SpotGarbage: Smartphone App to Detect Garbage Using Deep Learning

Gaurav Mittal, Kaushal B. Yagnik, Mohit Garg, and Narayanan C. Krishnan*
Indian Institute of Technology Ropar
Rupnagar, India
{gauravmi, yagnikkb, mohitg, ckn}@iitrpr.ac.in

ABSTRACT
Maintaining a clean and hygienic civic environment is an indispensable yet formidable task, especially in developing countries. With the aim of engaging citizens to track and report on their neighborhoods, this paper presents a novel smartphone app, called SpotGarbage, which detects and coarsely segments garbage regions in a user-clicked geo-tagged image. The app utilizes the proposed deep architecture of fully convolutional networks for detecting garbage in images. The model has been trained on a newly introduced Garbage In Images (GINI) dataset, achieving a mean accuracy of 87.69%. The paper also proposes optimizations in the network architecture resulting in a reduction of 87.9% in memory usage and 96.8% in prediction time with no loss in accuracy, facilitating its usage in resource constrained smartphones.

ACM Classification Keywords
I.5.4. Pattern Recognition: Applications—Computer Vision; I.5.1. Pattern Recognition: Models—Neural nets; I.2.1. Artificial Intelligence: Applications and Expert Systems

Author Keywords
Garbage Detection; Deep Learning; Computer Vision; Fully Convolutional Neural Networks; Smartphone; Android

INTRODUCTION
The city landscape of nations such as India is witnessing street corners and pavements transforming into garbage dumps. In addition to being an eye sore, this situation can lead to health hazards if not cleared on time. The issue worsens due to high population density and lack of awareness among the masses making it challenging for authorities to keep track of areas with garbage. A promising solution to check garbage is to engage the citizens by providing them an easy-to-access, prompt and reliable medium through which they can report the presence of garbage in their vicinity to the authorities. The recent proliferation of smartphones makes it an ideal platform for any solution to reach the masses. Moreover, most modern day smartphones are well equipped with reasonably powerful cameras. Considering the visually striking characteristics of garbage, it is intuitive and practical to perceive its detection based on images.

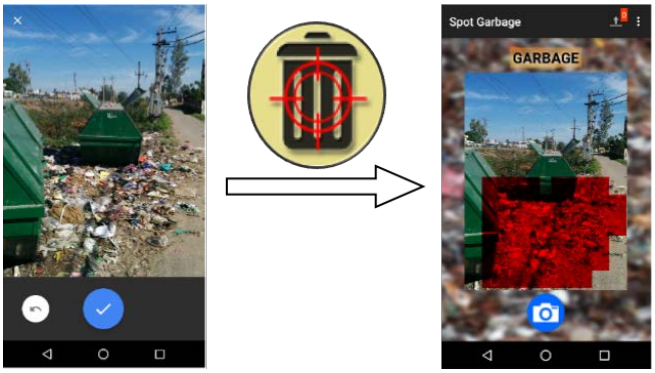

Figure 1. Objective of the SpotGarbage App - Left input image showing garbage, Right - coarsely segmented output image from the app

Motivation
In the past, there have been mobile app based initiatives that allow people to report such menace in their neighborhood by uploading images of garbage [21]. However, these solutions rely on humans for identifying garbage in an image making them impractical for large scale use. These apps have also not been successful due to a large number of spurious uploads (such as images of people and selfies) [26], requiring manual verification. Thus, a promising solution should automatically and reliably detect the presence of garbage in the image minimizing human intervention.

The simplest way to implement such a solution would require the app to upload every user-clicked image to a server for automatic garbage detection. However, the bottleneck of this approach is the slow and at times, erratic network connectivity, and people's mindset of frugal use of internet data plans on their smartphones [27]. Thus, the solution should try to avoid uploading every image, and rather process them on the phone itself. It should send minimal information such as GPS-coordinates, severity of garbage, and optionally, a segmented region of the image containing the garbage over the

*corresponding author

UbiComp '16, September 12-16, 2016, Heidelberg, Germany
 ISBN 978-1-4503-4461-6/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2971648.2971731

network. Processing the image on the phone also helps to elicit user-feedback for validating the machine learning model for garbage detection. Moreover, segmenting the garbage region in the image may allow determining the severity of garbage.

This paper takes the first step towards such a solution by introducing an Android app, SpotGarbage, which employs a Convolutional Neural Network (CNN) called GarbNet to automatically detect and localize garbage in unconstrained real-world images as illustrated in Figure 1. The input to the CNN is an arbitrary sized image and the output is a coarse-grained segmentation of the image highlighting garbage patches. Further, GarbNet is optimized to perform in a resource constrained environment. This facilitates its deployment on a ubiquitous mobile platform. This is the first time that the garbage detection in images is being dealt with using state-of-the-art deep learning and computer vision techniques. Overall, this paper makes the following contributions:

- The paper introduces a new annotated dataset, called Garbage In Images (GINI). The dataset is a collection of several in-the-wild images containing garbage. Each image is also annotated with perceived levels of severity and biodegradability.
- A fully convolutional architecture, GarbNet, trained on the GINI dataset to classify and detect garbage in images with high sensitivity and specificity.
- An android app, SpotGarbage, that deploys the optimized version of GarbNet to determine the presence of garbage in the user-clicked geo-tagged images in near real-time.

RELATED WORK

Although no prior literature exists that describes the task of garbage detection from images, it is possible to relate the task to other object recognition tasks in computer vision. This is due to the intrinsic nature of garbage being partially analogous to an object. Prior approaches for object recognition rely on hand crafted image descriptors for characterizing the object. The popular set of image features include histogram of oriented gradients (HOG) [6, 9], scale invariant feature transform (SIFT) [16, 24], Gabor filters [12], Gabor wavelets [17] and Fischer Kernels [19]. Over the recent years, deep learning techniques have become popular in computer vision due to their ability to automatically learn a hierarchy of rich feature representations directly from pixels intensities [4]. Convolutional Neural Networks (CNNs) have been pivotal in giving state-of-the-art performance for image recognition on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [13, 23, 18, 22, 10], scene labeling [8, 20], semantic segmentation [15], and material classification [3].

The ability to train/test large-scale CNNs has been possible due to the highly efficient parallel processing CUDA framework on GPUs. However, the limited performance of CUDA based CNN on general purpose CPUs restricts their application in computational resource constrained ubiquitous devices such as smartphones. Though many optimization procedures have been proposed to reduce the space and time complexity of CNNs [5, 7], it still remains an open challenge to utilize their full potential to offer practical solutions.

Figure 2. Garbage in Images (GINI) dataset. (a) depicts the annotation collection web portal and (b) highlights few instances of the four categories in the dataset.

GARBAGE IN IMAGES (GINI) DATASET

Training a CNN for detecting garbage requires a large image dataset with garbage related annotations. However, there are no garbage image datasets currently available. The first goal was to gather a diverse set of images that can be used to train a robust CNN for detecting garbage in images. Bing Image Search API was used to crawl the web for garbage and non-garbage related images. Queries (such as road side garbage, market waste) were used to obtain a diverse set of images containing garbage. This resulted in a compilation of 2561 images, out of which 956 images were obtained through garbage related queries.[1]

The images obtained from the garbage related queries could not be directly used for training the CNN for two reasons. Firstly, regions of the image containing garbage (refer to the set of images in the bottom left of Figure 2 (b)) had to be extracted for training the CNN. This is because using the entire image as an example of garbage, when only a portion of it corresponds to garbage, can affect the training process. Secondly, some of the garbage related queries resulted in images that did not contain garbage, but were 'figuratively' related to garbage as illustrated in the set of images on the top right side of Figure 2 (b). These images cannot be used for training the CNN with the label as garbage.

[1] http://cse.iitrpr.ac.in/ckn/Resources/spotgarbage.zip

Figure 3. Different types of garbage in comparison to well-defined objects and dull background.

A web based platform as shown in Figure 2 (a) was developed to obtain user annotations for these images. The annotator was presented with a sequence of images to be labeled as containing garbage or not. The portal also allowed the annotator to draw a bounding box to mark the region containing garbage. Further, if the image contained garbage, the annotator also reported his/her perceived level of garbage severity and biodegradability. Subsequently, a total of 1494 annotations were collected from 83 users belonging to the age group of 18–21 years for 534 images out of which 450 were used for the experiments conducted in this paper. The inter-rater reliability as measured using Cohen's Kappa was 0.615.

METHODOLOGY

The fundamental challenge in automatic garbage detection is to define garbage unambiguously. There are two ways in which garbage can be perceived in images. The first approach deals with the detection of individual objects whose existence is anomalous to the background in which they are present, such as plastic bottles and tin cans in a lush green landscape as illustrated in the bottom right image in Figure 3. Whereas the second approach considers garbage as a single entity being a conglomeration of indistinct garbled and/or decayed objects as illustrated in the bottom left image of Figure 3. Similar to objects, garbage has its own salient characteristics [1, 2], but unlike regular objects, exhibits an innate randomness and is not well-defined. Moreover, garbage possesses the amorphous trait of the background but differs in that it is rich in randomly occurring features; including but not limited to extreme variations in texture, color, edges, shape and size. Further, since garbage usually comprises of various materials such as plastic, paper, metal, ceramic and food, it cannot be regarded as a single material [3]. The approach discussed in this paper targets the detection of the latter kind of garbage that exhibits an amorphous trait.

The overarching goal of this paper is to detect the presence of garbage in an image and also approximately demarcate the regions in the image that correspond to garbage. This goal is achieved by training a model using patches extracted from images. The final prediction for a test image is obtained by again extracting patches and combining their predictions.

Patch Generation

The images from the GINI dataset are processed to generate fixed-sized patches. Garbage images that had regions not entirely garbage (illustrated by the partial garbage images in Figure 2 (b)) are excluded from this process to avoid ambiguity while learning the distinguishing characteristics for garbage. The rest of the images are first divided into 5 stratified folds to avoid correlation among the patches across the folds. The images in each fold are cropped to generate patches of different sizes, allowing the model to adapt to multiple scales and different levels of contextual information. The patch sizes were chosen to be 10%, 20%, 40% and 80% with stride as 9.1% of the smaller image dimension so as to perform Poisson-disk image subsampling [3]. The patches are further oversampled by performing random rotations between $[0, 2\pi]$. This increases the size of the training set, which helps to prevent overfitting. More importantly, it also makes the model rotation invariant. In total, this generates a set of 500,000 patches equally divided between the garbage and non-garbage classes.

GarbNet Model

The objectification of garbage allows the weights of GarbNet to be initialized using the pre-trained model, AlexNet [13], which has been trained on 1 million images for 1000-way object recognition. By doing so, GarbNet is able to exploit the rich hierarchy of already learned representations making it achieve a better generalization. The pretrained AlexNet model, used to initialize GarbNet, is an open source implementation from Caffe Model Zoo. The architecture has been modified to perform binary classification. The two fully connected layers of the network, following the five convolution layers and containing 4096 neurons each, are optimized to have 512 and 256 neurons respectively [14].[2] The supervised fine-tuning of GarbNet is conducted via 5-fold stratified cross validation with training and validation sets each consisting of approximately 380,000 and 20,000 patches respectively. The model is trained on Caffe using Nvidia TitanX for 150,000 iterations with batch size of 100 and validation is performed after every 5,000 iterations. The initial learning rate of 1×10^{-3} is reduced by a factor of 4 after every 25,000 iterations. The momentum is 0.9 and weight decay is 5×10^{-5}. Further, the patch samples are randomly mirrored and cropped while training to prevent overfitting.

Optimizing the GarbNet Model

It is imperative for the final model to yield a quick response and have a low memory footprint to be deployed on smartphones. Optimizing the fully connected layers to have less number of neurons results in a drastic reduction of 87.9% in size and number of model parameters, lowering the space requirements for operating the app.

Using a regular CNN with naïve sliding window approach results in a lot of redundant computation due to the overlapping receptive fields [15]. The convolutional layers of CNN are translation invariant. They operate on local input regions and are agnostic to the spatial size. This characteristic can be exploited to expedite the feedforward computation, by allowing

[2]The architecture of final deployed GarbNet model is provided in the supplementary material, along with details of the GINI dataset.

the entire image to be processed in a single pass instead of giving individual overlapping patches as input to the network. This is achieved by converting the fully connected layers into convolutional layers making the architecture convolve with the entire image in one go. For instance, the output of the final convolutional layer for an input image of size 227×227 is a set of 256 feature maps of size 6×6. So the weights in the following fully connected layer are rearranged to form 512 kernels of filter size 6×6 with a modifiable stride. This transforms the regular CNN into a fully convolutional network (FCN) without any change in the model size. The computation gets highly amortized reducing the time for FCN over the naïve sliding window approach by a factor of 12.

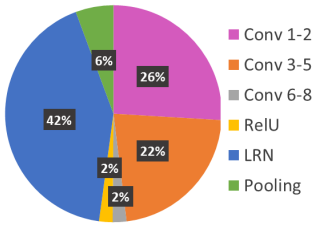

Figure 4. Breakup of the computational time on CPU for different components of GarbNet model using FCN with LRN

The FCN based GarbNet model produces a classification map for all the patches in a test image simultaneously instead of a single classification and returns positive if at least one image patch is classified as garbage. The size and overlap (stride) are varied to determine their optimal values.

The AlexNet model performs Local Response Normalization (LRN) after each of the first two convolutional layers. From Figure 4, it is evident that the LRN layers take approximately 42% of the total prediction time. The basis of incorporating a LRN layer in the Alex network is empirical. Therefore, another optimization step would be to remove these layers reducing the prediction time. Experiments are conducted to rule out adverse effects of removing the LRN layer from the network on the model accuracy.

Image Processing

There are no previously available results for detecting garbage in an image to benchmark the proposed method. A back propagation network that is trained using a large number of features extracted from the patches is used as the baseline. The feature extractors used are the state-of-the-art image descriptors commonly found in the literature. The patches are first warped to a fixed dimension of 256×256 to extract feature vectors of constant size. The warped patches are convolved with a filter bank of Gabor wavelets having 4 wavelengths and 4 orientations [11]. The result is smoothened with a Gaussian filter with width equal to half the wavelength of the corresponding filter. This is followed by an application of PCA on each pixel to extract the direction with the maximum variance. The output is downsampled to produce $16,384$ features. Further, HOG features are extracted using a cell size of 16×16 taking the maximum of the gradient magnitude across the 3 color channels, thus generating another set of $7,936$ features. Finally, the histograms corresponding to RGB, HSV and Lab color spaces, with 25 bins for each channel, for every 50% overlapping 64×64 subsample is appended to generate a feature vector of size $35,345$ in total. Thus, the feature vector provides an extensive characterization of an image patch. The learning rate and number of hidden layer nodes of the back propagation network are fine-tuned using cross validation experiments.

RESULTS AND DISCUSSION

The patch size and overlap parameters of the GarbNet model are set based on 5-fold cross validation experiments. The probability of classifying a patch as garbage is set to 0.99 to ensure maximum confidence in prediction and minimum false positives. The optimal patch size and overlap was determined to be 25% and 6.82% of the smaller image dimension respectively.

Model	Accuracy (%)	Sensitivity (%)	Specificity (%)	Time[3] (s)
HOG + Gabor + Color	80.32 ± 1.29	81.34 ± 5.38	79.19 ± 6.74	24.27
Sliding Window CNN	87.21 ± 1.53	81.46 ± 5.28	90.95 ± 1.28	48.38
GarbNet (FCN with LRN)	87.70 ± 1.67	83.41 ± 4.14	90.53 ± 1.22	4.11
GarbNet (FCN without LRN)	87.69 ± 0.93	83.96 ± 4.55	90.06 ± 1.94	1.50

Table 1. Comparison of different approaches to detect garbage.

Table 5 reports the performance of the different techniques for classifying individual patches in an image. It is evident that deep learning based methods considerably outperform the approach relying on image processing giving around 7% increase in accuracy and 11% in specificity. Moreover, it can be observed that GarbNet with LRN is able to perform the prediction 11 times faster than naïve sliding window based CNN without any reduction in the accuracy. The method also predicts 6 times faster in comparison to traditional image processing. Furthermore, removal of the normalization layers from the model results in the prediction time to drop by almost 63.5% and 96.8% in comparison to using FCN with LRN and sliding window CNN respectively without affecting the accuracy of the prediction.

A representative sample of predictions made by the GarbNet is shown in Figure 5. These images clearly demonstrate that the model is able to clearly delineate the garbage region from the rest of the image. The predictions are accurate in a scenario where a large heap of garbage/debris is present in the image. The model is also able to detect multiple swathes of garbage in an image. Moreover, the correctly classified non-garbage images illustrate the robustness of the model in distinguishing garbage from well-defined objects, amorphous background, highly varying textures and elements which are just contextually similar to garbage. At the same time, the misclassified images highlight some of the shortcomings of the model. It can be observed that GarbNet fails to detect garbage when it is scantily available in the image. Further, the model misclassifies an image as garbage when the objects in the image

[3]CPU prediction time as computed on Xeon® Workstation having Intel® Core™ i7-5930K processor with 32 GB RAM.

Figure 5. Prediction of GarbNet on various images of GINI dataset

either possess a randomness similar to garbage or lose their well-defined characteristics when located far away.

It is also observed that changing the number of parameters in the two fully connected layers as compared to the original AlexNet model reduced the memory usage of the model by bringing the size from approximately 233 MB down to 28 MB. This facilitates the deployment of the model on a wide spectrum of low-end smartphones and other IoT devices where memory space is a premium.

The experiments suggest the superiority of the GarbNet model optimized using FCN and without the LRN layers. This model is the fastest, has low memory footprint and yields good performance. As a result, this model is deployed in the smartphone application.

SPOTGARBAGE MOBILE APP

SpotGarbage[4], the mobile application that deploys the optimized GarbNet has been developed for the Android platform. Figure 6 depicts the work flow of the application.

The primary component of the app is the optimized GarbNet model which has been deployed by creating Java wrapper methods using Java Native Interface (JNI) over native C++ shared library. The library is developed to implement FCNs by building over the CPU-based functionality provided by the caffe-android-lib GitHub repository [25]. The geo-tagged image captured by the user is fed to the model to generate a coarse segmentation of garbage as shown in Figure 6. The locations of the images classified as garbage are marked and plotted on Google Maps.

SpotGarbage also has the functionality to collect feedback from the user about the correctness of the prediction. This feedback can be used to periodically update the GarbNet model. The app allows the user to choose the frequency with which he/she would like to provide the feedback. The app is also provisioned with an offline cache to store the locations containing garbage as detected by the GarbNet model from user-clicked

[4]The app and other resources to try out are available at https://github.com/KudaP/SpotGarbage

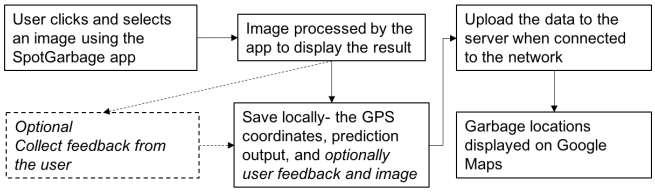

Figure 6. SpotGarbage app workflow. The optional parts of the workflow are highlighted using dashed lines.

images, along with the user feedback information. This prevents data loss when the network connectivity is slow or absent, which is common in developing countries. The GarbNet model processes the images on the phone itself eliminating the need for the images to be sent over the network to a server for processing, thereby reducing network usage.

The app has been successfully tested on the following smartphones— Motorola Moto G (Gen 3), Moto X, Nexus 6 and LG Nexus 5 for Android OS 6.0 (Marshmallow). The app, when tested on LG Nexus 5, gave a mean prediction time of 5.61 seconds while consuming around 67.3 MB of memory with a peak CPU usage of 83%.

SUMMARY AND FUTURE WORK

This paper introduces an Android app, SpotGarbage that can automatically detect and localize regions containing garbage in user-clicked unconstrained geo-tagged real-world images. The app utilizes the proposed fully convolutional network, GarbNet for coarsely segmenting image regions containing garbage. The GarbNet model has been optimized for resource constrained smartphones, by reducing the number of network parameters and removing local response normalization layers from the network. The experiments conducted on the new GINI dataset suggest a significant reduction in the space and time requirements of the application with no considerable effect on the accuracy. The model is able to classify images with an accuracy of 87.69%. While the app is able to classify images in near-real time, further reduction in the prediction time still remains an open challenge.

ACKNOWLEDGMENTS

The authors are grateful to the participants who helped to annotate the images and NVIDIA Corporation for donating the TitanX GPU used for this research.

REFERENCES

1. Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari, “What is an object?”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010.
2. Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. “Measuring the objectness of image windows”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11), 2189-2202, 2012.
3. Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala, “Material recognition in the wild with the materials in context database.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3479-3487, 2015.
4. Yoshua Bengio, “Learning deep architectures for AI”, Journal of Foundations and trends® in Machine Learning, 2(1), 1-127, 2009.
5. Wenlin Chen, James T. Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen, “Compressing neural networks with the hashing trick”, arXiv preprint arXiv:1504.04788, 2015.
6. Navneet Dalal, and Bill Triggs, “Histograms of oriented gradients for human detection”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 886-893, 2005.
7. Emily L Denton, WOjciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fegus, “Exploiting linear structure within convolutional networks for efficient evaluation”, Advances in Neural Information Processing Systems, 2014.
8. Clement Farabet, C Couprie, L Najman, and Yann LeCun, “Learning hierarchical features for scene labeling”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1915-1929, 2013.
9. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, “Object detection with discriminatively trained part-based models”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9), 1627-1645, 2010.
10. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep Residual Learning for Image Recognition”, arXiv preprint arXiv:1512.03385, 2013.
11. Anil K. Jain, and Farshid Farrokhnia, “Unsupervised texture segmentation using Gabor filters”, Proceedings of the IEEE Conference on Systems, Man and Cybernetics, 14-19, 1990.
12. Anil K. Jain, Nalini K. Ratha, and Sridhar Lakshmanan, “Object detection using Gabor filters,”, Pattern Recognition 30(2), 295-309, 1997.
13. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, “Imagenet classification with deep convolutional neural networks”, Advances in Neural Information Processing Systems, 2012.
14. Gil Levi, and Tal Hassner, “Age and gender classification using convolutional neural networks”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 34-42, 2015.
15. Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.
16. David G. Lowe, "Distinctive image features from scale-invariant keypoints”, International Journal of Computer Vision, 60(2) 91-110, 2004.
17. Bangalore S. Manjunath, and Wei-Ying Ma, “Texture features for browsing and retrieval of image data”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18(8), 837-842, 1997.
18. Wanli Ouyang, et al., “Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection”, Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2015.
19. Florent Perronnin, Jorge SÃąnchez, and Thomas Mensink,“Improving the fisher kernel for large-scale image classification”, Proceedings of the European Conference on Computer Vision, 143-156, 2010.
20. Pedro H.O. Pinheiro, and Ronan Collober, “Recurrent convolutional neural networks for scene parsing”, arXiv preprint arXiv:1306.2795, 2013.
21. Mahek Shah, “Swachh Bharat - Clean India Android App”, at `https://goo.gl/BG5KOJ`
22. Karen Simonyan, and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition”, Proceedings of the International Conference on Learning Representations, 2015.
23. Christian Szegedy, et al. “Going deeper with convolutions.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.
24. Jasper R.R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition”, International Journal of computer vision, 104(2), 154-171, 2013.
25. “caffe-android-lib,” at `https://github.com/sh1r0/caffe-android-lib`
26. “Swachchh Delhi App fails to meet expectations,” at `http://goo.gl/VqYDKP`
27. “Frugal innovation and innovation for the frugal” at `http://goo.gl/c7MX6k`