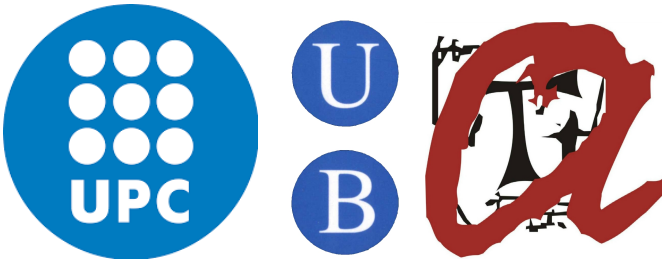
Master in Artificial Intelligence



UNIVERSITAT POLITÈCNICA DE CATALUNYA (UPC) – BarcelonaTech
UNIVERSITAT DE BARCELONA (UB)
UNIVERSITAT ROVIRA i VIRGILI (URV)

Master Thesis

Improving object detection by exploiting semantic relations between objects

Director: Mariella Dimiccoli
Co-director: Petia Radeva

FACULTAT DE MATEMÀTIQUES I INFORMÀTICA (UB)

Abstract

Object detection is a fundamental and challenging problem in computer vision. Detecting the objects visible in an image can give us a good understanding and description of the image. The extracted information can later be used to improve the results of other computer vision tasks like activity recognition, content-based image retrieval, scene recognition and more.

As technology and internet connection are becoming more accessible, billions of people upload photos and videos every day. In order to make use of this enormous amount of data we need to be able to extract information from these images in a quick and yet reliable way. Convolutional neural networks (CNN) have made possible enormous progresses in object detection and classification in recent years and have already established themself as the state of the art approach for these problems. In this work, we try to improve object detection performances by employing a CNN approach able to exploit object co-occurrences in natural images. Typically, real world scenes often exhibit a coherent composition of object in terms of co-occurrence probability. For instance, in a restaurant we typically see dishes, bottles and glasses. We aim at using this type of knowledge as a cue for disambiguating object labels in a detection task.

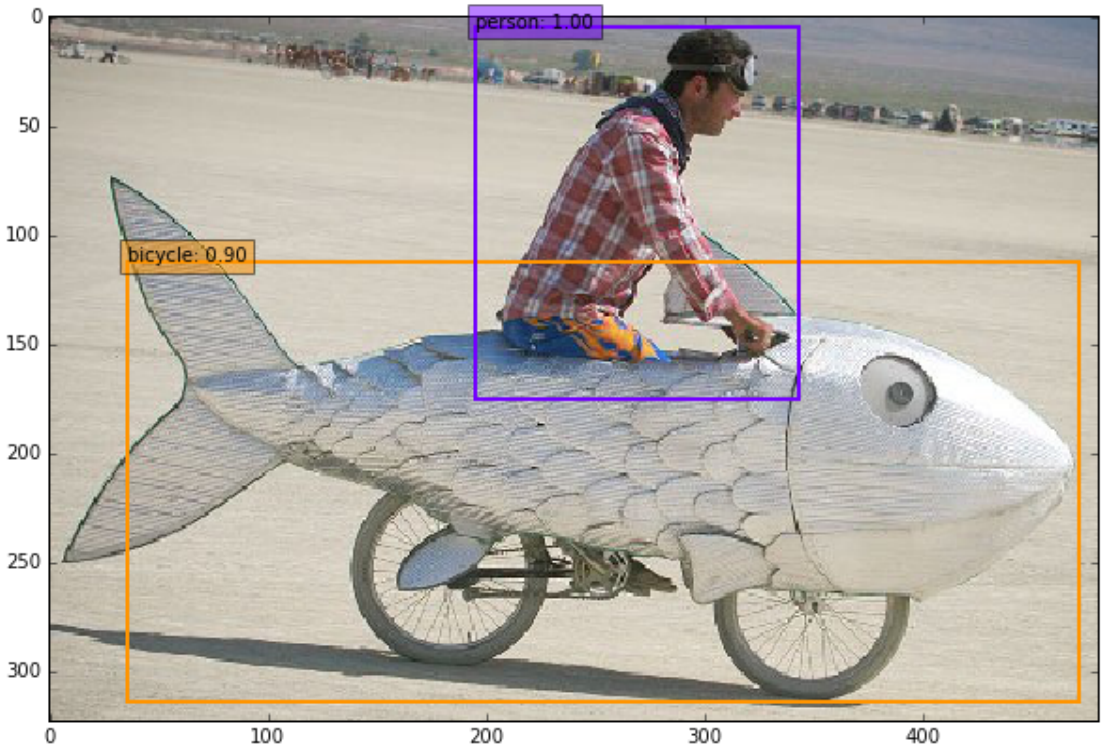
Contents

1 Introduction

One of the most investigated and challenging problems in computer vision is object detection. It is the problem of automatically detecting the objects that are present on a visual scene. A more formal definition states “given an image, an object detection algorithm returns all the instances of one or more type of objects in form of bounding boxes that tightly enclose them”. [1] As we can see from these definitions numerous tasks are involved in object detection including recognizing what objects are present and localizing these objects. An object detection algorithms receive as input image and produce as output a set of objects represented by labels with their confidence scores and bounding boxes **Fig. 1**. The confidence score specifies to what extend the model is confident that the object in the bounding boxes corresponds to the label, usually expressed as a number in the interval [0,1]. Obtaining information about the present objects and their location plays a vital role in the understanding of visual scenes. [3]

Object detection is an old and fundamental problem in computer vision. Despite the great progress made in the recent years it still remains an active area of research. One direction of ongoing research is the improvement of performance which is still lacking behind when taking human performance as a reference. The task is not a difficult one for humans. We can easily identify objects and their position in an image within milliseconds. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Algorithms for object detection with fast and accurate performance can allow computers to do more complex tasks without the need of specialized sensors. A particularly popular and challenging application example are self driving cars based only on image processing or even general purpose robotic systems. [4] [5]

Input image



Detected objects

Fig. 1

In the early days of computer vision the focus of research was on building models based on feature extraction tailored for the specific domain of application. Creating a good model required serious amount of data preprocessing, examining the specific characteristics of the target domain to extract the best set of features. Even though progress was made in the area, general purpose models did not produce satisfying results. All of this changed with the introduction of Convolutional Neural Networks. Current state of the art solutions for object detection are based on the use Convolutional Neural Networks (CNN).

The problem of object detection is more complex than that of object recognition. The current trend in this field of computer vision is to reduce object detection to a series of object recognition problems. One approach for reducing the detection task to an object recognition task was proposed by Girshick et al. In his work object detection is done by using exhaustive sliding-window detector. By using a window that slides through the image horizontally and vertically we can try to recognize an object in the current window location. Ideally the recognized objects with a confidence score above a given threshold can then be used as an end result of the algorithm's work. [6][7]

But the object detection task is even more complex as it requires accurate localization of the objects. This raises two main issues. First is overlapping object detections. How do we know which of the overlapping detections actually point to the same object? And second - which localization is the best of all proposed? Additional concerns arise when we take into account the complexity and performance of an algorithm that tries to deal with the overlapping detections. A faster approach that tries to solve these issues described above is proposed in another work by Girshick et al. The approach is called Faster Region-based Convolutional Neural Networks (Fast R-CNN). This method build on previous work to efficiently classify object proposals using CNNs, while several innovations have been utilized to increase both training and testing performance as well as accuracy of the model. The idea of the Fast R-CNN is to start with over-segmentation. Then over iterations similar regions are merged together until satisfying region object proposals are produced. In the

end object recognition task is performed on the candidate region object proposals.

Current state of the art solutions for object detection are based on the use Convolutional Neural Networks (CNN).

As we can deduct from the proposed approaches a typical solution is to apply a CNN for object recognition to each image region estimated to be an object by a so called *object proposal* algorithm. However, since the object recognizer is applied to each object proposal separately, its output does not take into account the occurrence of other objects in the scene. We can compare the current work object detection algorithms to a Naive Bayes Classifier as they do not take into account the relations between different objects in the same visual scene. Several studies suggest that extracting semantic relations between objects and using them in classification tasks can improve the accuracy of the models by some margin [8] [11]. However, such experiments have never been conducted in a Convolutional Neural Network framework.

In this paper we propose a novel approach that extract semantic relations between objects and exploit them in a state of the art CNN architecture. The aim is to produce a fast and reliable general purpose model for object detection exploiting the object semantic relations data to maximize object label agreement in object recognition.

2 Related work

There are two main theories about how objects are related. The widely accepted one is that our understanding of objects and their relationships with one another can be usefully captured by analysing the properties they possess, often referred to as semantic features. A number of large-scale feature listing studies have been conducted, in which participants are asked to generate features for a large set of objects (Cree & McRae, 2003; Devlin, Gonnerman, Andersen, & Seidenberg, 1998; Garrard, Lambon Ralph, Hodges, & Patterson, 2001; Tyler, Moss, Durrant-Peatfield, & Levy, 2000; Vinson, Vigliocco, Cappa, & Siri, 2003; Zannino, Perri, Pasqualetti, Caltagirone, & Carlesimo, 2006). In such studies, participants tend to produce features derived from perceptual experience (e.g., lemons are yellow), functional features concerned with behaviours or goals associated with the object (lemons are used to make drinks) and more abstract information that can typically only be expressed verbally (lemons are a type of citrus fruit). On this view, two objects are conceptually related to the extent that they share similar features; so oranges are semantically linked with lemons because they too are citrus fruits and are used to make drinks.

Feature generation studies of this kind have strongly endorsed the view that object knowledge is organised in terms of taxonomic category. Objects that belong to the same taxonomic category tend to share features (Cree & McRae, 2003) and, moreover, items that share many features with other items from their category are judged to be more prototypical members of the category (Garrard et al., 2001). Dilkina and Lambon Ralph (2012) recently demonstrated that items within the same category most frequently shared features that referred to their perceptual qualities, though functional and more abstract encyclopaedic features were also somewhat linked to taxonomic organisation.

The patterning of correlations amongst features and the relative salience of different types of feature have also been shown to vary across living and non-living things (Farah & McClelland, 1991; Garrard et al., 2001; Tyler et al., 2000). Living things are more strongly associated with perceptual features, for

example, and manufactured artefacts with functional features. These differences have been proposed to account for patterns of category-selective semantic deficits sometimes observed in a variety of neurological conditions (Cree & McRae, 2003; Farah & McClelland, 1991; Warrington & Shallice, 1984).

The feature-based approach to object knowledge has proved fruitful, with a number of models of object knowledge assuming that object concepts are structured in terms of their featural similarity (Collins & Quillian, 1969; McRae, deSa, & Seidenberg, 1997; Rogers et al., 2004; Rogers & McClelland, 2004; Tyler et al., 2000; Vigliocco, Vinson, Lewis, & Garrett, 2004). The idea that taxonomic category is a key organising principle for object concepts has also guided recent neuroimaging studies that have used multi-voxel pattern analysis to investigate representational structure (Devereux, Clarke, Marouchos, & Tyler, 2013; Fairhall & Caramazza, 2013; Kriegeskorte et al., 2008; Peelen & Caramazza, 2012).

Some limitations of the feature-based approach have been noted, however. It has been suggested that the feature generation task is biased towards features that distinguish objects from their category neighbours and towards aspects of information that can be easily expressed verbally (Hoffman & Lambon Ralph, 2013; Rogers et al., 2004).

Another, perhaps more fundamental, limitation is the fact that participants generating semantic features are asked to consider each object in isolation. The relationships between objects are therefore inferred indirectly, in terms of their feature overlap. This is not representative of our natural experience with objects. Environments typically contain many objects and most activities require us to interact with multiple objects simultaneously, which often have few features in common. To extend our earlier example, in order to make lemonade, life must give you not only lemons but water, sugar and a jug. How does the co-occurrence of these objects influence our conceptual representations of each of them?

An alternative approach to semantic representation has developed in the field of computational linguistics, based on the idea that semantic

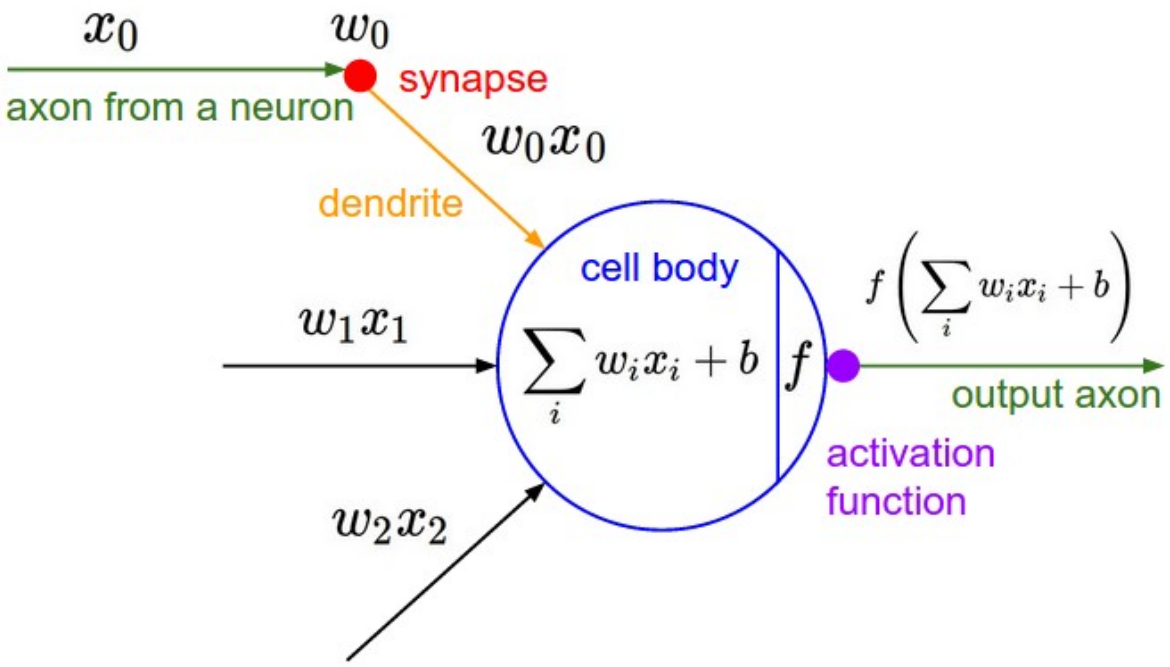
representations of words can be derived through statistical analysis of their distribution in large text corpora (Firth, 1957; Griffiths, Steyvers, & Tenenbaum, 2007; Landauer & Dumais, 1997; Lund & Burgess, 1996; Rohde, Gonnerman, & Plaut, 2006). The central tenet underpinning the distributional approach is the idea that words that occur in similar linguistic contexts are related in meaning. On this view, oranges and lemons would be considered similar because they co-occur with a similar set of words in natural language. For example, we might expect both orange and lemon to frequently occur in sentences that contain words like squeeze, cut, peel, pips, juice and marmalade. On the face of it, this does not sound so different to the featural approach.

However, the distributional approach allows for the possibility that objects from different taxonomic categories which share few features may nevertheless share a semantic relationship (e.g., lemon and ice may be considered semantically related because both words are used when we talk about making drinks). These associative or thematic relationships are known to play an important role in lexical-semantic processing. For example, significant semantic priming effects occur for word pairs that share an associative relationship as well as items that share semantic features (Alario, Segui, & Ferrand, 2000; Perea & Gotor, 1997; Seidenberg, Waters, Sanders, & Langer, 1984). Furthermore, children readily group objects according to their associative relationships and may even prefer this to grouping by taxonomic similarity (Kagan, Moss, & Sigel, 1963; Smiley & Brown, 1979), suggesting that associations play an important role in the development of concepts. Therefore lexical co-occurrence likely serves as an additional source of constraint over the structuring of object concepts, since it is able to capture associative relationships between items that share few features.

However, semantic models based on the distributional principle have been criticised because they rely solely on linguistic data and therefore do not take into account, at least in any direct way, the sensory-motor information available when we perceive and interact with objects in the real world (Andrews, Vigliocco, & Vinson, 2009; Glenberg & Robertson, 2000). Linguistic corpora may code perceptual experiences indirectly, of course, through verbal descriptions of sensory experiences.

[8][12]

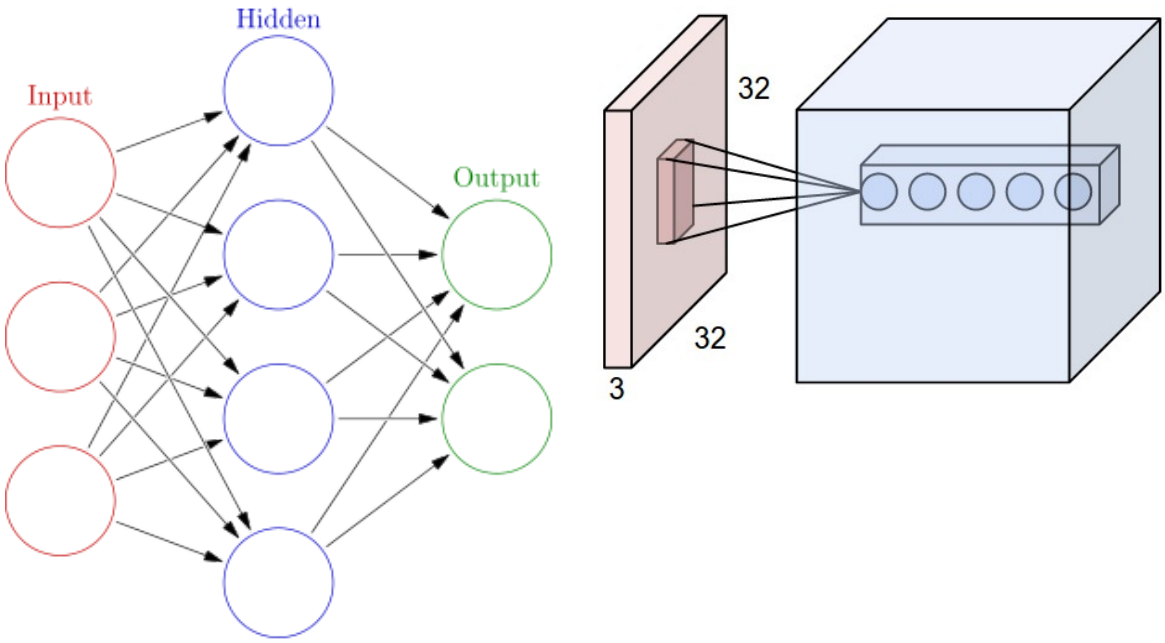
We’ve witnessed a significant improvements on the object detection task in recent years thanks to the advances in deep learning[1] .



Single neuron

Fig. 2

Convolutional Neural Networks are very similar to ordinary Neural Networks: they are made up of connections of neurons. Each neuron has learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. The whole network still expresses a single differentiable score function: from the raw image pixels on one end to class scores at the other. And they still have a loss function (e.g. SVM/Softmax) on the last (fully-connected).[13]



Neural Network (left) and part of Convolutional Neural Network (right)

Fig. 3

Current detection systems recast the detection problem into a classification problem. To detect an object, these systems first extract object proposal regions at various locations and scales in a test image and then apply a classifier for each proposal and evaluate it. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image. [4]

More recent approaches like R-CNN use region proposal methods to first generate potential bounding boxes in an image and then run a classifier on these proposed boxes. After classification, post-processing is used to refine the bounding boxes, eliminate duplicate detections, and rescore the boxes based on other objects in the scene . These complex pipelines are slow and hard to optimize because each individual component must be trained separately.[4]

Current state-of-the-art object detection systems are variants of the following approaches: hypothesize bounding boxes, resample pixels or features for each box, and apply a high quality classifier. This pipeline has prevailed on

detection benchmarks since the Selective Search work through the current leading results on PASCAL VOC, COCO, and ILSVRC detection all based on Faster R-CNN albeit with deeper features such as . While accurate, these approaches have been too computationally intensive for embedded systems and, even with high-end hardware, too slow for real-time applications. [2]

Often detection speed for these approaches is measured in seconds per frame (spf), and even the fastest high-accuracy detector, Faster R-CNN, operates at only 7 frames per second (fps). There have been many attempts to build faster detectors by attacking each stage of the detection pipeline but so far, significantly increased speed comes only at the cost of significantly decreased detection accuracy. [2]

3 Method

3.1 Overview

The goal of the proposed approach is to apply a kind of semantic regularization to object detection, aiming at disambiguate the recognition of multiple objects in the same image by taking into account object co-occurrence probabilities estimated from a training set. It's important to say that the semantic regularization only affect the recognition task within object detection. The localization accuracy is not changed as the experiment doesn't aim to improve localization, but rather to improve the recognition of similar objects by using contextual information.

3.2 Co-occurrence object relations

In order to represent contextual object relations we decided to use an object co-occurrence metric suggested by Dieu-Thu Le et. al. The idea is to obtain all conditional probabilities to gain idea of the context. Context is useful in visual recognition for two reasons: Firstly, context can significantly reduce the number of possible object categories simplifying the problem. Secondly, when the object appearance is inconclusive for its identity, context can be used for disambiguation. For example, a grey rectangle on a desk may be recognized as a pen, while a grey rectangle on a table may be recognized as a knife. As the recognition systems are not always reliable, the use of context can greatly improve results. [18]

Given an image I with N detected objects Error, the main idea is to force, for each object o_i, the conditional probabilities of finding o_i given all other objects except o_i, $O \setminus \{o_i\}$, [18] observed in the image, to be similar to the conditional probabilities observed in the training set. For example, if *laptop*

always co-occurrec with *hands* and *mug* in the training set, we expect to have the same in the test set.
By using the Naive Bayes assumption, the conditional probabilities can be computed as follows:

Error (1)

In this scenario, we need conditional the relations $P(o_j \mid o_i)$ and priors $P(o_i)$ which can be computed as follows.

$$P(o_j \mid o_i)$$

where o_i and o_j are objects detected on the image I. The probability for each object o_i we get from the following formula

$$P(o_i) = \frac{\#images\ having\ o_i}{\#images} \qquad (2)$$

Then

$$P(o_j \mid o_i) = \frac{\#images\ having\ o_j\ and o_i}{\#images\ having\ o_i} \qquad (3)$$

We can easily compute (1) from the training set. We can then use the obtained co-occurrence probabilities to embed them in a model in order to improve its predictions. We'll add an additional loss term in the loss function that compares the difference between the co-occurrence probabilities predicted by the model and the co-occurrence probabilities calculated by (1).

From the output of the model we have $P(o_i)$ for each object detected in the image (this correspond to the confidence of the detection) and the conditional probability $P(o_j \mid o_i)$ is equal to 1 for each j as we always have o_j given o_i, because they are present in the output obtained from the network.

A good candidate for a loss term is cross entropy. The cross entropy between two probability distributions p and q over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set, if a coding scheme is used that is optimized for an "unnatural" probability distribution q, rather than the "true" distribution p. It is commonly

used as regularization term in deep learning framework since has the benefit that, unlike the quadratic cost, it avoids the problem of learning slowing down.

3.3 Building the model

Current state-of-the-art object detection systems are variants of the following approach: hypothesize bounding boxes, resample pixels or features for each box, and apply a CNN classifier. This pipeline has prevailed on detection benchmarks since the Selective Search work through the current leading results on PASCAL VOC, COCO, and ILSVRC detection all based on Faster R-CNN albeit with deeper features. While accurate, these approaches have been too computationally intensive for embedded systems and, even with high-end hardware, too slow for real-time applications. [2]

Often detection speed for these approaches is measured in seconds per frame (spf), and even the fastest high-accuracy detector, Faster R-CNN, operates at only 7 frames per second (fps). There have been many attempts to build faster detectors by attacking each stage of the detection pipeline (see related work in Sec. 4), but so far, significantly increased speed comes only at the cost of significantly decreased detection accuracy. [2]

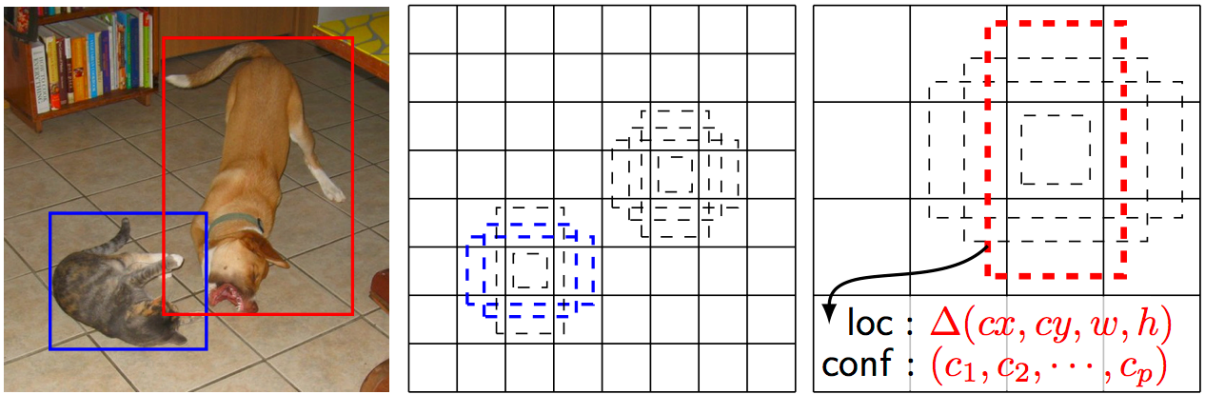
System	VOC2007 test *mAP*	FPS (Titan X)	FPS (Titan X)	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
Faster R-CNN (VGG16)	63.4	45	98	448 x 448
SSD300* (VGG16)	77.2	**46**	8732	300 x 300
SSD512* (VGG16)	**79.8**	19	24564	512 x 512

Multibox CNNs comparison

Table 1

For the purpose of applying the object co-occurrence data to a CNN we needed a fast and yet reliable CNN capable of detecting multiple objects. Based on the data on **Table 1** we decided to use the SSD CNN.

SSD is a single-shot detector for multiple categories that is faster than the previous state-of-the-art for single shot detectors (YOLO), and significantly more accurate, in fact as accurate as slower techniques that perform explicit region proposals and pooling (including Faster R-CNN). The core of SSD is predicting category scores and box offsets for a fixed set of default bounding boxes using small convolutional filters applied to feature maps. To achieve high detection accuracy it produces predictions of different scales from feature maps of different scales, and explicitly separate predictions by aspect ratio. These design features lead to simple end-to-end training and high accuracy, even on low resolution input images, further improving the speed vs accuracy trade-off. Experiments include timing and accuracy analysis on models with varying input size evaluated on PASCAL VOC, COCO, and ILSVRC and are compared to a range of recent state-of-the-art approaches. [2]



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

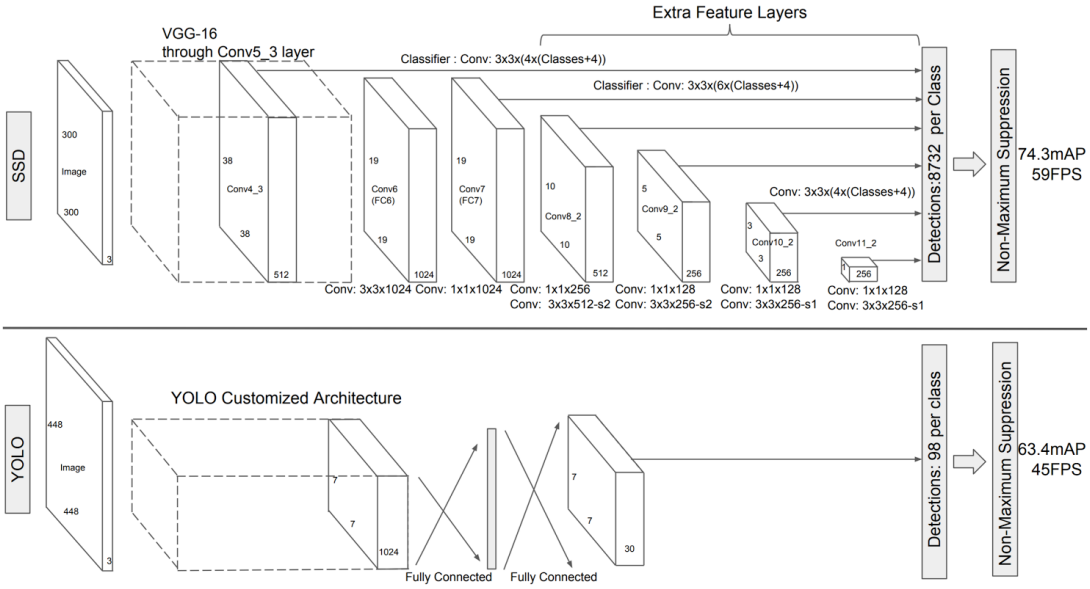
SSD Framework

Fig. 4

The architecture of the SSD (on **Fig. 5**) had to be modified following the idea presented the 3.2. So we redesigned the architecture of the SSD adding a few new layers shown on **Fig. 6**:

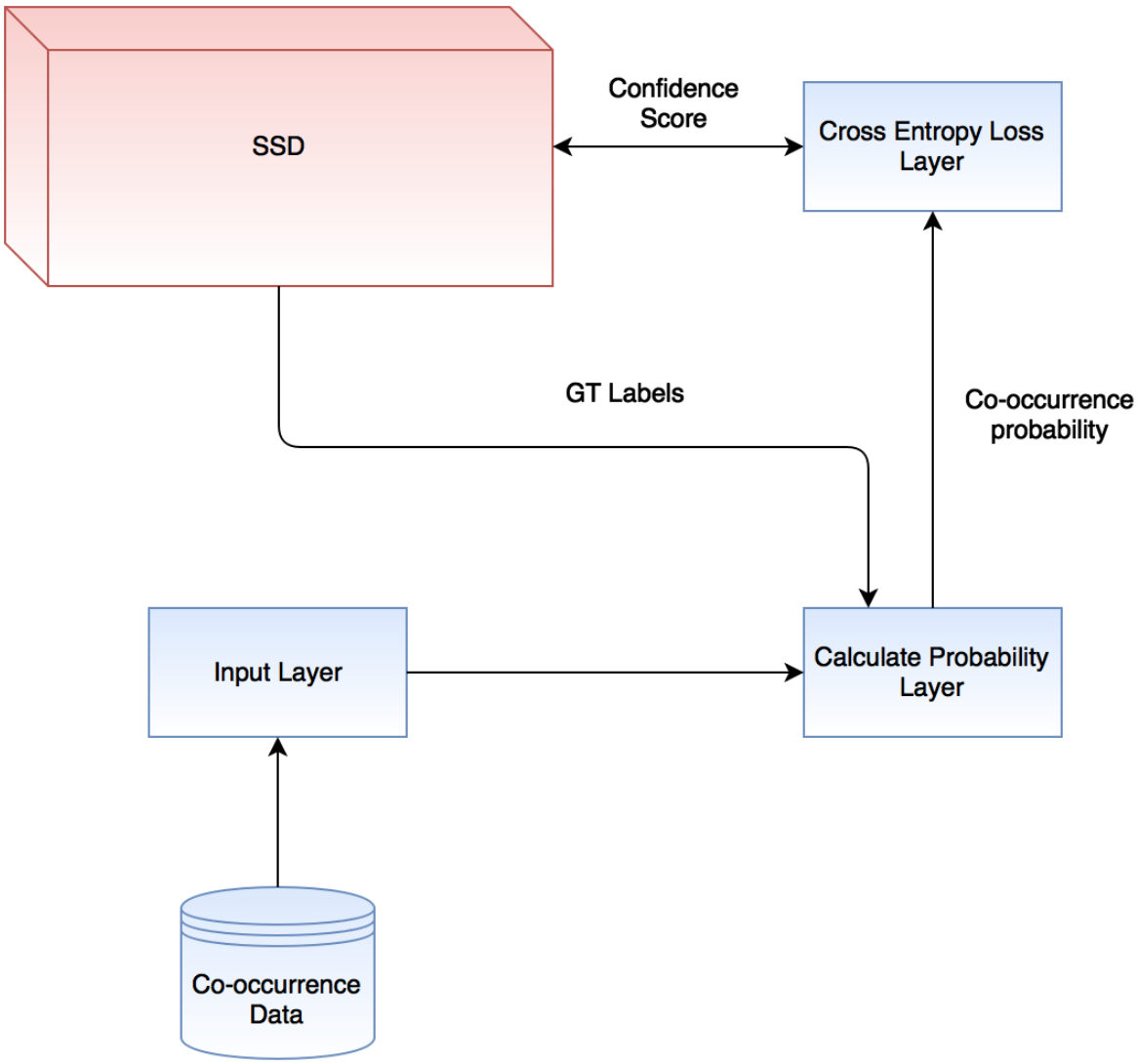
- An Input layer that loads the precalculated data from formulas (2) and (3);

- Calculate Probability Layer that uses as input the data from the new Input layer and takes the grand truth labels. This information is used to apply formula (1) and calculates the co-occurrence probabilities for the ground truth classes.
- Lastly we added a Cross Entropy Loss layer which takes as input both the predictions from the original SSD model and the co-occurrence probabilities and acts as semantic regularization.



SSD architecture

Fig. 5



Proposed design

Fig. 6

4 Data

As our experiments rely heavily on extracting a correct and reasonable data about the semantic relations of objects, we need data sets that match specific criteria. For a dataset to be usable for this work, except for the large amount of data and quality, which are essential for any machine learning task, it needs to have a good variety of objects, it needs to have bounding box annotations and there should be multiple objects per image. Each of this should be true for us to be able to extract the necessary semantic object relations information. Finding such natural images is hard[3][8] Throughout our research we inspected several data sets that match our criteria - MS-COCO, VOC Pascal Visual Object Classes (VOC), ILSRVC2016 and more.

MS COCO examples

Fig. 7

Microsoft COCO (Common Objects in COntext) is “a new large-scale dataset that addresses three core research problems in scene understanding: detecting non-iconic views (or non-canonical perspectives) of objects, contextual reasoning between objects and the precise 2D localization of objects”[3]. It’s focus is on collecting images that depict scenes in order to push research in contextual reasoning rather than objects in isolation. The dataset consists of 91 common object categories with 82 of them having more than 5,000 labeled instances. In total the dataset has 2,500,000 labeled instances in 328,000 images. Some examples can bee seen on **Fig. 7**.

Following the current trends in computer vision we investigated the available databases of wearable devices. Wearable devices or just wearables are, as the name suggests, small electronic devices that people wear with them throughout the day. They come in many forms - bracelets, necklaces, glasses, etc. **Fig. 8** Usually their goal is to collect and/or transmit some sort of data. For example the fitness oriented wearables are used to track the physical activities of the wearer such as steps count, walking distance, burned calories as well as sleep data. Some of these devices focus on images and collect data about human interactions and activities in the form of videos or image sequences. This last set of devices is of particular interest as it presents the wearers world in a perspective close to his own vision. Also the rich amount of information present on a visual scene allows for many applications like tracking habits and lifestyle - physical activity, diet, surroundings, etc. This data can later be used in conjunction with health reports to extract correlations. [14][15][16]

Some wearable devices

Fig. 8

One such dataset constructed from wearable cameras and matching the criteria for our experiment is the Activities of Daily Living dataset or ADL (**Fig. 9**). ADL is a dataset of 1 million frames of dozens of people performing unscripted, everyday activities. The dataset is annotated with activities, object tracks, hand positions, and interaction events. ADLs differ from typical actions in that they can involve long-scale temporal structure (making tea can take a few minutes) and complex object interactions (a fridge looks different when its door is open). The dataset itself consists of several videos of daily activities. For each video there is labeled data with bounding boxes for some of the more interesting frames - the appearances and disappearances of objects on the visual scene. One downside of the dataset is that the frames have to be extracted manually from the full videos in order to be used for training and testing purposes. [9]

Sample of annotated frames from the ADL dataset

Fig. 9

5 Implementation

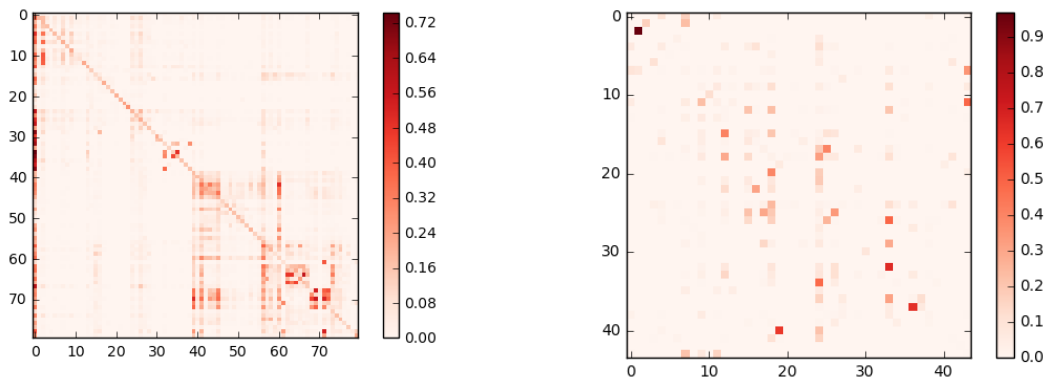
5.1 Co-occurrence probabilities

Researchers today across all academic disciplines often need to write computer code in order to collect and process data, carry out statistical tests, run simulations or draw figures. The widely applicable libraries and tools for this are often developed as open source projects (such as NumPy, Julia, or FEniCS), but the specific code researchers write for a particular piece of work is often left unpublished, hindering reproducibility.

Notebooks - documents integrating prose, code and results - offer a way to publish a computational method which can be readily read and replicated.

Jupyter notebook calculating co-occurrence for the COCO dataset

Fig. 10



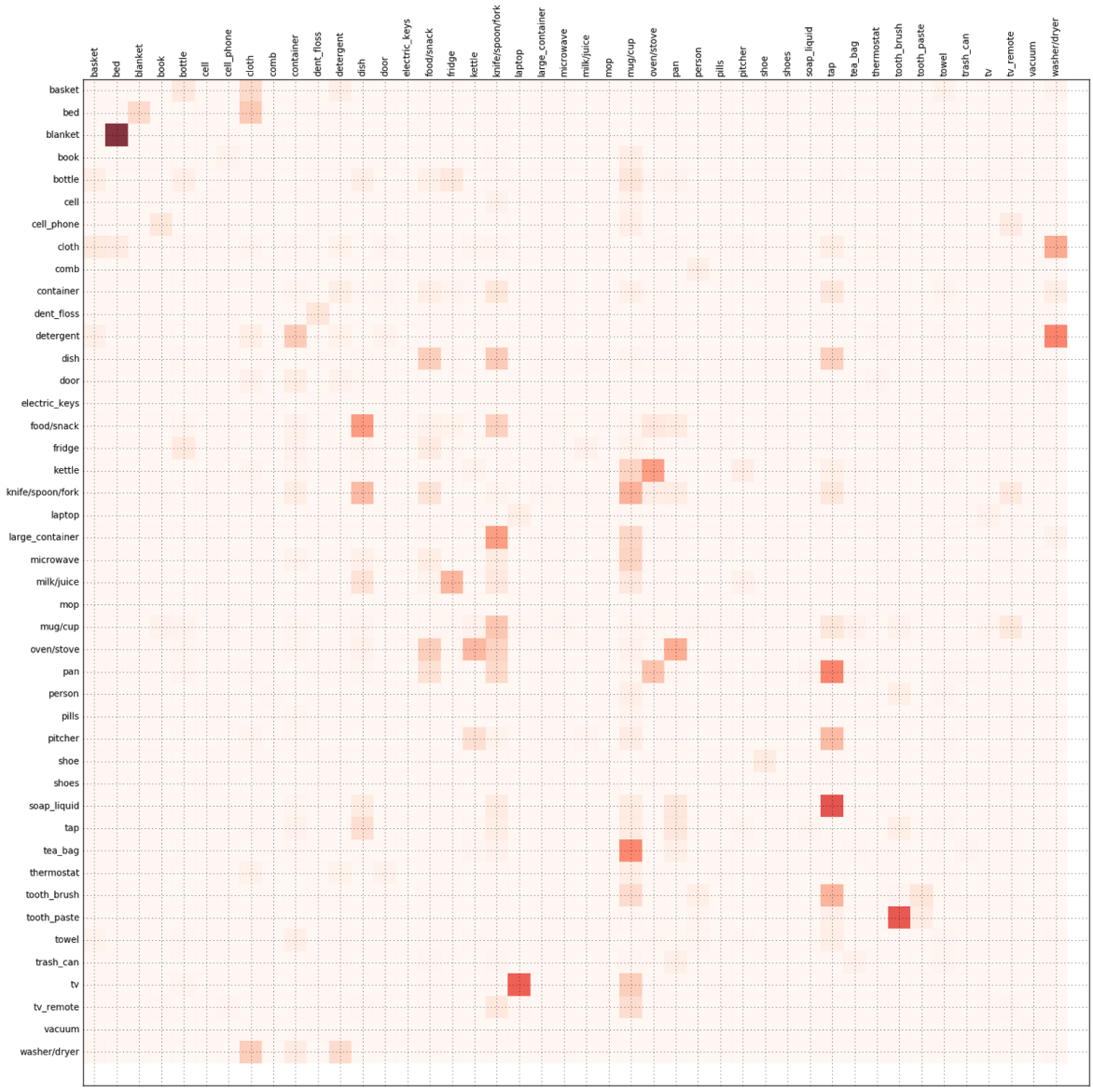
a) MS COCO b) ADL

MS COCO and ADL simple co-occurrence heat map

Fig. 11

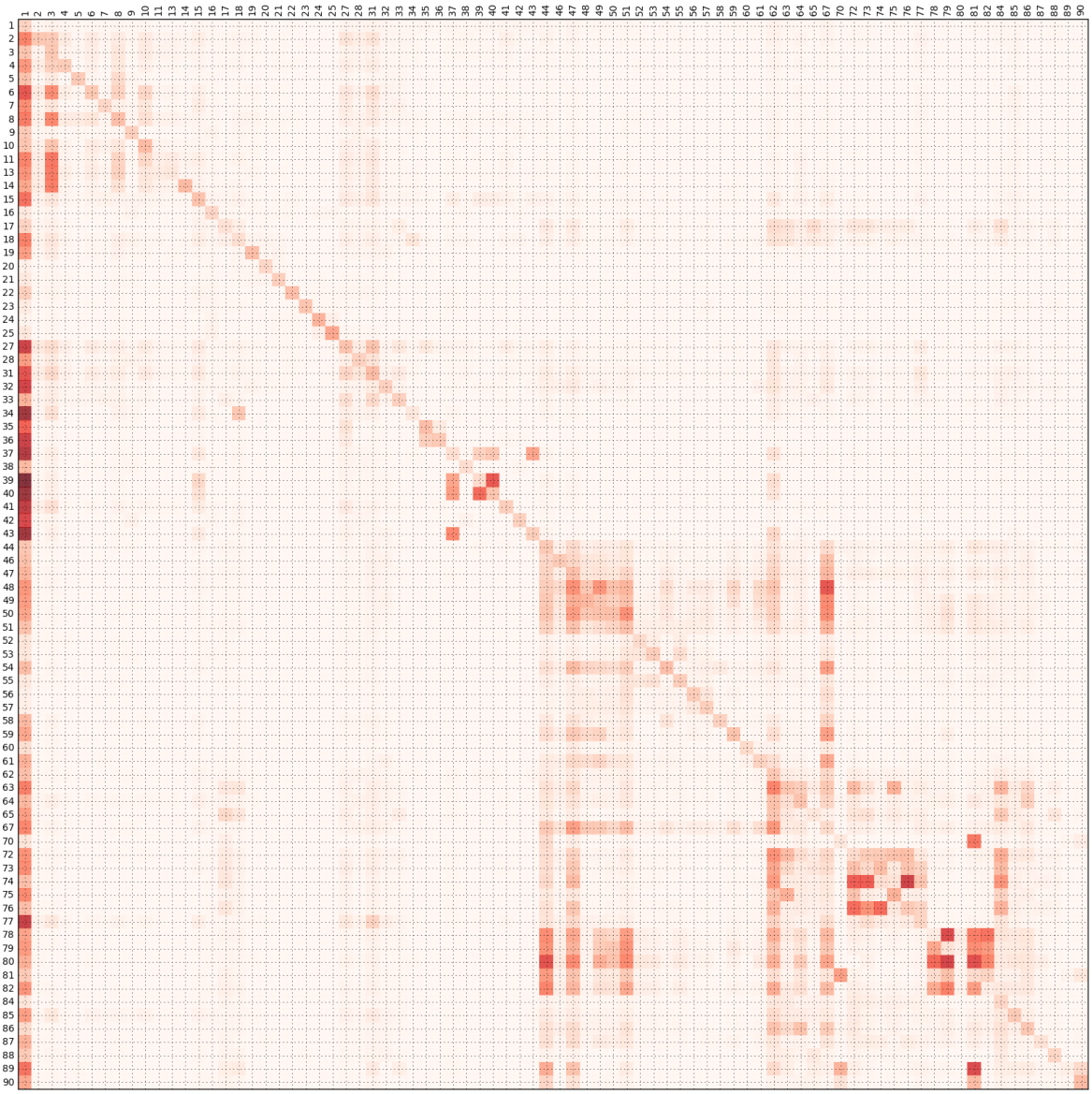
One such Notebook is Jupyter. Jupyter is an open source project, which can work with code in many different programming languages. Different language backends, called kernels, communicate with Jupyter using a common, documented protocol; over 50 such backends have already been written, for languages ranging from C++ to Bash. Jupyter grew out of the IPython project (Pérez & Granger, 2007), which initially provided this interface only for the Python language. IPython continues to provide the canonical Python kernel for Jupyter. [17]

The logic behind calculating the co-occurrence probabilities and preparing the ADL dataset in format suitable for the SSD convolutional neural network is implemented in python in the form of Jupyter Notebooks (**Fig. 10**). The heatmaps of the two data sets co-occurrence can be seen seen on **Fig. 11-13**



Detailed ADL co-occurrence heatmap

Fig. 12



Detailed MS COCO co-occurrence heatmap

Fig. 13

5.2 Models

The Single Shot Detector is implemented using the Caffe framework. Caffe provides multimedia scientists and practitioners with a clean and

modifiable framework for state-of-the-art deep learning algorithms and a collection of reference models. The framework is a BSD-licensed C++ library with Python and MATLAB bindings for training and deploying general purpose convolutional neural networks and other deep models efficiently on commodity architectures. Caffe fits industry and internet-scale media needs by CUDA GPU computation, processing over 40 million images a day on a single K40 or Titan GPU (≈ 2.5 ms per image). By separating model representation from actual implementation, Caffe allows experimentation and seamless switching among platforms for ease of development and deployment from prototyping machines to cloud environments. Caffe is maintained and developed by the Berkeley Vision and Learning Center (BVLC) with the help of an active community of contributors on GitHub. It powers ongoing research projects, large-scale industrial applications, and startup prototypes in vision, speech, and multimedia. [10]

Having the majority of the code written using this framework and having a lot of pretrained models and CNN architectures in format for this specific framework, the right decision was to implement the designed modifications on that platform.

This was the most complex part of the whole experiment. The SSD implementation has brought many changes to the Caffe framework. This includes modifications to existing code, addition of new functions and even new custom layers. The Caffe framework by itself lacks good documentation and the addition of custom implementations on top of it makes modifications on the framework extremely hard. A lot of effort was put into understanding the concepts of the framework itself as well as the implementation and behavior of the SSD additions brought to it. This knowledge was used to later build the additional layers and required modifications to the SSD code to make the object co-occurrence experiment be possible on the SSD network.

6 Experimental results

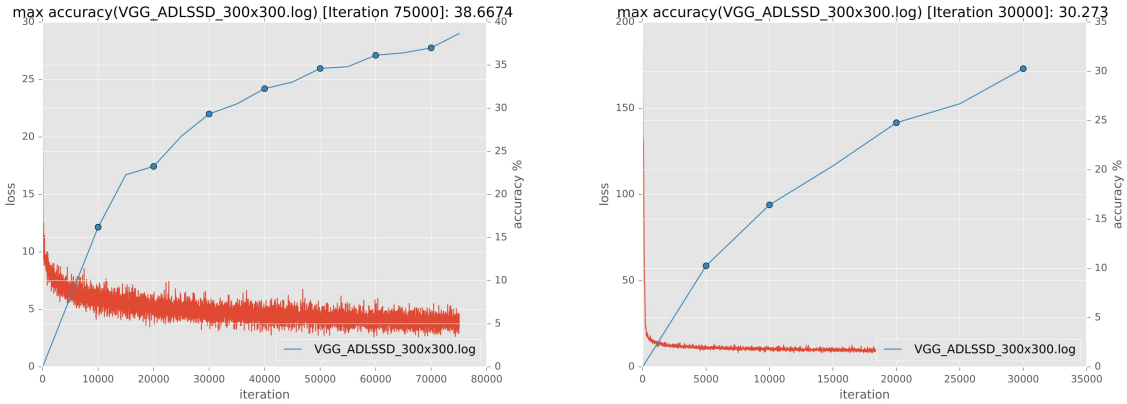
As the ADL data set is a new dataset for the Single Shot Detector, the first part of experiments was designed around training and fine tuning a model based on this dataset. On **Table 2** are shown some of the results of for several configurations. We iterate over different batch sizes, learning rates, solutions including additional back propagation to layers, reducing the number of classes to the most common ones, fine tuning a pretrained COCO mode and different learning rates. Some experiments were terminated prematurely due the lack perspective of achieving good results compared to other competitor configurations over the same iterations.

Dataset	Number of classes	Iterations	Score
ADL (batch size 8)	44	70 000	25%
ADL (batch size 16)	44	35 000	29%
ADL FT (batch size 16, low lr, no back propagation)	44	2 000	3%
ADL FT (batch size 16, no back propagation)	44	30 000	22%
ADL FT (batch size 16, no back propagation)	44	120 000	39%
ADL (batch size 16)	44	120 000	**40.05%**
ADL (batch size 16)	21	40 000	30%

ADL SSD models

Table 2

The second part of experiments was oriented towards improving the SSD model accuracy by using object co-occurrence data. As with the first experiment we conducted experiments on a wide range of parameters and tweaks to the way we calculate the cross entropy loss. One of the first set of experiments was to try to apply the regularization while we also do the initial training of the model. This approach gave somewhat optimistic hopes as it managed to beat the standard training with about **1%** for the first 30k iterations (**Fig. 14**). Unfortunately it had just the opposite effect on the MS COCO dataset where the model could not pass the 1% test accuracy barrier on the first 10k iterations.



a) Regular ADL training b) ADL training with cross entropy

Fig. 14

Our second set of this experiment was targeted around applying the cross entropy loss layer to pretrained models by fine tuning. This time both datasets showed consistent behavior. Unfortunately the proposed approached worsen the results on the pretrained models by a little margin as seen on the table below.

Dataset	Pretrained model accuracy	Accuracy after fine tuning 1k iterations
MS COCO	43.0362	43.0265
ADL	40.05	40.04

6 Conclusions and future work

The best result that was achieved for the ADL data set is **40.05%**. This result was achieved over **120,000** iterations with batch size **16**, learning rate of **0.006** on a fresh training over **44** classes. This result does not seem high by itself. However, if we take into account that the dataset was created to be difficult for object detection having objects looked from different perspectives and after some transformations (for example close look at a fridge with opened door). Also the images in the dataset were extracted from video files shot with a low quality wearable camera which adds some blur and motion noise (see **Fig. 15**). In addition the size of the dataset is not that large compared to the sizes of other state of the art datasets.

Example of noisy samples from the ADL dataset

Fig. 15

The results of the application of object co-occurrence on convolutional neural networks are a bit disappointing. However, this can be expected when experimenting with something that has not been done before. What's interesting is the big gap in the behavior of the model on the two datasets on a fresh train. The model performed a bit better on the ADL dataset with cross entropy layer added while on the MS COCO side the result was disastrous. This is probably caused by the nature of the datasets and the calculations used to get the co-occurrence scores.

The ADL dataset has a very limited number of co-occurrences per picture. They average at about 2 objects per picture. This can be seen on the heatmap for ADL on **Fig. 12**. Having a look back at formula (1) this means that the time the co-occurrence score for an object will be just the product $P(o_i) * P(o_j \mid o_i)$.

On the other hand the MS COCO has a lot of objects per image - "s 2,500,000 labeled instances in 328,000 images". This means an average of 7.6 detections per image. Let's look at formula (1) and think about the following scenario - that each co-occurrence is equally likely $P(o_j \mid o_i) = y$ and $P(o_i) = x$. Then for ADL we'll have on average $x * y$ and for COCO we'll have $x * y^7$. taking into account that x and y are in the interval [0,1] the chances are that for MS COCO we'll get results very close to zero as co-occurrence most of the time. As these co-occurrences are considered ground truth at the cross entropy layer, it's no surprise that we see a negative effect on the performance of the model.

Further enhancement of object detection by taking into account relative positions between objects: Bar et al. [M. Bar and S. Ullman. Spatial context in recognition. Perception. 25:343-352., 1993] examined the consequences of pairwise spatial relations between objects that typically co-occur in the same scene on human performance in recognition tasks. This study has shown that proper spatial relations among objects decreases error rates in the recognition of individual objects. Future work will aim to exploit the knowledge about spatial relations to improve the recognition performances.

Enhancement of image tagging: Exploiting object co-occurrence has a direct application to image tagging, whose goal is to label an image with a set of tags that describe the image content, including the objects appearing in it. In image tagging there is no need of localising objects, so the proposed framework could be easily adapted to this context in the future.

7 References

[1] Gidaris, Spyros, and Nikos Komodakis. "Object detection via a multi-region and semantic segmentation-awarecnn model." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

[2] Liu, Wei, et al. "SSD: Single shot multibox detector." *European Conference on Computer Vision*. Springer International Publishing, 2016.

[3] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." *European Conference on Computer Vision*. Springer International Publishing, 2014.

[4] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[5] Cordts, Marius, et al. "The cityscapes dataset for semantic urban scene understanding." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[6] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[7] Girshick, Ross, et al. "Deformable part models are convolutional neural networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[8] Sadeghi, Zahra, James L. McClelland, and Paul Hoffman. "You shall know an object by the company it keeps: An investigation of semantic representations derived from object co-occurrence in visual scenes." *Neuropsychologia* 76 (2015): 52-61.

[9] Pirsiavash, Hamed, and Deva Ramanan. "Detecting activities of daily living in first-person camera views." *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012.

[10] Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014.

[11] Galleguillos, Carolina, Andrew Rabinovich, and Serge Belongie. "Object categorization using co-occurrence, location and appearance." *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008.

[12] Yang, Bishan, et al. "Embedding entities and relations for learning and inference in knowledge bases." *arXiv preprint arXiv:1412.6575* (2014).

[13] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[14] Fishkin, Kenneth P., Matthai Philipose, and Adam Rea. "Hands-on RFID: Wireless wearables for detecting use of objects." *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*. IEEE, 2005.

[15] Mayol, Walterio W., et al. "Applying active vision and slam to wearables." *Robotics Research* (2005): 325-334.

[16] Cartas, Alejandro, et al. "Recognizing Activities of Daily Living from Egocentric Images." *arXiv preprint arXiv:1704.04097* (2017).

[17] Kluyver, Thomas, et al. "Jupyter Notebooks—a publishing format for reproducible computational workflows." *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (2016): 87.

[18] Le, Dieu-Thu, Jasper RR Uijlings, and Raffaella Bernardi. "Exploiting Language Models for Visual Recognition." EMNLP. 2013.