in more accurate localization of targeted objects. To achieve this additional module is added into traditional classification network and extract relative foreground regions from background.	***	Top-1 and Top-5 localization error	ACoL[79], SPG[81]

Table A.3: Summary of WSOL literature - 2019 - 2

Paper	**Glance**	**Method**	**Proof**	**Evaluated Against**
[90]	Object localization relies on different features than classification feature learnt by networks.They propose a convolutional, multi-scale spatial localization network that provides accurate localization for the object of interest.	Instead of using the classic CAMs, we show that using A conv space transformer can lead to high performance returning the bbox of the object, similar to in fully supervised approaches.	Top-1 classification and localization accuracy	CAM[76], HaS[77], ACoL[79], SPG[81]
[91]	Typical CAMs are compressed to large false positive regions. So this propose a novel techniqe for WSOL, consists of a localizer and a classifier, where the localizer is restricted to find relevant and irrelevant regions using entropy with the goal to decrease false positive areas.	They considered modeling the uncertainty of the model prediction over positive, or negative regions using conditional entropy.	Image level classification error and Pixel level localization error	CAM[76], Grad-CAM[15], Wild-CAT[24], Deep-MIL[119]
[93]	Two fold approach for object localization. 1) model is encouraged to detect foreground objects using salient object detection module. 2) Perceptual triplet loss enhances the foreground object detection capability.	This composed of four subnetworks, including an encoder for feature extraction, a classifier for performing image classification, a decoder for detecting salient objects, and an ImageNet pretrained network for enhancing the capability of salient object detection.	Top-1 classification and localization accuracy, GT-known localization acc	CAM[76], HaS[77]

Table A.4: Summary of WSOL literature - 2019 - 3

Paper	Glance	Method	Proof	Evaluated Against
[79]	Adversarial Complementary Learning(ACoL) proves that objects can be localized using category specific activation maps of end conv layer.	Requires long description.	Top-1 and Top-5 localization and classification error. GT-known loc error	CAM[76], HaS[77], Feedback[120], MWP[98]
[94]	Using class peak responses to extract instance masks from classification network. pekas in class response maps are back propagated and mapped to highly informative regions in image. So generated maps are called Peak Response Maps(PRMs).	First class peak responses are generated using imaged cues and passed to back propagation to further improve the peaks for each instance.	mAP, mIOU, ABO	CAM[76], , SPN[99], MELM[117]WildCAT[24], DeepMIL[119], WSLoc[121]
[95]	According to authors [95] "Count-guided weakly supervised localization(C-WSL), an approach that uses per-class object count as a new form of supervision to improve weakly supervised localization (WSL)"	High quality positive regions are selected from given object proposals based on count specified per class. These picked proposals are used to train WSL detector. This decreases predicted boundaries that are not likely and contain more than two object instances.	CorLoc Correct Localization and mAP mean Average Precision	WSDDN[122], multifold[123], WSCC[124], self-taught[125], Contextlocnet[101]

Table A.5: Summary of WSOL literature - 2018 - 1

Paper	**Glance**	**Method**	**Proof**	**Evaluated Against**
[97]	Introduced fine grained object retrieval method to address issues due to coarsely extracted features at image level and issues due to triplet loss. So they use centralized ranking loss and weakly supervised attractive feature extraction.	***	recall@K and IoU	CAM[76], part-selection[126]
[98]	Excitation Backprop is a technique to go through top-down signals downwards in the network architecture through a probability based Winner-Take-All process. Introduced contrastive attention to make the top-down attention maps more discriminative.	***	localization error rate	CAM[76], Feedback[120]
[82]	Provide better network visualization than Grad-CAM and improve the localization of objects and finding all occurrences of multiple objects that belong to same category.	Reformulated the structure of weights in Grad-CAM look paper for mathematical explanation.	Average drop %, % increase in confidence, Win %	Grad-CAM[15]

Table A.6: Summary of WSOL literature - 2018 - 2

Paper	Glance	Method	Proof	Evaluated Against
[77]	Hide some patches in training images randomly and force network to learn other less important parts when most discriminative parts randomly hidden.	During training phase each input image is divided into equal grids and each patch is hidden randomly with a probability and train a classification network. During testing input is given without any hidden patches.	Top-1 loc and classification accuracy, GT-known loc	CAM[76]
[99]	Implementation of WSOL using CNNs as a full pipeline in an end-to-end learning manner. To achieve this Soft Proposal layer is introduced into CNNs architecture to enable cost free object proposals.	Design and integrating Soft Proposal layer into classical architectures, explained in Section 3 of[99]	CorLoc, mAP, localization accuracy	WSDDN[122], multi-fold[123], CAm[76], Context-Loc[101]
[24]	A new, weakly supervised learning dedicated to learning discriminating localized visual features by using only image-level captions during training.	***	mAP of localization	WSLocalization[121], Deep-MIL[102], ProNet[127]
[100]	Training consists of two stages, First is conventional CNNs to find discriminative parts and second stage suppress the output from first stage by inference conditional feedback.	The first Network (with solid layers, colored gray) records an input image and issues heat cards. Only the heat maps match The classes in the captions are selected and become a suppression mask after applying the threshold.	IoU	No comparison for localization task

Table A.7: Summary of WSOL literature - 2017 - 1

B

Dataset Collection Tool

As mentioned in Chapter 4, RoboCup@Work dataset is collected using Easy Augment [112] tool. In this chapter, a brief description of the GUI based data collection tool is presented. The sample GUI of the tool is shown in Figure B.1 which includes the main window, capture window, and artificial image generator.

B.1 Architecture

The architecture of Easy Augment [112] is given in Figure B.2. It consists of three parts in Figure B.2, from the left-hand side first part, is for capturing image data. Followed by artificial image generation which is used for generating augmented data and the Final part consists of handling labelme [128] annotation and generating augmented data from labelme annotated files.

B.2 Requirements

To use the tool following are the minimum system requirements:

- Ubuntu 16.04 (18.04 testing)
- Intel RealSense Camera
- Processor Intel i5 or higher
- Python 3.5
- Installed pcl 1.7 library

Dependencies:

- GUI: PyQt5
- Data Processing: OpenCV, pcl-, Scipy, Numpy, pascal-voc-tools, pascal-voc-writer
- Hardware: PyRealSense2

- Testing: pytest-qt
- Others: tqdm, matplotlib, imutils, joblib

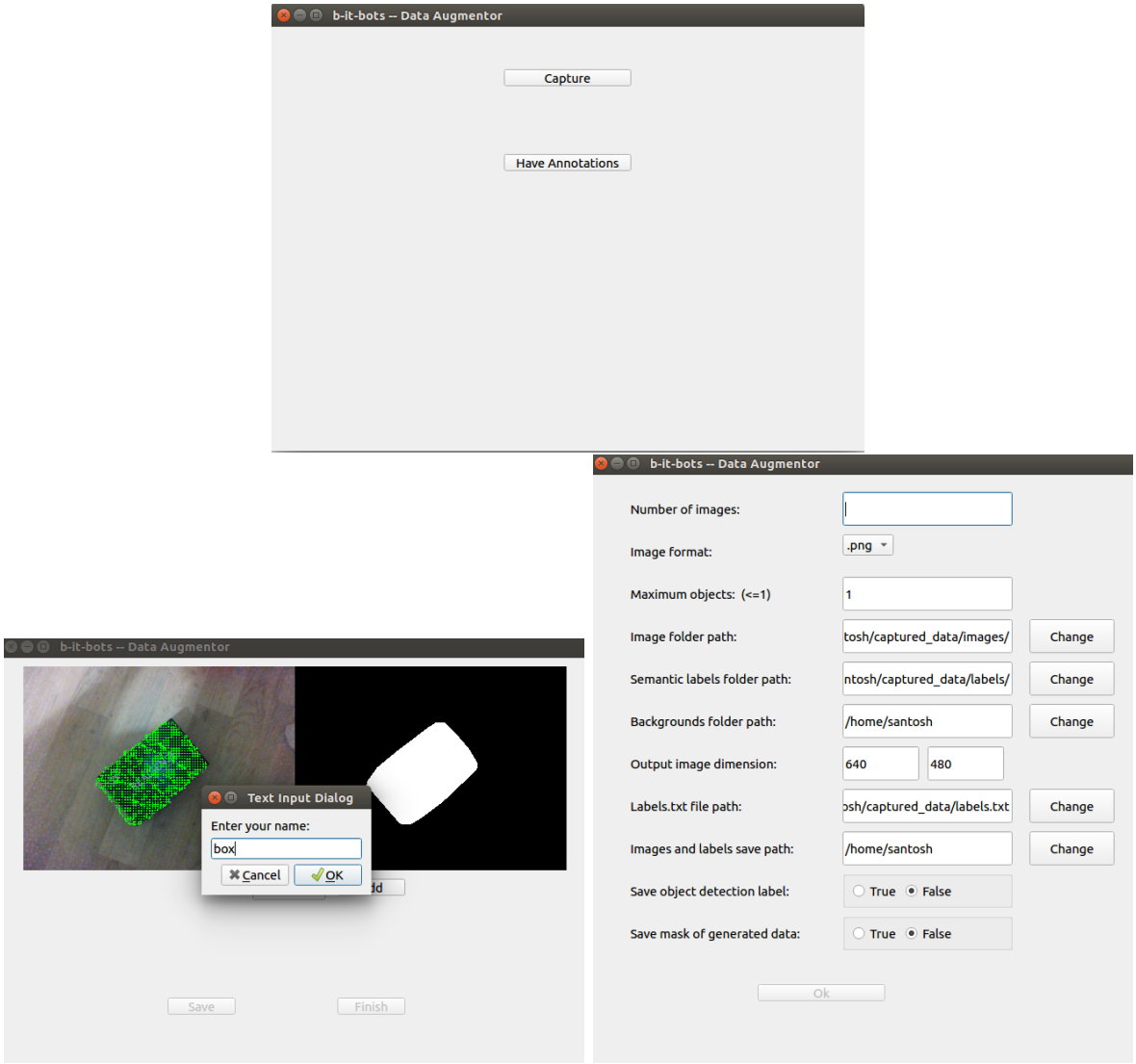


Figure B.1: Sample GUI windows from left to right: Main window, Capture window and Artificial image generator window

B.3 Capabilities

The tool is capable of performing following tasks:

- Live feed of RGB images and corresponding mask.

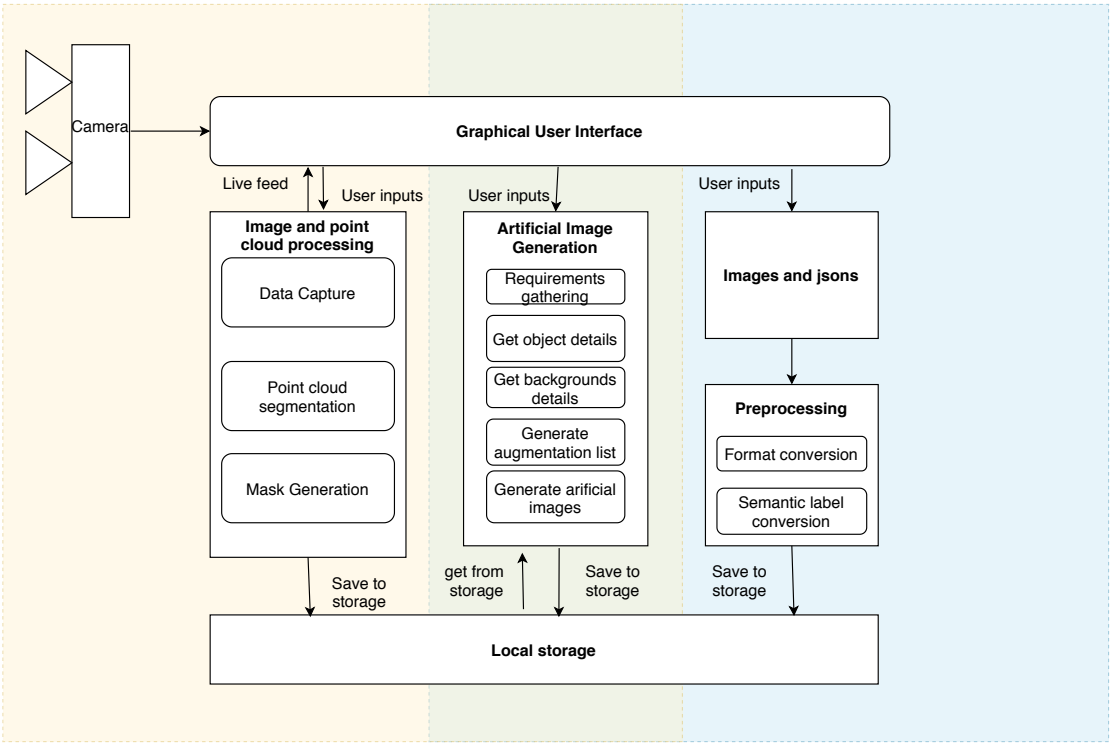


Figure B.2: Architecture of the developed project, left: handles capturing part and right: handles labeleme annotations

- Save Image and corresponding semantic label.
- Generate artificial data from captured images.
- Generate artificial data from labelme annotated images.
- Control shape of artificial generated data.
- Types of labels generated:
  - Semantic label with class information.
  - Pascal VOC semantic label.
  - Bounding box in xml file.
- Package delivered as pip installable.

B.4 Development Status

Current development status and limitations are presented below:

Installation and Usage:

Following are the steps to setup the tool:

- pip3 install easy-augment
- "easy-augment" command launches the tool.

Limitations:

This tools comes with few limitations and they are:

- Only RealSense camera can be used.
- Number of classes captured should be more than or equal to two.
- Only one object per scene is allowed.

Bibliography

[1] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. "Ssd: Single shot multibox detector". In: Springer. 2016, pp. 21–37.

[2] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[4] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 580–587. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf.

[5] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. "Bottom-up and top-down attention for image captioning and visual question answering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6077–6086.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2 (June 2010), pp. 303–338.

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 3213–3223. DOI: 10.1109/CVPR.2016.350.

[10] "Big challenge in Deep Learning: training data". In: *Hackernoon* (2017). URL: https://bit.ly/36zbyRm (visited on 12/16/2019).

[11] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. "What's the point: Semantic segmentation with point supervision". In: *European conference on computer vision*. Springer. 2016, pp. 549–565.

[12] Guofeng Cui, Ziyi Kou, Shaojie Wang, Wentian Zhao, and Chenliang Xu. "Weakly Supervised Object Localization with Inter-Intra Regulated CAMs". In: *arXiv preprint arXiv:1911.07160* (2019).

[13] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. "Evaluating Weakly Supervised Object Localization Methods Right". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. to appear. 2020. published.

[14] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size". In: *arXiv:1602.07360* (2016).

[15] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[16] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1409.1556.

[17] 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 (CVPR)*. June 2016.

[18] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. "The YCB object and Model set: Towards common benchmarks for manipulation research". In: *2015 International Conference on Advanced Robotics (ICAR)*. July 2015, pp. 510–517. DOI: 10.1109/ICAR.2015.7251504.

[19] Justin Gong. *Supervised Learning*. 2018. URL: https://www.gong-jj.com/sl/ (visited on 03/03/2020).

[20] Justin Gong. *Unsupervised Learning*. 2018. URL: https://www.gong-jj.com/ul/ (visited on 03/03/2020).

[21] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]". In: *IEEE Transactions on Neural Networks* 20.3 (2009), pp. 542–542.

[22] Fabien Lotte. "Signal Processing Approaches to Minimize or Suppress Calibration Time in Oscillatory Activity-Based Brain–Computer Interfaces". In: *Proceedings of the IEEE* 103 (June 2015), pp. 871–890. DOI: 10.1109/JPROC.2015.2404941.

[23] Amit Chaudhary. "The Illustrated Self-Supervised Learning)[book reviews]". In: 3 (24-Feb-2020), pp. 542–542. URL: https://amitness.com/2020/02/illustrated-self-supervised-learning/.

[24] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. "Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 642–651.

[25] Hakan Bilen. *Weakly supervised object detection (WSOD)*. 2018. URL: https://hbilen.github.io/wsl-cvpr18.github.io/assets/wsod.pdf.

[26] Jason Brownlee. *4 Types of Classification Tasks in Machine Learning*. 2020.

[27] Barbara Kitchenham and Stuart Charters. "Guidelines for performing systematic literature reviews in software engineering". In: (2007).

[28] B. Alexe, T. Deselaers, and V. Ferrari. "Measuring the Objectness of Image Windows". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (Nov. 2012), pp. 2189–2202. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2012.28.

[29] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. "What is an object?" In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 73–80.

[30] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. "BING: Binarized Normed Gradients for Objectness Estimation at 300fps". In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*. CVPR '14. USA: IEEE Computer Society, 2014, pp. 3286–3293. ISBN: 9781479951185. DOI: 10.1109/CVPR.2014.414. URL: https://doi.org/10.1109/CVPR.2014.414.

[31] Ziming Zhang, Yun Liu, Tolga Bolukbasi, Ming-Ming Cheng, and Venkatesh Saligrama. "BING++: A Fast High Quality Object Proposal Generator at 100fps". In: *Computing Research Repository (CoRR)* abs/1511.04511 (2015). arXiv: 1511.04511. URL: http://arxiv.org/abs/1511.04511.

[32] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104.2 (Sept. 2013), pp. 154–171. ISSN: 1573-1405. DOI: 10.1007/s11263-013-0620-5. URL: https://doi.org/10.1007/s11263-013-0620-5.

[33] J. Carreira and C. Sminchisescu. "CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.7 (2012), pp. 1312–1328.

[34] Ian Endres and Derek Hoiem. "Category-independent object proposals with diverse ranking". In: *IEEE transactions on pattern analysis and machine intelligence* 36.2 (2013), pp. 222–234.

[35] Santiago Manen, Matthieu Guillaumin, and Luc Van Gool. "Prime object proposals with randomized prim's algorithm". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 2536–2543.

[36] Pekka Rantalankila, Juho Kannala, and Esa Rahtu. "Generating object segmentation proposals using global and local search". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2417–2424.

[37] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. "Multiscale combinatorial grouping". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 328–335.

[38] Philipp Krähenbühl and Vladlen Koltun. "Geodesic object proposals". In: *European conference on computer vision*. Springer. 2014, pp. 725–739.

[39] C Lawrence Zitnick and Piotr Dollár. "Edge boxes: Locating object proposals from edges". In: *European conference on computer vision*. Springer. 2014, pp. 391–405.

[40] Esa Rahtu, Juho Kannala, and Matthew Blaschko. "Learning a category independent object detection cascade". In: *2011 international conference on Computer Vision*. IEEE. 2011, pp. 1052–1059.

[41] Ziming Zhang, Jonathan Warrell, and Philip HS Torr. "Proposal generation for object detection using cascaded ranking svms". In: *CVPR 2011*. IEEE. 2011, pp. 1497–1504.

[42] Piotr Dollár and C Lawrence Zitnick. "Structured forests for fast edge detection". In: *Proceedings of the IEEE international conference on computer vision*. 2013, pp. 1841–1848.

[43] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks". In: *arXiv preprint arXiv:1312.6229* (2013).

[44] Nikolaos Karianakis, Thomas J Fuchs, and Stefano Soatto. "Boosting convolutional features for robust object proposals". In: *arXiv preprint arXiv:1503.06350* (2015).

[45] Pedro OO Pinheiro, Ronan Collobert, and Piotr Dollár. "Learning to segment object candidates". In: *Advances in Neural Information Processing Systems*. 2015, pp. 1990–1998.

[46] Weicheng Kuo, Bharath Hariharan, and Jitendra Malik. "Deepbox: Learning objectness with convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2479–2487.

[47] Amir Ghodrati, Ali Diba, Marco Pedersoli, Tinne Tuytelaars, and Luc Van Gool. "Deepproposal: Hunting objects by cascading deep convolutional layers". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 2578–2586.

[48] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[49] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. "Hypernet: Towards accurate region proposal generation and joint object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 845–853.

[50] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. "Transforming auto-encoders". In: *International Conference on Artificial Neural Networks*. Springer. 2011, pp. 44–51. URL: https://link.springer.com/chapter/10.1007/978-3-642-21735-7_6.

[51] Taco Cohen and Max Welling. "Group Equivariant Convolutional Networks". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2990–2999. URL: http://proceedings.mlr.press/v48/cohenc16.html.

[52] Taco S. Cohen and Max Welling. "Steerable CNNs". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: https://openreview.net/forum?id=rJQKYt5ll.

[53] Jan Eric Lenssen, Matthias Fey, and Pascal Libuschewski. "Group Equivariant Capsule Networks". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., 2018, pp. 8844–8853. URL: http://papers.nips.cc/paper/8100-group-equivariant-capsule-networks.pdf.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

[55] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. "Adversarial Feature Learning". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: https://openreview.net/forum?id=BJtNZAFgg.

[56] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martın Arjovsky, Olivier Mastropietro, and Aaron C. Courville. "Adversarially Learned Inference". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017. URL: https://openreview.net/forum?id=B1ElR4cgg.

[57] Huaibo Huang, zhihang li zhihang, Ran He, Zhenan Sun, and Tieniu Tan. "IntroVAE: Introspective Variational Autoencoders for Photographic Image Synthesis". In: *Advances in Neural Information Processing Systems 31*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Curran Associates, Inc., 2018, pp. 52–63. URL: http://papers.nips.cc/paper/7291-introvae-introspective-variational-autoencoders-for-photographic-image-synthesis.pdf.

[58] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U. Gutmann, and Charles Sutton. "VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 3308–3318. URL: http://papers.nips.cc/paper/6923-veegan-reducing-mode-collapse-in-gans-using-implicit-variational-learning.pdf.

[59] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014. URL: http://arxiv.org/abs/1312.6114.

[60] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and Composing Robust Features with Denoising Autoencoders". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland:

Association for Computing Machinery, 2008, pp. 1096–1103. ISBN: 9781605582054. DOI: 10.1145/1390156.1390294. URL: https://doi.org/10.1145/1390156.1390294.

[61] Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. "Contractive Auto-Encoders: Explicit Invariance during Feature Extraction". In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. Bellevue, Washington, USA: Omnipress, 2011, pp. 833–840. ISBN: 9781450306195.

[62] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel Recurrent Neural Networks". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1747–1756. URL: http://proceedings.mlr.press/v48/oord16.html.

[63] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu koray, Oriol Vinyals, and Alex Graves. "Conditional Image Generation with PixelCNN Decoders". In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, Inc., 2016, pp. 4790–4798. URL: http://papers.nips.cc/paper/6527-conditional-image-generation-with-pixelcnn-decoders.pdf.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., 2017, pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[65] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. "Unsupervised Visual Representation Learning by Context Prediction". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. ICCV '15. USA: IEEE Computer Society, 2015, pp. 1422–1430. ISBN: 9781467383912. DOI: 10.1109/ICCV.2015.167. URL: https://doi.org/10.1109/ICCV.2015.167.

[66] Richard Zhang, Phillip Isola, and Alexei A. Efros. "Colorful Image Colorization". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 649–666. ISBN: 978-3-319-46487-9. DOI: https://doi.org/10.1007/978-3-319-46487-9_40.

[67] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. "Discriminative Unsupervised Feature Learning with Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 766–774. URL: http://papers.nips.cc/paper/5548-discriminative-unsupervised-feature-learning-with-convolutional-neural-networks.pdf.

[68] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. "Deep Clustering for Unsupervised Learning of Visual Features". In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Cham: Springer International Publishing, 2018, pp. 139–156. ISBN: 978-3-030-01264-9. DOI: https://doi.org/10.1007/978-3-030-01264-9_9.

[69] George Papandreou, Liang-Chieh Chen, Kevin P. Murphy, and Alan L. Yuille. "Weakly- and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.

[70] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. "What's the Point: Semantic Segmentation with Point Supervision". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 549–565. ISBN: 978-3-319-46478-7.

[71] Jifeng Dai, Kaiming He, and Jian Sun. "BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015.

[72] Dim P. Papadopoulos, Alasdair D. F. Clarke, Frank Keller, and Vittorio Ferrari. "Training Object Class Detectors from Eye Tracking Data". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 361–376. ISBN: 978-3-319-10602-1.

[73] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. "ScribbleSup: Scribble-Supervised Convolutional Networks for Semantic Segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[74] Seunghoon Hong, Hyeonwoo Noh, and Bohyung Han. "Decoupled Deep Neural Network for Semi-supervised Semantic Segmentation". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 1495–1503. URL: http://papers.nips.

cc/paper/5858-decoupled-deep-neural-network-for-semi-supervised-semantic-segmentation.pdf.

[75] Chen-Lin Zhang, Yun-Hao Cao, and Jianxin Wu. "Rethinking the Route Towards Weakly Supervised Object Localization". In: *arXiv preprint arXiv:2002.11359* (2020).

[76] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[77] Krishna Kumar Singh and Yong Jae Lee. "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization". In: *2017 IEEE international conference on computer vision (ICCV)*. IEEE. 2017, pp. 3544–3553.

[78] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features". In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019.

[79] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S. Huang. "Adversarial Complementary Learning for Weakly Supervised Object Localization". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[80] Junsuk Choe and Hyunjung Shim. "Attention-Based Dropout Layer for Weakly Supervised Object Localization". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[81] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. "Self-produced Guidance for Weakly-supervised Object Localization". In: *European Conference on Computer Vision*. Springer. 2018.

[82] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 839–847.

[83] Seunghan Yang, Yoonhyung Kim, Youngeun Kim, and Changick Kim. "Combinational Class Activation Maps for Weakly Supervised Object Localization". In: *The IEEE Winter Conference on Applications of Computer Vision (WACV)*. Mar. 2020.

[84] Haolan Xue, Chang Liu, Fang Wan, Jianbin Jiao, Xiangyang Ji, and Qixiang Ye. "DANet: Divergent Activation for Weakly Supervised Object Localization". In: *The IEEE International Conference on Computer Vision (ICCV)*. Oct. 2019.

[85] Yi Zhu, Yanzhao Zhou, Huijuan Xu, Qixiang Ye, David Doermann, and Jianbin Jiao. "Learning Instance Activation Maps for Weakly Supervised Instance Segmentation". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[86] Yukun Zhou, Zailiang Chen, Hailan Shen, Qing Liu, Rongchang Zhao, and Yixiong Liang. "Dual-attention Focused Module for Weakly Supervised Object Localization". In: *arXiv preprint arXiv:1909.04813* (2019).

[87] Pei Lv, Haiyu Yu, Junxiao Xue, Junjin Cheng, Lisha Cui, Bing Zhou, Mingliang Xu, and Yi Yang. "Multi-scale discriminative Region Discovery for Weakly-Supervised Object Localization". In: *arXiv preprint arXiv:1909.10698* (2019).

[88] K. Huang, F. Meng, H. Li, S. Chen, Q. Wu, and K. N. Ngan. "Class Activation Map Generation by Multiple Level Class Grouping and Orthogonal Constraint". In: *2019 Digital Image Computing: Techniques and Applications (DICTA)*. Dec. 2019, pp. 1–6. DOI: 10.1109/DICTA47822.2019.8946068.

[89] Ziyi Kou, Wentian ZhaoGuofeng Cui, and Shaojie Wang. "Weakly Supervised Localization Using Background Images". In: *arXiv preprint arXiv:1909.03619* (2019).

[90] Akhil Meethal, Marco Pedersoli, Soufiane Belharbi, and Eric Granger. "Convolutional STN for Weakly Supervised Object Localization and Beyond". In: *arXiv preprint arXiv:1912.01522* (2019).

[91] S. Belharbi, J. Rony, J. Dolz, I. Ben Ayed, L. McCaffrey, and E. Granger. "Min-max Entropy for Weakly Supervised Pointwise Localization". In: *coRR* abs/1907.12934 (2019).

[92] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. "C-MIL: Continuation Multiple Instance Learning for Weakly Supervised Object Detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[93] Y. Chen and W. H. Hsu. "Saliency Aware: Weakly Supervised Object Localization". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2019, pp. 1907–1911. DOI: 10.1109/ICASSP.2019.8682756.

[94] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. "Weakly supervised instance segmentation using class peak response". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 3791–3800.

[95] Mingfei Gao, Ang Li, Ruichi Yu, Vlad I Morariu, and Larry S Davis. "C-wsl: Count-guided weakly supervised localization". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 152–168.

[96] Jiajie Wang, Jiangchao Yao, Ya Zhang, and Rui Zhang. "Collaborative Learning for Weakly Supervised Object Detection". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 971–977. DOI: 10.24963/ijcai.2018/135. URL: https://doi.org/10.24963/ijcai.2018/135.

[97] Xiawu Zheng, Rongrong Ji, Xiaoshuai Sun, Yongjian Wu, Feiyue Huang, and Yanhua Yang. "Centralized Ranking Loss with Weakly Supervised Localization for Fine-Grained Object Retrieval". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, July 2018, pp. 1226–1233. DOI: 10.24963/ijcai.2018/171. URL: https://doi.org/10.24963/ijcai.2018/171.

[98] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. "Top-down neural attention by excitation backprop". In: *International Journal of Computer Vision* 126.10 (2018), pp. 1084–1102.

[99] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. "Soft proposal networks for weakly supervised object localization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1841–1850.

[100] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. "Two-phase learning for weakly supervised object localization". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 3534–3543.

[101] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. "Contextlocnet: Context-aware deep network models for weakly supervised localization". In: *European Conference on Computer Vision*. Springer. 2016, pp. 350–365.

[102] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. "Is object localization for free? - Weakly-supervised learning with convolutional neural networks". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 685–694. DOI: 10.1109/CVPR.2015.7298668.

[103] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft COCO: Common Objects in Context". In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.

[104] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009.

[105] Adam Coates, Andrew Ng, and Honglak Lee. "An analysis of single-layer networks in unsupervised feature learning". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 2011, pp. 215–223.

[106] Ya Le and Xuan Yang. "Tiny imagenet visual recognition challenge". In: ().

[107] Alina Kuznetsova, Mohamad Hassan Mohamad Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale". In: (2020).

[108] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. "Learning deep features for scene recognition using places database". In: *Advances in neural information processing systems*. 2014, pp. 487–495.

[109] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. *Caltech-UCSD Birds 200*. Tech. rep. CNS-TR-2010-001. California Institute of Technology, 2010.

[110] Alex Krizhevsky. "Learning Multiple Layers of Features from Tiny Images". In: (2009).

[111] Berk Calli, Arjun Singh, James Bruce, Aaron Walsman, Kurt Konolige, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. "Yale-CMU-Berkeley dataset for robotic manipulation research". In: *The International Journal of Robotics Research* 36.3 (2017), pp. 261–268. DOI: 10.1177/0278364917700714. eprint: https://doi.org/10.1177/0278364917700714. URL: https://doi.org/10.1177/0278364917700714.

[112] Muthireddy Santosh. *Easy Augment*. https://github.com/santoshreddy254/easy_augment. 2019.

[113] Simone (https://stats.stackexchange.com/users/2719/simone). *a general measure of dataset imbalance*. Cross Validated. URL:https://stats.stackexchange.com/q/239982 (version: 2016-10-13). eprint: https://stats.stackexchange.com/q/239982. URL: https://stats.stackexchange.com/q/239982.

[114] Jesse Read. *Multi-label Classification*. URL: https://users.ics.aalto.fi/jesse/talks/Multilabel-Part01.pdf. July 2013.

[115] E. Kefalea. "Object localization and recognition for a grasping robot". In: *IECON '98. Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (Cat. No.98CH36200)*. Vol. 4. 1998, 2057–2062 vol.4.

[116] Hema C.R, Paulraj M.P, A. H. B. Adom, K. F. Sim, and R. Palaniappan. "An intelligent vision system for object localization and obstacle avoidance for an indoor service robot". In: *2011 IEEE Student Conference on Research and Development*. 2011, pp. 117–122.

[117] Fang Wan, Pengxu Wei, Jianbin Jiao, Zhenjun Han, and Qixiang Ye. "Min-entropy latent model for weakly supervised object detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1297–1306.

[118] Mahdi Biparva and John Tsotsos. "STNet: Selective tuning of convolutional networks for object localization". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, pp. 2715–2723.

[119] Maximilian Ilse, Jakub Tomczak, and Max Welling. "Attention-based Deep Multiple Instance Learning". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, Oct. 2018, pp. 2127–2136. URL: `http://proceedings.mlr.press/v80/ilse18a.html`.

[120] Chunshui Cao, Xianming Liu, Yi Yang, Yinan Yu, Jiang Wang, Zilei Wang, Yongzhen Huang, Liang Wang, Chang Huang, Wei Xu, et al. "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 2956–2964.

[121] Archith John Bency, Heesung Kwon, Hyungtae Lee, S Karthikeyan, and BS Manjunath. "Weakly supervised localization using deep feature maps". In: *European Conference on Computer Vision*. Springer. 2016, pp. 714–731.

[122] Hakan Bilen and Andrea Vedaldi. "Weakly supervised deep detection networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2846–2854.

[123] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. "Weakly supervised object localization with multi-fold multiple instance learning". In: *IEEE transactions on pattern analysis and machine intelligence* 39.1 (2017), pp. 189–203.

[124] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. "Weakly supervised cascaded convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 914–922.

[125] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. "Deep self-taught learning for weakly supervised object localization". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1377–1385.

[126] Xiangteng He and Yuxin Peng. "Weakly supervised learning of part selection model with spatial constraints for fine-grained image classification". In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[127] Chen Sun, Manohar Paluri, Ronan Collobert, Ram Nevatia, and Lubomir Bourdev. "ProNet: Learning to Propose Object-Specific Boxes for Cascaded Neural Networks". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.

[128] Kentaro Wada. *labelme: Image Polygonal Annotation with Python*. `https://github.com/wkentaro/labelme`. 2016.