

A Deep Learning-Based Substructure for Trash Detection and Face Recognition

AbigaSansuri.M¹, Anusuya K V², Nivethitha.M³
{abigasansuri@gmail.com¹, kva.ece@psgtech.ac.in², niveee27@gmail.com³}

Associate Professor², Department of ECE, PSG College of Technology, Coimbatore.^{1,3}

Abstract. Traditional Street cleaning procedures demand larger manpower. It necessitates the identification of people who litter on the roadside using scientific approaches. In the proposed system, a deep neural network algorithm named Convolutional Neural Network (CNN), precisely Mask Region Convolutional Neural Network (MRCNN) is used a long with the camera as fixed at the crowded areas such as bus stops and markets. This deep learning method is used in the suggested schema to evaluate the photos of streets and detect litter (if any) in them. The detection of litter further progresses with the identification of the person who has thrown it. This technique uses the built-in library files of Python, to generate and compare face encodings. Moreover, it is involved in assisting with better monitoring and reducing operational costs.

Keywords: Cleanliness, Face Recognition, litter detection, MASK - RCNN, Neural Networks.

1 Introduction

Littering on the roadside is a widespread problem all around the world. It's even been referred to as chronic by some. People are unable to avoid littering on highways, railway lines, floors, water bodies, and other surfaces. Swachh Bharat Abhiyan (Swachh Bharat Abhiyan) is a government of India initiative aimed at making people more accountable for their surroundings and pushing them to clean up after themselves. The process of detecting litter on roadways is currently not automated, and it involves human interaction at practically every level, making it even more time-consuming. Deep learning approaches can be used to automate most of the activities involved in street cleanings, such as litter identification and classification. To accomplish these tasks, an automated approach is proposed in this paper. Using object detection and face recognition techniques, the suggested system attempts to improve a city's environment. As a result, tough rules and measures may be efficiently enforced, making citizens more environmentally conscious.

2 Literature Survey

The mobile processing component of the proposed framework is discussed in the paper [1]. This component's main aim is to process the collected images and assess if they are

worthy of further processing. The collected photos are subjected to further object detection and classification. One of the solutions is to use CNN to detect objects. Due to advances in the field of deep learning, optimized variations of CNN, such as Region Convolutional Neural Network (R-CNN) [2], [3], have been developed. However, the R-CNN-based training model is space and time-intensive, and it performs slow object detection [4]. Fast R-CNN is a more advanced implementation with considerable performance gains. It uses deep neural networks with numerous layers of convolutional and max-pooling layers to deliver object recognition at near real-time rates. To generate predictions, selective search is utilized, which takes a long time. Faster R-CNN overcomes this by replacing the selective search approach with a region proposal network, which significantly speeds up the algorithm [5]. Faster R-CNN, on the other hand, cannot accomplish pixel-to-pixel alignment between network inputs and outputs. The Region of Interest (RoI) Pool is used to do coarse spatial quantization for feature extraction. RoI Align is a simple, quantization-free layer that preserves the exact spatial placements. Mask R-CNN [6] makes advantage of it. Theoretically, Mask R-CNN is simple. Faster RCNN produces a class label and a bounding-box offset. A third branch is also introduced, which outputs the object mask (a binary mask that shows the pixels in the bounding box where the item is). Mask R-CNN is used to extract the spatial layout of an object spatial layout.

3 Proposed Methodology

The proposed schema is constructed and shown in Fig. 1 using the above-mentioned literature. The proposed scheme uses deep learning algorithms on photos of streets recorded with a camera to automate the task of litter detection. The images are recorded every 20 seconds by a pi camera and fed into the processing component, which has the primary objective of processing the captured images and determining their worth for more in-depth analysis. The Object Detection Block (ODB), which employs the Convolutional Neural Network (CNN), particularly the Mask RCNN, receives this information. The ODB is in charge of successfully distinguishing litter from non-litter objects.

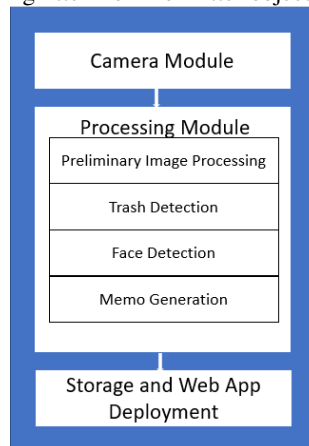


Fig. 1 The proposed system

An individual network with a selective search algorithm [10] is used to generate 2000 region ideas for the input image. To reshape the predicted region proposals, a Non-quantized RoI Align layer is used. Each region proposal image is scaled down to 224x224 pixels. These region proposal images are then sent into the trained CNN, which creates a 2000x4096 dimensional matrix by applying a 4096-dimensional feature vector to all 2000 region proposals.

For each class, SVM (Support Vector Machines) [7] is used to classify each region proposal. For all N classes, the SVM weights(4096-dimensional) are constructed into a matrix and multiplied with the feature matrix. As a result, a matrix is created in which each class to which a region proposal belongs is assigned a score. A request is made for the class with the highest score. As a response, a class label is assigned to each of the 2000 region proposals or bounding boxes in the image. Many of the boundary boxes would be redundant. Therefore, the overlapping bounding boxes must be deleted, which is done via a non-maximum suppression algorithm [9].

The method uses the scores to discriminate between litter and non-litter. The control is moved to the face detection block for trash. Python has built-in functions for generating and comparing face encodings [12]. The algorithm takes into account the color, size, slant of the eyes, and gap between the brows, among other aspects of the face. Face encoding is defined by the combination of attributes and is used to identify a given image (face). In addition, the recognized person's name is sent to the memo generating module in Fig. 5. A CSV file is used to hold a dataset that includes a person's name, date of birth, phone number, email address, and account number. The absolute path is used to read this dataset, which is read using the Panda library. The dataset is stripped of the identifiable person's name, email address, and account number. To connect to the server, an SMTP object is generated. A customized memo is attached to the message object, which is then dispatched. A fine can be deducted using the bank account number, which is outside of our scope. The online application, which can be accessed by the user through a web browser with an active internet connection, displays an image with the bounding box over the garbage and the person who dropped litter, as well as labels.

4 Implementation

The proposed algorithm is implemented using Python script language. The processing component is made up of several program files and is shown in Fig. 2. The function of each program is narrated below.

- Main Program that consists of trash detection and face detection instructions.
- Mask RCNN folder that consists of python files of the base configuration class and the main mask RCNN model implementation. It also contains common utility functions and classes, displays, and visualization functions.
- The training set provided in the trash folder includes all the annotations of JSON files. VGG image annotator [11] was utilized for image annotation.
- A separate folder is made use to record all the images captured by the raspberry pi camera module.
- To train the network, weights are exploited.
- A set of known images are stored in a distinct folder.

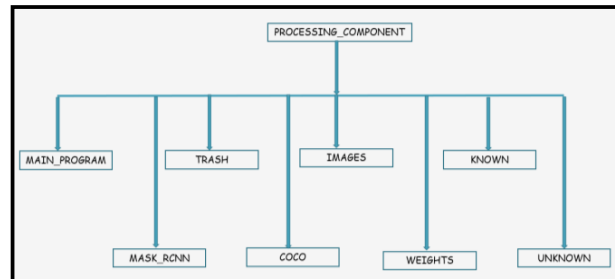


Fig. 2. Processing Component

- *Main Program*

As shown in Fig. 3, all the necessary library files such as Pycocotools, Kaggle, Mrcnn, TensorFlow (1.14.0), Keras, NumPy, Scipy, (2.2.0), Face_recognition, Pillow, TensorFlow, Skimage, h5py, Ipython, Cv2, Matplotlib, and Python are initially installed.

- *Trash detection and Face detection*

All the required weights along with the configuration functions of MASK-RCNN are imported. The number of GPU to work on the processing of the image is set, in addition to the number of the image processed per GPU. The configuration details of the MASK_ RCNN are displayed. The device is set to load the neural network on. The model is inspected in training or inferences mode. The validation dataset is loaded and a model is created in inference mode. On creating, all the trained weights are loaded. Further, the images from the directory (wherein the images are stored) are obtained once it is captured by Raspberry Pi every 20 seconds. Object detection on the images is performed and the results are displayed. If the scores of the result are larger than the threshold, then the images processed by the trash detection block are stored in a folder called unknown. With the help of functions in face recognition, the encodings from the face present in the image are generated. Similarly, encodings for the set of known images are generated. Comparison between the encodings is carried out until a match is found. Eventually, the output is shown with labels and scores. The flowchart illustrating the flow of the face detection process is shown in Fig. 4. The hardware diagram is demonstrated in Fig. 6. The connections to the Raspberry Pi model [8] are depicted in Fig.7.

With the help of functions in face recognition, the encodings from the face present in the image are generated. Similarly, encodings for the set of known images are generated. Comparison between the encodings is carried out until a match is found. Finally, the output is displayed with labels and scores. The flowchart illustrating the flow of the face detection process is shown in Fig. 4. The hardware diagram is demonstrated in Fig. 5. The connections to the Raspberry Pi model [8] are depicted in Fig. 6.

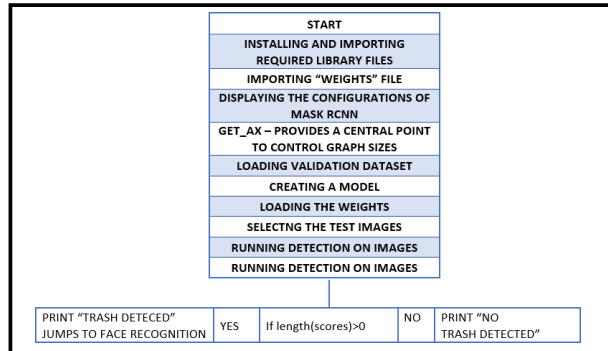


Fig. 3. The flow of the Main Program

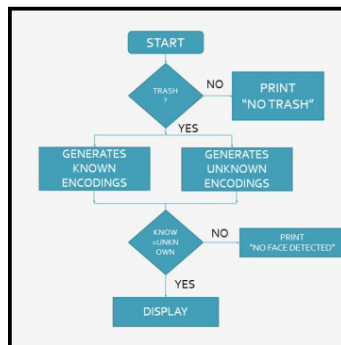


Fig. 4. Flowchart of Face Detection

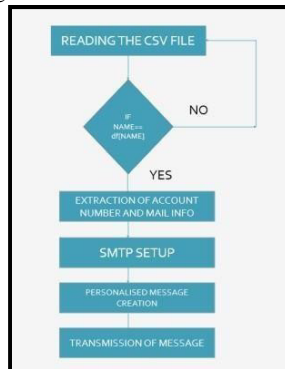


Fig. 5 Memo Generation Block



Fig. 6. Raspberry pi 4 with camera module



Fig. 7. Connections in raspberry pi 4

- *Memo Generation*

SMTP (Simple Mail Transfer Protocol) is used for framing the communication guideline between two servers. User Agent (UA) and Mail Transfer Agent (MTA) constitute SMTP. Message creation, envelope creation, and placement of message into the envelope are carried out by the User-Agent. MTA transfers the mail across the internet. The steps involved to send a mail with the aid of SMTP are discussed below.

 - Import the SMTP and Email packages
 - Create an SMTP object to establish a connection to the SMTP server.
 - Read the dataset. Extract the email and bank account information.
 - User mail ID, App password, Gmail port address, host address, email ID of the identified person and subject along with the body are defined.
 - Email_alert() is used for sending the mail.
- *Deployment*



Fig. 12. Output with trash and face detected



Fig. 13. Output with no trash

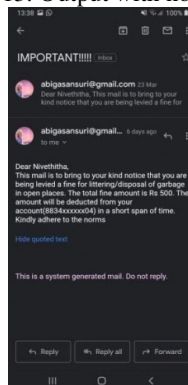


Fig. 14 A sample Memo sent to the person of interest

Conclusion

At all stages of life, cleanliness is essential. It will be easier to develop a clean city without endangering our health if tight laws are enforced. With the use of cameras installed in public places such as bus stops, marketplaces, and other locations, this system will help us detect violators in the shortest possible time while using the least number of manual resources.

References

- [1] ChandniRamchandani, RakshithKoravadiHatwar, ParteekMakkar, Yanki Shah, PoojaYelure, MagdaliniEirinaki, "A deep learning framework for smart street cleaning" in 2017 IEEE Third International Conference on Big Data Computing Service and Applications. [Online]. Available: <https://ieeexplore.ieee.org/document/7944927/>
- [2] A. Alzu'bi, A. Amira, and N. Ramzan, "Compact root bilinear CNN for content-based image retrieval," Aug 2016, pp. 41–45.
- [3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.
- [4] R. Girshick, "Fast R-CNN," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
- [5] Lokanath M, Sai Kumar K, and SanathKeerthi E, " Accurate object classification and detection by faster RCNN," in IOP Conference Series Materials Science and Engineering, November 2017. [Online].
- [6] Available: [https://HYPERLINK](https://www.researchgate.net/publication/321478922_Accurate_object_classification_and_detection_by_faster-RCNN)
"http://www.researchgate.net/publication/321478922_Accurate_object_classification_and_detection_by_faster-RCNN"
- [7] Kaiming He, Georgia Gkioxari, PiotrDollár, Ross Girshick, "Mask RCNN" arXiv:1703.06870,2017.[Online].Available:<https://arxiv.org/abs/1703.06870>"Support Vector Machines". [Online]. Available: <https://scikit-learn.org/stable/modules/svm.html>
- [8] "Raspberry Pi 4 Model B Datasheet, "June 2019. [Online]. Available: <http://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711>"
- [9] <http://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711>"
- [10] /rpi_DATA_2711_1p0_preliminary.pdf
- [11] Jan Hosang, Rodrigo Benenson, BerntSchiele "Learning Non-Maximum Suppression Algorithm," arXiv:1705.02950 May 2017. [Online]. Available: <https://arxiv.org/abs/1705.02950>
- [12] "Selective search for Object Detection," September 18, 2017. [Online]. Available:[HYPERLINK](https://www.learnopencv.com/selective-search-for-object-detection-cpp-python/)
"https://www.learnopencv.com/selective-search-for-object-detection-cpp-python/"
- [13] "Getting Started with VGG Image Annotator for Object Detection Tutorial," September 25, 2020. [Online].
- [14] Available: <https://blog.roboflow.com/vgg-image-annotator/>
- [15] " Recognizing Face using the "face_recognition" library," March 9, 2020. [Online].
- [16] Available:[HYPERLINK](https://medium.com/analytics-vidhya/recognising-face-using-the-face-recognition-library-afdb6d86bcf0) "https://medium.com/analytics-vidhya/recognising-face-using-the-face-recognition-library-afdb6d86bcf0"://medium.com/analytics-vidhya/recognising-face-using-the-face-recognition-library-afdb6d86bcf0
- [17] AbigaSansuri M, Anusuya K.V, Nivethitha M, GuttaAbhinav Krishna Sai, "Framework for Roadside Litter Identification and Face Recognition using Convolutional Neural Networks", in Proceedings of ACM/CSI/IEEE-CS Research & Industry Symposium on IoT Cloud For Societal Applications (IoTCloud'21),2021, pp.103-106.