



### Android Automatic object detection

---

By Ohad Zadok

#### Introduction

---

##### Motivation:

---

Automatic Object Detection can be used to detect objects in a photo and classify them.

##### Daily usages:

1. Security usage, when trying to detect enemy units in a satellite images or weapons in security cameras there is a need to identify a final amount of objects in an image.
2. Context of photos in Social networks, when uploading a photo to a social network the app can try to determine the context in which the photo was taken.

##### Project Goal:

---

Successfully identify objects from a given set after training a model.

#### Approach and Method

---

To successfully be able to identify objects in a given photo I decided to use a machine learning algorithm named SVM. The usage of SVM requires treating the image as a vector of features, or descriptors. A training set of the objects to be identified had to be created before the predicting process in order to train the model. Each Item was shot several times from different positions and angles in order to create a good variety of images. Each photo of an item was processed and a Vector of descriptors was extracted from it. The descriptors were mainly color related such as histograms, standard deviations and means for each color channel including b/w. after the set has been created the model was trained and a prediction was made on a new photo of an item by creating the same vector of features and applying the model on it. A preprocessing step was made by applying the GrabCut algorithm in order to prevent background noise from affecting the features.

##### Method:

---

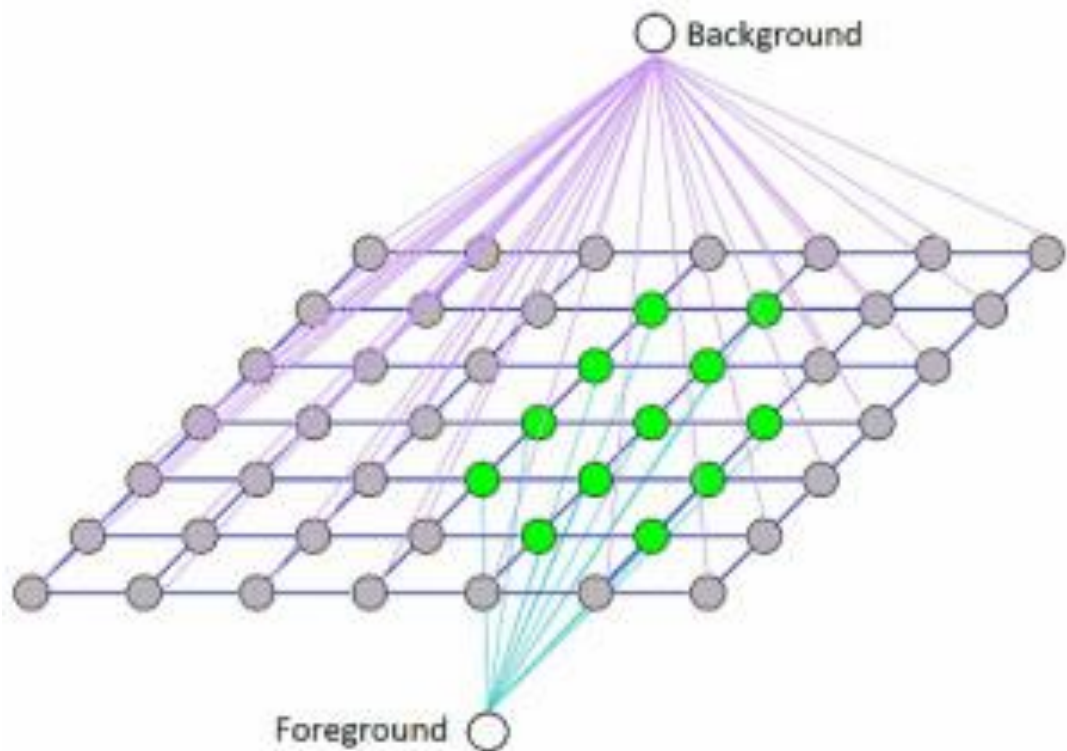
1. Training
  - a. Create new item
  - b. Take photos of the item, for each photo extract Vector of descriptors and save them.
  - c. Vector of features:
    - i. Cut the main part of the image using GrabCut algorithm.
    - ii. Transfer the image to matrix.
    - iii. Calculate mean and standard deviations of each color channel.
    - iv. Calculate Histograms for 4 levels.
    - v. Calculate Features using opencv feature detector and log the number of features in the image.
  - d. Create an SVM model using the vectors of photos of the labeled data.
2. Predicting
  - a. Take a photo.
  - b. Create a vector of descriptors similar to the one at step 1.
  - c. Run the SVM model on the given vector.
  - d. Display the result.



### Algorithms

#### GrabCut

GrabCut is a segmentation technique that uses region and boundary information contained in an image to perform segmentation. GrabCut is using graphs to store region and boundary information. The information is used to create an energy function which when minimized gives the best segmentation result. The graphs are built in a way where nodes represent pixels in the image. Two special nodes are added – the Sink and the Source nodes. The source node represents the foreground of the image and the sink node represents the background. The energy function adds weights to the graph between every two pixels and the sink and source nodes. The weights information is given by the edge information of the image, a strong indication of an edge meaning a small weight and vice versa. A Min Cut algorithm is then applied on the graph and then the source and the sink are being separated, all pixels which are connected to the sink become the background and the ones which are connected to the source become the foreground.



i



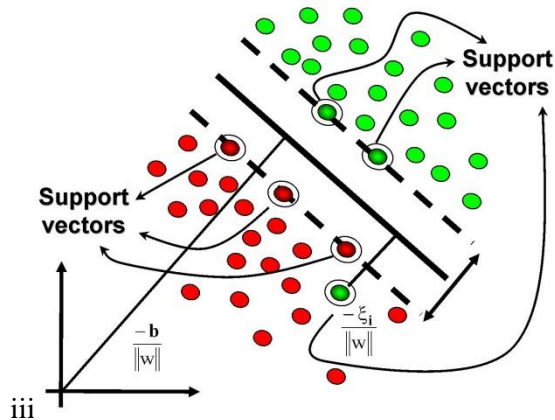
Running example:





### SVM<sup>ii</sup>

Support vector machine is a supervised learning algorithm used in machine learning that analyze data and recognize patterns. Given a set of training examples marked as belonging to a certain category the algorithms built a model that assigns new observation into a category. The model is a representation of the examples as points in space such that there is a clear gap between the categories.





### Implementation

---

The entire project was built under the Android System and is fitted to an android device. The photos were taken using the device's native camera. A vector of descriptors was then calculated using opencv extensions and stored in memory using the serializable feature. An SVM model was created using LibSVM library. The Entire project is run on a device with no need for external resources.

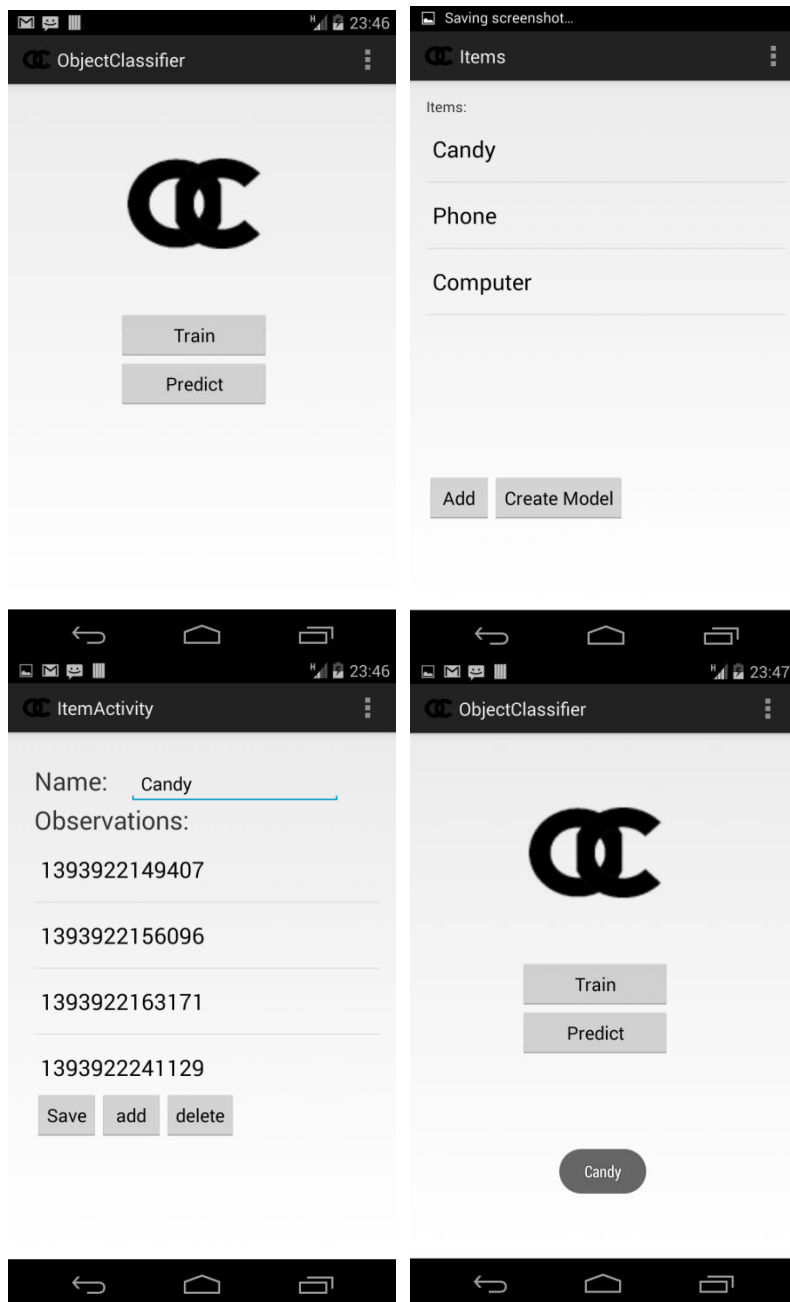
### Conclusion

---

When applying the system to relatively small number of training sets the results are pretty accurate. When we apply it to a bigger set of objects it become unreliable, Future versions of the system can try to use different descriptors such as contours and get even more reliable results.



### Screenshots



<sup>i</sup> <http://grabcut.weebly.com/uploads/1/3/7/5/13757421/2018613.jpg?1351369992>

<sup>ii</sup> <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

<sup>iii</sup> [https://en.wikipedia.org/wiki/Support\\_vector\\_machine](https://en.wikipedia.org/wiki/Support_vector_machine)  
<http://www.cs.ru.ac.za/research/g02m1682/>