# THRASHOLDER

CMPE 492: Project Report


**Team**

Ahmet Genç (115200056)

Ayhun Demir (113200027)

Umut Can Cömert (116200081)

**Advisor**

Uzay Çetin

**Instructor**

Elif Pınar Hacıbeyoğlu



DEPARTMENT OF COMPUTER ENGINEERING

ISTANBUL BILGI UNIVERSITY

# Abstract

The environment is an external environment in which living things maintain their relationships throughout their lives. Air, water and soil are the physical elements of this environment and human, animal, plant and other microorganisms are the biological elements. Disruption of natural balance as a result of excessive and wrong use of the resources on the world, and also bringing about many problems is called environmental pollution.

In addition to the many positive and comfort brought to our lives by the developing technology, the impact of this development on the environment, the size of the pollution, increases rapidly each year. Nowadays, due to the increase in environmental pollution effects and the direct negative effects of these effects on human health, the importance given to the prevention of environmental pollution and the studies to be done in this direction is increasing.

Technology is advancing day by day and new techniques are developing. Recently, image processing techniques, which have become a trend, play an important role here.

Automatically running image processing models have great potential to detect and classify objects that are garbage. By using these models, these objects can be classified and parsed according to their purpose.

According to a World Bank research, 98 percent of the waste around the world is sent to landfills. Although the vast majority of these materials are recyclable, they are throwing into the oceans or burning *[1].*

This fact that we have to face makes clear how important waste management is. Technology and artificial intelligence play a crucial role at this point. The aim of this project is to automatically decompose garbage and increase recycling rates. Here, successful models will be developed using different deep learning techniques. In this way, substances that need to be recycled and cannot be converted can be recycled and evaluated.

# CONTENTS

## List Of Figures



## List Of Tables

# 1 Introduction

## *1.1 Problem Statement*

In order to meet the needs of the rapidly growing world population, industrialization needs to increase due to the development of technology. With this increase in industrialization, it causes the depletion of natural resources rapidly. While natural resources are rapidly depleted, disposing of wastes from production and consumption into the environment without taking precautions provides an environment for environmental pollution. In fact, many of these wastes are produced from recyclable raw materials, but this is not considered because of the ignorance and indifference of people. Technology comes into play here. The waste management processes of the municipalities begin with the waste thrown into the trash bins. The waste collected in the trash cans is taken to the collection centers by private company trucks. In these centers, worker power is used to classify these wastes. The workers working here create a cost for companies and the work done by people cannot be done quickly enough. Also, while some of the products collected here can be separated for recycling, most of them are disposed of as waste. Therefore, many recyclable products are wasted. If these wastes can be separated automatically in each garbage bin and separated from there, they can be recycled at the collection center. Another point is for the waste coming to the factories in complex form. If the wastes coming here can be separated automatically, the recycling rate here will be greatly increased. If the recycling rate does not rise in the near future, it is not a good situation for the world ecosystem. Non-recycling of these wastes means extra harm to the creatures in the nature. Because these wastes do not disappear in nature for many years. They are threat the habitats of other living creatures and disrupt their biology. In order to keep the balance of nature, not to consume natural resources and to build a good future for new generations, environmental pollution should be reduced to the lowest levels by recycling the wastes that must be recycled at the highest rates. This project aims to increase the percentage of recyclable products. Thanks to artificial intelligence based image processing models, this process is very simple by automatically sorting out garbage, so that recyclable products can be recycled.

## *1.2 Application*

This project will decompose garbage in real time using a camera through the raspberry Pi, by force of the CNN-based image processing model embedded inside the raspberry Pi. The key point here is to create the most accurate model to be trained. There are two important factors in creating the model. The first is to work with a quality data set, and the second is to get the picture from the camera in its cleanest form by using the selected image processing techniques correctly.

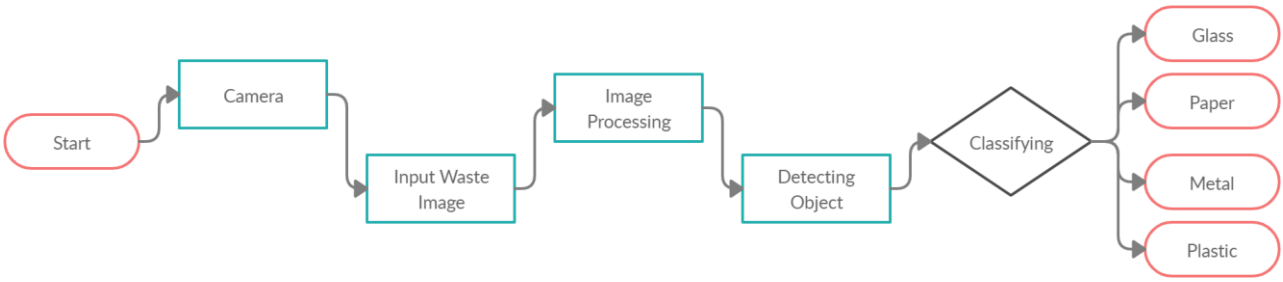


*Figure 1: Workflow Diagram*

In order to work with a quality data set, a wide data set was prepared to meet the needs of the project. Many trash pictures were taken from different types and angles. When taking pictures, it was important to work with the model at maximum efficiency. The variety of posture types and backgrounds of garbage is provided in the pictures as well as the type of garbage. The pictures were then tagged.

### *1.3 Challenges*

The biggest challenge is to produce a model with a good accuracy rate. Model architecture, data set and image processing techniques are the main factors to be used for this. The model architecture to be chosen varies according to the purpose of the work done, so a model architecture should be created for the purpose. The selected data set must contain many and varied images. Finally, the image processing techniques to be applied must be used in the most accurate way.

Apart from the difficulties expected during the construction of the project, many other problems were encountered. Some of these are hardware difficulties such as overheating of the computer's GPU used during model training and experiencing difficulties in transferring the trained model to Raspberry Pi. Others are experience-based issues like the team working with Raspberry Pi for the first time. In addition, although it did not directly affect the development of the project, the importance of staying home around the world due to the pandemic disease (COVID-19) at the time of the project's development had an adverse effect on intra-team communication. It extended the time to take action against task sharing and problems encountered.

## 2 Hardware

### *2.1 Raspberry pi*

Raspberry Pi is a single board computer which is credit card sized.

This tool can be used by attaching it to a computer monitor or any display provider screen. Besides being portable and portable, it caters to all kinds of people who want to meet the computer. You can browse the internet, watch videos and various activities in Raspberry pi. In addition, code can be written in programming languages such as Scratch and Python. In addition, since there are different external plugins, it can be used in many areas for different experiences and projects. By way of example, a camera to be connected to the raspberry Pi for image processing can be used for object detection, recognition, monitoring and classification.

# 3 Object Detection

Object detection is a computer vision technique. Allows determine which classes of objects in video and images belong to. Models are trained to perform this operation. Machine learning algorithms are used to train these models. The aim of object detection is to gain the ability to parse the objects to computers.

## *3.1 CNN (Convolutional Neural Network)*

At this point, you may definitely think about machine learning and deep learning, a software engineering branch that reviews the structure of calculations that can learn. Deep learning is a subfield of machine learning that is propelled by artificial neural networks, which thusly are enlivened by natural neural systems *[2]*.

In deep learning, a convolutional neural system is the class of deep neural systems most commonly applied to exploring visual imagery. They are in any case considered artificial neural systems whose normal burdens are invariant or field invariant depending upon their plan and understanding invariance characteristics.

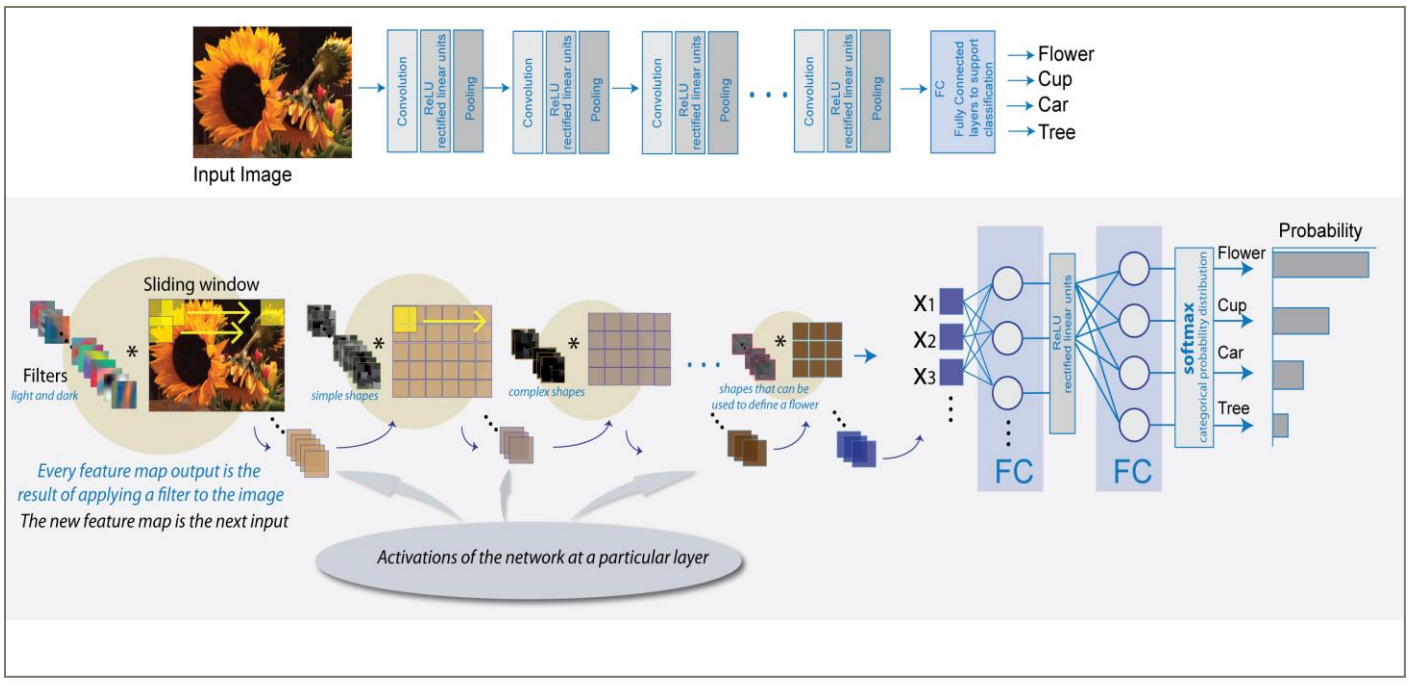


*Figure 2: How to work CNN?*

A particular sort of such a deep neural system is the convolutional organize, which is generally alluded to as CNN or ConvNet. It's a profound, feed-forward fake neural system.

CNNs explicitly are roused by the organic visual cortex. The cortex has little locales of cells that are delicate to the particular regions of the visual field. This thought was extended by an enthralling trial done by Hubel and Wiesel in 1962. In this investigation, the scientists demonstrated that some individual neurons in the cerebrum enacted or terminated uniquely within the sight of edges of a specific direction like vertical or even edges. For instance, a few neurons terminated when presented to vertical sides and some when indicated an even edge. Hubel and Wiesel found that these neurons were all around requested in a columnar manner and that together they had the option to deliver visual observation. This thought of particular segments within a framework having explicit errands is one that machines use too and one that you can likewise discover back in CNNs.

A neural framework is a plan of interconnected fake "neurons" that exchange messages between each other. The affiliations have numeric burdens that are tuned during the readiness methodology, so a properly arranged framework will respond successfully when given an image or guide to see. The orchestrate involves various layers of feature recognizing "neurons". Each layer has various neurons that respond to different mixes of commitments from the past layers.

In this way, we see that it can easily be treated as a biological issue and that people who work in the biological field will clearly use CNN.

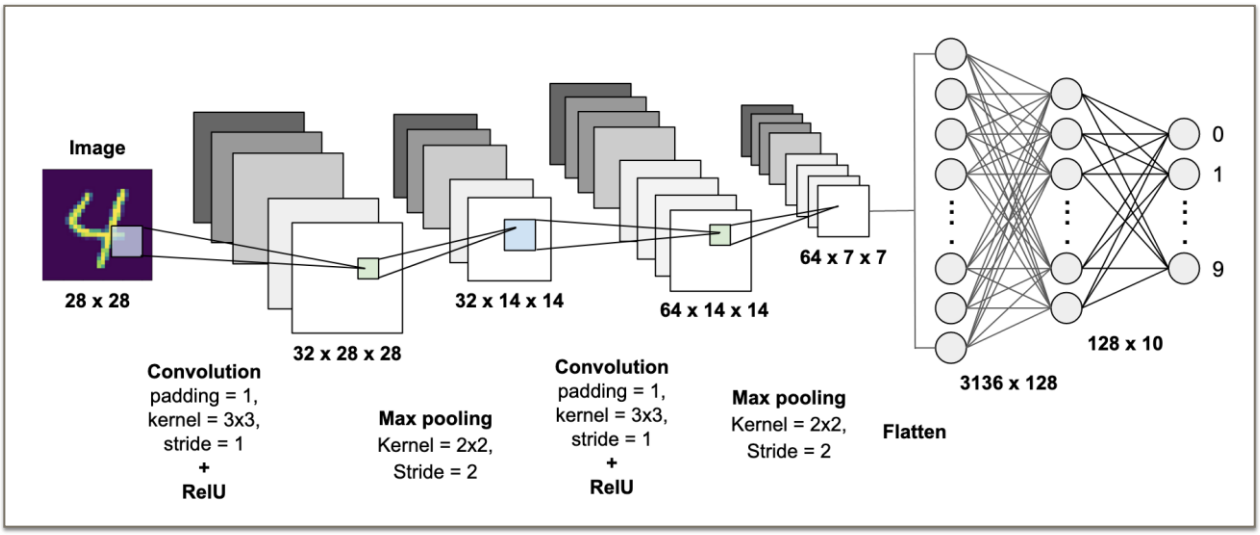


*Figure 3: Architecture of CNN*

### *3.1.1 Architecture*

A CNN comprises of various convolutional and subsampling layers alternatively followed by completely associated layers. The contribution to a convolutional layer is a m x m x r picture where m is the stature and width of the picture and r is the quantity of channels, for example a RGB picture has r=3. The convolutional layer will have k channels (or bits) of size n x n x q

where n is littler than the element of the picture and q can either be equivalent to the quantity of channels r or littler and may shift for every portion *[3].* The size of the channels offers ascend to the privately associated structure which are each convolved with the picture to deliver k include maps of size m−n+1. Each guide is then subsampled regularly with mean or max pooling over p x p touching districts where p runs between 2 for little pictures and is typically not more than 5 for bigger sources of info. Either previously or after the subsampling layer an added substance inclination and sigmoidal nonlinearity is applied to each element map *[4].*

After the convolutional layers there might be any number of completely associated layers. The thickly associated layers are indistinguishable from the layers in a standard multilayer neural system.

### *3.1.2 Convolutional Layer*

It is utilized to recognize properties. This layer is the principle building square of CNN *[5]*. It is answerable for seeing the properties of the picture. This layer applies a few channels to the picture to evacuate the low-and elevated level highlights in the picture. The convolutional layer's parameters comprise of a lot of learnable channels. Each channel is little spatially (along width and tallness), yet reaches out through the full profundity of the information volume. For instance, an average channel on a first layer of a CNN may have size 5x5x3 (for example 5 pixels width and stature, and 3 since pictures have profundity 3, the shading channels). During the forward pass, we slide (all the more exactly, convolve) each channel over the width and stature of the information volume and process speck items between the sections of the channel and the contribution at any position. As we slide the channel over the width and stature of the information volume we will deliver a 2-dimensional actuation map that gives the reactions of that channel at each spatial position. Instinctively, the system will learn channels that enact when they see some sort of visual element, for example, an edge of some direction or a smudge of some shading on the main layer, or in the end whole honeycomb or wheel-like examples on higher layers of the system.

### *3.1.3 Non-Linearity Layer*

The non-Linearity layer is generally trailed by all Convolutional layers. So why is appearing exactness an issue? The issue is that since all layers can be an exact capacity, the neural system goes about as a solitary discernment, implying that the outcome can be determined as a straight mix of yields. This layer is known as the Activation Layer since one of the actuation capacities is utilized. Before, non-exact capacities, for example, sigmoid were utilized, yet in addition to started to utilize this capacity as it gave the Rectifier (ReLu) work the best outcome on the speed of neural system preparing.

ReLu work f (x) = max (0, x)

### *3.1.4 Pooling Layer*

This layer is regularly included between progressive convolutional layers in CNN. The errand of this layer is to decrease the spatial size of the portrayal and the quantity of parameters and calculations inside the system. Along these lines, the system inconsistency is checked. There are many Pooling activities, yet the most mainstream is max pooling. There are additionally normal pooling and L2-standard pooling calculations that take a shot at a similar guideline. To begin with, how about we make a channel of 2×2 size. This permits the neural system to utilize littler yields that contain enough data to settle on the correct choice. In mix with this, numerous individuals don't like to utilize this layer. Rather, the bigger Stride (moving the channel) is favored in the Convolutional layer. Moreover, increasingly beneficial models, for example, VAEs or generative antagonistic systems totally expel the pooling layer.

### *3.1.5 Stride*

This worth demonstrates that the channel, which is the weight network for the convolution procedure, will move the picture in one pixel step or in bigger advances. This is another parameter that legitimately influences the yield size. For instance, when the cushioning esteem is p=1, the yield size is (3)x(3) if the quantity of steps is determined for the size=5 and F=3 of the yield framework when s=2 is chosen. Another approach to apply it is to duplicate the estimation of the Pixel close to it.

### *3.1.6 Padding*

Dealing with the size distinction between the info mark and the yield mark after the convolution procedure is a count that we have. This procedure is accomplished with additional pixels to be added to the info network. This is called padding.

### *3.1.7 Flattening Layer*

The undertaking of this layer is basically to set up the information in the section of the last and most significant layer, the Fully Connected layer. All in all, neural systems take input information from a one-dimensional cluster. The information in this neural system is a one-dimensional cluster of lattices from Convolutional and Pooling layers.

### *3.1.8 Fully-Connected Layer*

The fully-connected layer *[6]* works on an info where each information is associated with all neurons. FC layers, assuming any, are normally situated towards the finish of the CNN engineering and can be utilized to advance objectives, for example, class scores. This layer is the last and most significant layer of CNN. It takes the information from the Flattening procedure and makes the learning procedure conceivable through the neural system.

### *3.1.9 Why CNN?*

While neural systems and other example recognition techniques have been around for as far back as 50 years, there has been noteworthy improvement in the region of convolutional neural systems in the ongoing past. This area covers the benefits of utilizing CNN for picture acknowledgment. Recognition utilizing CNN is tough to contortions, for example, change fit as a fiddle because of camera focal point, distinctive lighting conditions, different presents, nearness of incomplete impediments, flat and vertical movements, and so on. Notwithstanding, CNNs are move invariant since a similar weight setup is utilized across space. In principle, we likewise can accomplish move invariants utilizing completely associated layers. Be that as it may, the result of preparing for this situation is various units with indistinguishable weight designs at various areas of the information. To become familiar with these weight arrangements, countless preparing occasions would be required to cover the space of potential varieties. In this equivalent theoretical situation where we utilize a completely associated layer to remove the highlights, the information picture of size 32x32 and a concealed layer having 1000 highlights will require a request for 106

coefficients, a tremendous memory prerequisite. In the convolutional layer, similar coefficients are utilized across various areas in the space, so the memory necessity is definitely diminished. Again utilizing the standard neural system that would be comparable to a CNN, in light of the fact that the quantity of parameters would be a lot higher, the preparation time would likewise increment proportionately. In a CNN, since the number of parameters is definitely diminished, preparing time is proportionately decreased. Likewise, accepting immaculate preparing, we can plan a standard neural system whose presentation would be same as a CNN. In any case, in down to earth preparing, a standard neural system proportional to CNN would have more parameters, which would prompt more clamor expansion during the preparation procedure. Subsequently, the exhibition of a standard neural system identical to a CNN will consistently be less fortunate.

Among the promising territories of neural systems inquire about are intermittent neural systems (RCNNs) utilizing long momentary memory (LSTM). These zones are conveying the present best in class in time-arrangement acknowledgment assignments like discourse acknowledgment and penmanship acknowledgment. RCNN/autoencoders are likewise fit for creating penmanship/discourse/pictures with some known appropriation. Profound conviction organizes, another promising sort of system using confined Boltzman machines (RMBs)/autoencoders, are equipped for being prepared covetously, each layer in turn, and henceforth are all the more effectively trainable for profound systems. CNNs give the best execution in design/picture acknowledgment issues and even beat people in specific cases.

## *3.2 Model Types*

### *3.2.1 SSD (Single Shot Detector)*

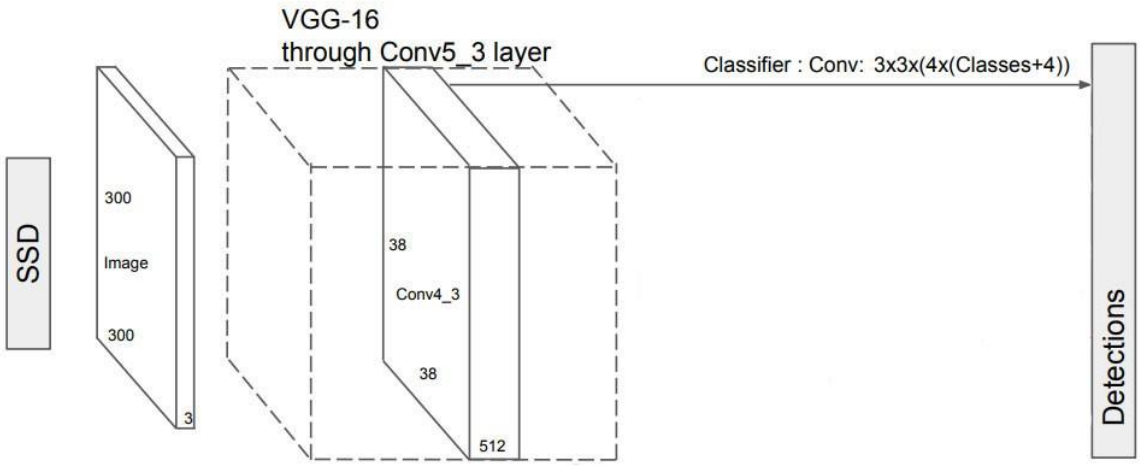


*Figure 4: Architecture of SSD*

SSD stands for “single shot detector”. In other words, a single shot is made to detect the objects in the images. In the RPN (Region Proposal Network) based approaches such as Faster-RCNN, two shots are taken. One of these shots is made for the region offer, and the other is for finding the objects in these offers. These two events happen in SSD at once. Therefore, SSD models are faster than RPN-based approaches.

SSD's architecture is built on the VGG-16 architecture, but it does not contain fully connected layers. VGG-16 has a successful performance in image classification works, so it has been chosen for use in SSD *[7]* architecture. Auxiliary convolutional layers are used here instead of fully connected layer. In this way, it is ensured that the features are removed in multiple scales and the size of the entrance to each next layer is decreased gradually.

How the SSD works is as follows, an image is taken as input. Feature maps of different sizes are obtained by passing this image through convolutional neural network. In all feature maps, a small amount of bounding rectangles are obtained with the help of 3x3 convolutional filter. Classes and constraints are then determined for each rectangle. Because these rectangles are on every activation map, they can recognize both small objects and large objects. The correct boundaries are compared with the boundaries that are estimated during the training. As a result, it takes images as input and returns tensor values as output.

### *3.2.2 Faster RCNN*

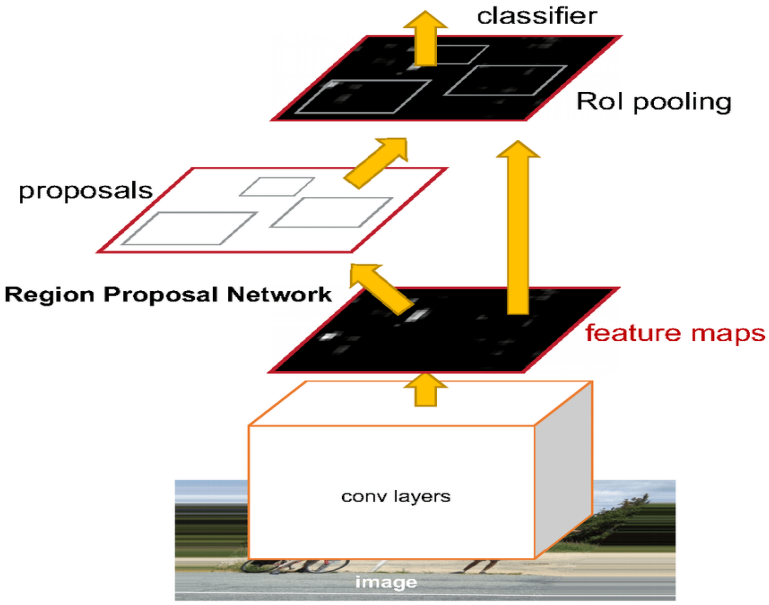


*Figure 5: Architecture of Faster-RCNN*

Faster RCNN is an architecture for object detection. An image is taken as input. This image is put into the process of convulsion as a whole. In this way, operations are made much faster than the classical RCNN method. Because passing a single image is a much shorter process than passing a lot of regions through CNN. A high resolution feature map that matches the original image is obtained as output. Unlike Fast-RCNN, Faster-RCNN uses RPN (Region Proposal Network) [8], which is a separate region suggestion network instead of receiving region suggestions with selective search. Region proposals are made on this network. This network is the part that makes Faster RCNN fast. RPN *[8]* has basically two different roles. Each incoming region should check whether there are any objects in it and it should determine the proposed region's window size. After that outputs are given to fully connected layer. Then these outputs are given to the fully connected layer, this layer classifies the regions and determines the boundaries of the objects. Softmax method is used for classification of regions and linear regression method is used to determine object boundaries.

### *3.2.3 Mask RCNN*

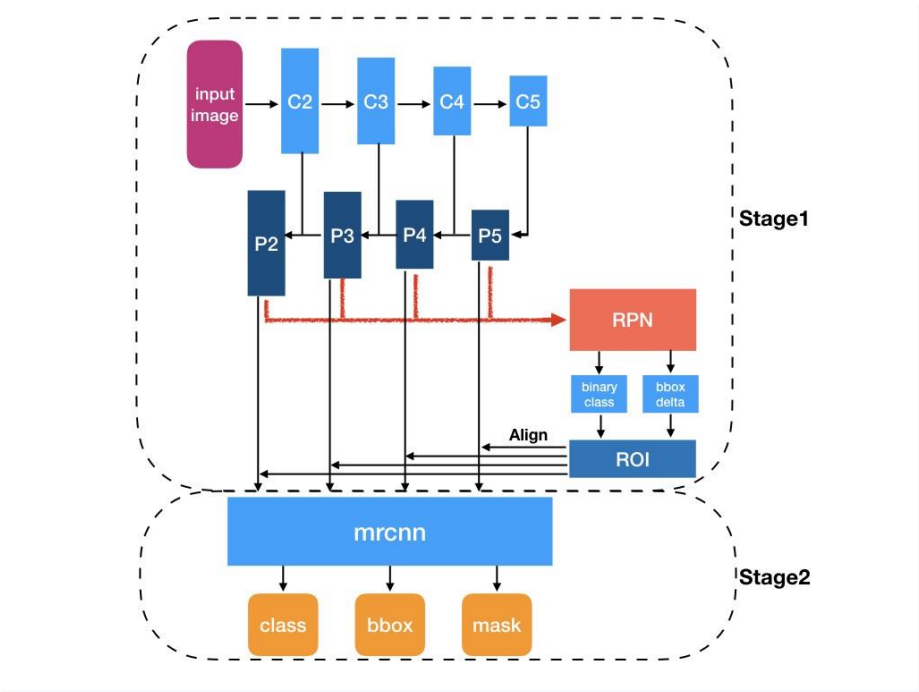


*Figure 6: Architecture of Mask-RCNN*

Unlike other object detection methods, Mask-RCNN objects are segmented. In other words, all the pixels of the object are detected. This event is called image segmentation. Mask-RCNN is basically very similar to Faster-RCNN. Faster-RCNN is used in this model because it is successful in object detection problems. Mask-RCNN *[9]* can be examined in two steps. First step convolution is applied to image, such as Faster-RCNN. Region proposals are determined with the help of RPN from the feature map obtained in the result. Then, the boundaries of the object are determined by determining whether an object exists in these regions. The first step ends in this way. In the second step, the objects detected from the feature map are given to FCNN (Full convolutional neural networks). Here, a binary matrix is obtained as output. In this binary matrix 0's represent the pixels without objects, 1's represent the areas where the object is located, that is masked regions. Masking is done in this way.

## *3.3 Tools*

### *3.3.1 YOLO Object Detection*

Yolo is a real time object detection system implemented on Darknet. Other systems repurpose classifiers or localizers top reform detection. YOLO *[10]* pipeline is a single network. So a single neural network predicts class probabilities and bounding boxes directly from full images in one evaluation. This architecture works pretty fast. The basic YOLO model processes 45 frames per second in real time. Fast-YOLO, which is a smaller network, processes 155 frames per second. It evaluates the model with PASCAL VOC detection dataset. Compared to other detection systems, YOLO makes more localization errors. The systems give some scores to regions and then classify the detected objects. The YOLO model has some differences form other systems.*[4]*

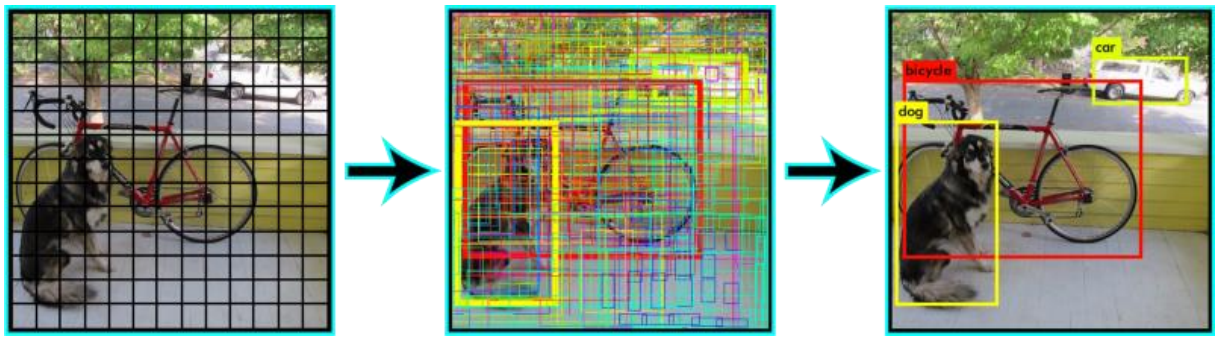


*Figure 7: The technique of YOLO*

As it seems in figure 6, it divides the image into NxN grids and creates M bounding boxes. Then it makes prediction for each bounding box. In other words, It makes NxNxM predictions in total and detect the necessary objects. The most of these unreliable boxes.

### *3.3.2 Tensorflow Object Detection API*

In computer vision, it is not an easy task to develop machine learning models that can localize and identify multiple objects in an image. The Tensorflow Object Detection API *[11]* is an open-source Google-supported tool for creating, training and distributing object-aware models.

In addition to this, Tensorflow also has support for different platforms and it's call Tensorflow lite *[12]*. It is a tool which to help run models on IoT, embedded mobile devices. It makes the models low latency and a small binary size.

# 4 Timeline

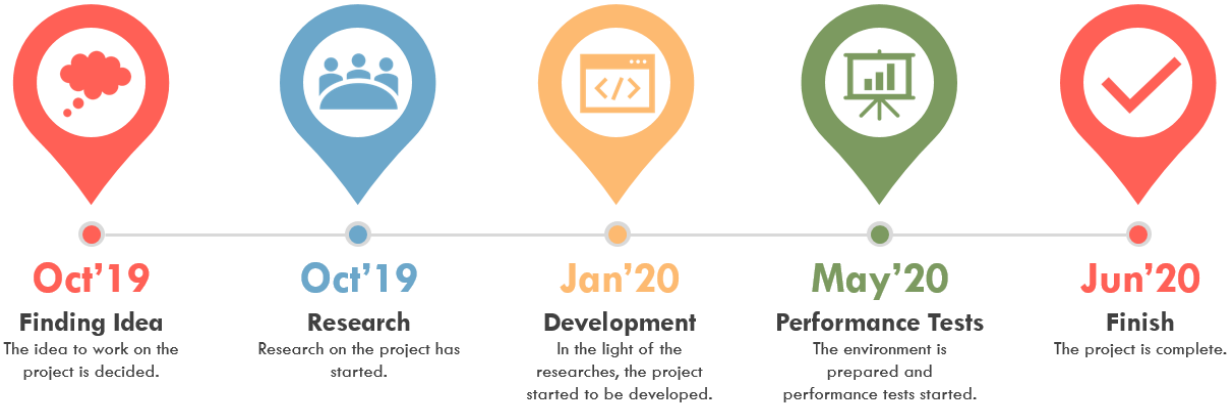


*Figure 8: Timeline*

# 5 Risk Analysis

Risk analysis is the process of identifying and analyzing potential problems that are thought to affect projects negatively. Thanks to this process, measures against problems that may occur will be taken earlier.

| **Factor** | **Risk** |
|---|---|
| The waste to be separated is inside a closed object such as a plastic bag. | This situation reduces the success of the model by making it difficult to predict the objects to be detected. |

*Table 1: Risks Analysis*

# 6 Related Works

There are many algorithms for classifying images. These are algorithms like SVM, RNN, CNN. Of these, Convolutional Neural Network (CNN) is the most performing and well-functioning. For a long time, many different CNN architectures have been developed for image classification and object detection problems. In this way, a lot of automatic classification projects were done.

There is a study *[13]* for garbage detection and collection at Vishwakarma Government Engineering College. In this study *[similar works 1]*, when garbage is detected, it determines the location of the garbage by using only the camera. It then determines whether the detected object is valuables or garbage. It does this with 95% accuracy rate. Finally, the garbage detected by a micro controller arm is put into the trash. The system processes approximately 3-4 frames per second on Raspberry pi.

Ying Liu has published a paper *[14]* about automatic garbage detection. Urban decoration garbage detection is done manually with the images from the cameras. In the study carried out to automate this mechanism, garbage accumulated in the field of view of the camera will be detected. The garbage which detected will inform the background processing center in real time with the help of a GPS which is the Internet of Things module. Lightweight Yolov2 object detection model was used in this study. The average success is 89%.

On another paper *[15]* about Waste Classification and Recycling, it *[similar Works3]* proposes a multilayer hybrid deep-learning system (MHS) to automatically separate waste. Wastes are classified as paper, plastic, glass, metal, fruit / vegetable, plant and others. This study achieves a good accuracy with MHS higher than 90%.

In a study *[16]* conducted by Mind Yang and Gary Thung, it was aimed to make a method that can automatically classify and sort the garbage. The main target here is to increase efficiency in recycling facilities. Since there was no open source data set at that time, a data set was created with images collected from Flickr Material Database and Google images and used in the project. For models, SVM (support vector machine) and CNN (convolutional neural networks) classification methods were deemed appropriate and preferred. The images used in the dataset are reduced to 384x384 size. A structure similar to Alexnet was established for CNN and this model was trained in this way. In the test set 63% accuracy achieved with SVM method. In CNN, proper learning did not occur and the model was only able to achieve 22% success in the test set. As a result, it was concluded that garbage can be classified with machine learning methods.

In a study *[17]* conducted by Umut Özkaya and Levent Seyfi, it was aimed to develop an algorithm for classifying garbage. Deep networks are used in this study. SVM and Softmax methods were used for the classification process in the last layer of deep networks. ThrashNet, which was published by Gary Thung publicly and also used for this project, was used as the data set. Among the different fine-tuned CNN structures trained, GoogleNet achieved 97.86% accuracy with SVM method, giving the most successful results. Studies have shown that garbage classification can be done on machine learning methods, higher success can be achieved with more data and SVM method is more successful than Softmax.

In another study *[18]* on waste disposal, it was stated that by recycling garbage should do at an earlier stage. In this way recycling rate will increase and pollution will reduce. The image classification methods here are examined in four different categories. These are classification based on shape, based on reflectance properties, based on materials and using CNN. The improvements provided by these methods are mentioned. As a result, it is concluded that these classification methods will not be as successful as the test results in real life. For images which used for garbage classification, the classification process is done on a single object, but in real life these methods will fail because garbage is in more complex forms.

# 7 Implementation

## *7.1 Model Training*

In this project, Tensorflow Object Detection API is used to train the models. The detection API has a pipeline as followed for an SSD model training:

First of all, feature extraction is done. The first part of the entire network is a convolutional block from another network. VGG, Inception and so on are used for these blocks. The outputs here are fed to the detection section. The next part is the detection part. This section contains a series of detection blocks. Each block has its own output, which contributes to the network's final output. The size of the feature maps are reduced after each block. In this way, objects of all scales be captured. Each detection block has 3 branches. These are box generation, classification and localization correction. The first one is cropping the boxes on the feature map. The second one is responsible for predicting the confidence scores of the classes. The third concerns the settings of their positions for the boxes produced. Outputs of the detection blocks are collected in the non-maxima suppression layer. The loss value used for training is the sum of the Classification and Localization loss values.

## *7.2 Scripts*

### *7.2.1 train.py*

The train.py script starts the training process. It contains the configurations to be used in this process. By taking the configuration files of the models, it provides model training with the train and validation data in Tensorflow's record files. The weights learned in this process are recorded at regular intervals as they are trained. Saves ckpt files as output.

### *7.2.2 detect.py*

The detect.py script takes the trained model graphs and camera frames as an input. After that, it classifies the objects which detected by the trained model to each frame. It returns classes, accuracy score and the boundary boxes to the user by drawing on the frame.

## *7.3 Detector*

The contents of software written in the Python programming language which is one of the most well-known language utilized for artificial intelligence errands. Tensorflow deep learning framework used to train models and the Open-cv library used for some video processing tasks. Four different models were trained using two different model architectures for the detector. Unlike these, only one type model can work on Raspberry pi *[19]* in good performance. This model type is called TFLite. The complexity of the model is lower than others, which means it is lighter. In this way, it can work comfortably and quickly on machines with low processing power.

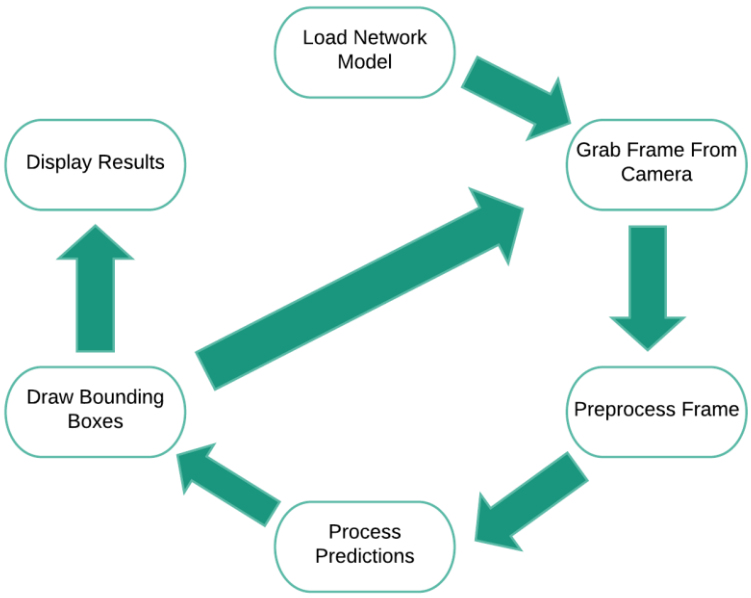


*Figure 9: Detector Pipeline*

The pipeline of the detector is as shown in the figure. Models with different architectures that were previously trained are called for use. The way of running each of the architectures is different. Then, thanks to the Open-cv library, frames are captured from the camera. Frames taken from here are made suitable for processing, this process is called preprocessing. In addition, different image processing techniques can be applied to the images in the pre-processing part to make the image cleaner. When the preprocessing part is completed, it is provided to trained models to detect the objects. Objects with a high score are estimated from the objects detected here, thanks to the threshold value. As a last step, the classes of the objects on the threshold score and detected boundary boxes are drawn on the output image. This final image is shows to the user as output. At the same time, the output values which are the predicted class and score values, are provided to the user as objects.

## *7.4 Dataset*

Thrash Net *[20]* dataset was used to train the model. The dataset contains many different garbage images. This dataset, which consists of 6 different classes but in this project 4 different classes used which is divided into classes like metal, plastic, paper and glass. It contains 1987 images.

In addition to the Thrash Net dataset, the dataset used in project contains 2,870 images in total for these four classes, together with images taken by the project team. In other words, approximately 45% data increase was made. These images added later include wastes that we use in daily life and that can be separated. These images were taken in many different environments such as restaurants, cafes, social areas and homes. Dataset contains many different objects from different angles and in different ways. Images are resized and reduced to 512x384. The images are resized because working with high resolution images means a lot more effort for machines. Images were taken using the Apple iPhone 6, Apple iPhone7 and Nokia-3.2 Smartphone. As an extra, Trash Net is a classification dataset. The images in this So are not tagged. Trash Net and all the images added afterwards were tagged one by one.

The content of the dataset containing 2870 images in total is divided as follows:

- 685 glass
- 602 metal
- 801 paper
- 783 plastic

# 8 Results

As a result of the works performed in this project, 6 different models were trained. The models are taken form Tensorflow detection model zoo models *[21]*.Trained models are pre-trained which previously trained with the COCO *[22]* dataset. As a result of the augmentation in the data set in these models trained with 3 different model types, the accuracy of the objects determined on 200 test images was measured and the results were determined.

The test images used have a resolution of 512 * 384. These results are shown in the table below.

| **Model** | **Dataset** | **Accuracy** | **FPS** |
|---|---|---|---|
| ssd_mobilenet_v1_fpn_coco | ThrashNet | 81.5 | 10 |
| ssd_mobilenet_v1_fpn_coco | ThrashNet + New Images | **84** | 11 |
| faster_rcnn_inception_v2_coco | ThrashNet | 71.5 | 7 |
| faster_rcnn_inception_v2_coco | ThrashNet + NewImages | 80.5 | 8 |
| ssd_mobilenet_v2_coco | ThrashNet | 74 | **17** |
| ssd_mobilenet_v2_coco | ThrashNet + New Images | 80 | **17** |

*Table 2: Model Performance Benchmark*

According to the benchmark table above, accuracy results of different models were obtained, and the model with the highest accuracy yielded results as the ssd_mobilenet_v1_fpn_coco model trained with an increased data set. Another point is that the speed between models is different. These models have been tested with the Nvidia GTX 1050Ti graphics card. While ssd_mobilenet_v2 process 17 frames per second, ssd_mobilenet_v1 only process 10 frames. There is a serious speed difference about processing time.

These models work quickly on the GPU. The required environment was set up on Raspberry-pi, and Raspberry-pi processed the images at low speed (0.15 FPS) on the CPU as a result of the tests and gave successful results. Apart from these models, one ssd_mobilenet_v2_quantized model was trained to run on Raspberry-pi and it was converted to TFLite format with TFlite converter method. Models in this format can operate at higher speeds on CPU machines. However, this trained model has a very low accuracy rate and it has not been successful.

# 9 Conclusion

As a result, a system was developed that classifies wastes with a high accuracy rate. The latest image processing and deep learning techniques were used in this project. This system detects the garbage in the images and produces an output of which class the garbage belongs according to the input received. The main inference in the project is that the process of sorting garbage will be faster by force of this system developed. In addition the manpower used will be reduced.

Environmental pollution will increase to a great extent without attention and will be inevitable. Therefore, it is very important both to raise people's awareness and to use technology for the benefit of humanity in order to minimize environmental pollution.

Although this project is a stand-alone project, it should not be forgotten to use it as an auxiliary project in the development of other projects. The model trained in this project can be applied to different devices and many different jobs can be done with many different machines that will operate based on the results.

Recycling will be very easy with many other projects where this project can be used. Since object detection and classification is used, it can be used in projects that can be used in many projects from garbage collection to garbage classification, and it can be used in the processing of garbage that is collected and separated in this field. Although Trasholder does not contribute to recycling alone, its effects will be very helpful in recycling.

## *9.1 Future Works*

Due to time constraints and lack of hardware, studies could only be carried out on certain models. More powerful models with stronger hardware should be trained and tested in less time. In these models, improvements should be made on different hyper parameters. Data should be increased with different filters on the processed images, and thus the model can be improved. The images used in the dataset should be increased so that the model can recognize more objects because each of the objects separated in the project has different shapes. Here, creating a larger variety of objects will have a positive effect on the model. Designing a physical waste bin for the model created. The separation of the detected wastes into different compartments according to their classes in the trash can.

# 10 References

[1] Avivasa Dijital Garaj "*Geri Dönüşüm Teknolojilerinde Yapay Zeka Etkisi*", https:// avivasadijitalgaraj.com/geri-d%C3%B6n%C3%BC%C5%9F%C3%BCm-teknolojilerinde-yapay- zeka-etkisi-1f10c405271a, 2019, [Accessed: 2020]

[2] https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python?utm_source=adwords_ppc&utm_campaignid=1455363063&utm_adgroupid=65083631748&utm, [Accessed: 2020]

[3] http://deeplearning.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/, [Accessed: 2020]

[4] https://cs231n.github.io/convolutional-networks/#conv, [Accessed: 2020]

[5] Keiron Teilo O'Shea, Ryan Nash, "An Introduction to Convolutional Neural Networks", 2015

[6] Jianxin Wu, "Convolutional neural networks", pp. 25-28, 2020

[7] Wei Liu1, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector",pp. 2-6, 2016

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", 2016

[9] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, "Mask R-CNN", pp. 3-5, 2018

[10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", 2016

[11] GitHub - Tensorflow detection API, https://github.com/tensorflow/models/tree/master/research/object_detection, [Accessed: 2020]

[12] https://www.tensorflow.org/lite/guide, [Accessed: 2020]

[13] Siddhant Bansal, Seema Patel, Ishita Shah, Prof. Alpesh Patel, Prof. Jagruti Makwana, Dr. Rajesh Thakker, "AGDC: Automatic Garbage Detection and Collection", 2019

[14] Ying Liu, "Research on Automatic Garbage Detection System Based on Deep Learning and Narrowband Internet of Things", 2018

[15] Yinghao Chu, Chen Huang, Xiaodan Xie, Bohai Tan, Shyam Kamal, Xiaogang Xiong, "Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling", 2018

[16] Mindy Yang, Gary Thung, "Classification of Trash for Recyclability Status", 2016

[17] Umut Özkaya, Levent Seyfi, "Fine-Tuning Models Comparisons on Garbage Classification for Recyclability", 2019

[18] AdhithyaPrasanna .M, S. Vikash Kaushal, P. Mahalakshmi, "Survey on identification and classification of waste for efficient disposal and recycling", 2018

[19] Adam Gunnarsson , "Real time object detection on a Raspberry Pi", pp. 10-16, 2019

[20] Github – TrashNet, https://github.com/garythung/trashnet, [Accessed 2020]

[21] GitHub - Tensorflow detection model zoo, https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md#coco-trained-models, [Accessed 2020]

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollar, "Microsoft COCO: Common Objects in Context",2015