Object Recognition Through Smartphone Using Deep Learning Techniques

Kiran Kamble(✉), Hrishikesh Kulkarni, Jaydeep Patil, and Saurabh Sukhatankar

Department of Computer Science and Engineering, Walchand College of Engineering, Sangli, Sangli, India
kirankamble5065@gmail.com,
kulkarnihrishi97@gmail.com,
jaydeeppatil3232@gmail.com,
sukhatankarsaurabhl997@gmail.com

Abstract. Object recognition technology has matured to a point at which exciting applications have become possible. Indeed, industry has created a variety of computer vision products and services from the traditional area of machine inspection to more recent applications such as object detection, video surveillance, or face recognition. This paper is about achieving the goal of object recognition through advanced techniques like deep learning on handy devices like smartphones and tablets. Deep learning algorithms (Convolutional Neural Networks (CNN)) are used for the primary aim of object recognition. Images are clicked through the camera of the smartphone during experimentation and are fed to the CNN network. The top four results predicted by the network are depicted on the smartphone screen in the audio and the visual form i.e. predicted object name and the probability of predicted object being the one actually clicked in the decreasing order of probabilities. The accuracy obtained in object recognition is about 93% through the application.

Keywords: Modern object recognition technique · Deep learning application CNN · Mobile application

1 Introduction

Object detection and recognition is the first step in computer vision based research fields. Most of the applications are real time applications, thus object recognition system should be robust, fast and efficient. Object recognition system usually separates out the background and finds the dominating objects present in particular visual media like images or videos. Such systems can be implemented and deployed on wearable devices to facilitate the visually challenged people to ease their life through artificial vision. Obtaining 3-dimensional objects from 2D images and videos is a challenging task. To perform this at the start formulation of the problem is essentially: with input of pre knowledge of how certain objects may seem, plus an image of a scene probably having those objects, to detect which objects are existing in the scene and where. Credit is accomplished by identical features of an image and model of an object. The two most

I. Zelinka et al. (Eds.): ICSCS 2018, CCIS 837, pp. 242–249, 2018.
https://doi.org/10.1007/978-981-13-1936-5_27

important concerns that a method must address are the definition of a feature, and how the identical is found. Obviously these desires are generally difficult to accomplish, for example difficult to recognize objects in images occupied in complete darkness. Even for traditional machine learning models, designing a feature extraction algorithm was essential which generally involved a lot of thick mathematics (complex design), wasn't very efficient, and didn't execute too sound at all (accuracy level just wasn't suitable for real-world applications). This was followed by designing a whole classification model to classify the input given the mined features [1]. With the rise of autonomous vehicles, smart video surveillance, facial detection and various people counting applications, fast and accurate object detection systems started rising in demand. These systems involve not only recognizing and classifying every object in an image, but localizing each one by drawing the appropriate bounding box around it. This makes object detection a significantly harder task than its traditional computer vision predecessor, image classification [2]. Deep learning models have become fairly laid-back to implement, particularly with high-level open source libraries such as Keras, Pytorch, and Tensor Flow [3]. The smartphones provide the hosting platform for the objective of object identification and recognition. Considering the increase in the use of smartphones, they serve as the handy and perfect platform to demonstrate object recognition through deep learning. The paper is ordered as tracks: Sect. 2 discusses previous work with respect to object recognition. In Sect. 3 the proposed method is presented with necessary details. Section 4 gives experimental results. Future scope is provided in Sect. 5.

2 Previous Work

Since the object recognition has gained much influence as one of the significant applications of deep learning, many systems, applications and software have been embedded with this feature. Here are some of the systems exhibiting object recognition [4].

A. YOLO: Provides the users with real time recognition application. The application uses a sole neural network to the whole image. The network splits the image into sub regions and predicts bounding boxes and likelihoods for each section. These bounding boxes are weighted by the predicted likelihoods. Though better than many of the applications, the application hasn't made its way in many of the android smartphones.
B. Glass: Is an android application available on the Google Play Store. It recognizes the dominant object from the image clicked or uploaded. The application suffers drawbacks as far as server speed and real time computation of prediction is concerned.
C. Object Recognition-Free Computer Vision: This is yet another android application on Google Play Store which serves the users with the feature of object recognition from the clicked image. The application though helps to recognize the objects along with the probabilities denoting the accuracy, there are always limited number of categories in which the object could be classified in and all the categories (about 10) and the probability of object being belonging to that is displayed to the user.

D. Machine Learning detection: The android application on Google Play Store serves users with real time object recognition feature. It provides the results with the recognized object being bounded along with the predicted object name and accuracy. The application size is about 90 mb and lists single probability when the object is focused upon by the camera.
E. Click2Know (C2K): is a very handy, an optimal size Android application which facilitates the users with multiple object recognition from the clicked image. It helps users to get the results computed real timely (about 2 s from clicking and strong internet connection) and lists out four results based on predicted objects and corresponding accuracies in decreasing order of significance.

3 Proposed System

The main aim of proposed system is to recognize objects through smart phone for visually challenged people using the deep learning on Android platform. A flowchart in the Fig. 1 shows the actual methodology. Initially user (visually challenged) has to create an account simply by speaking create an account. Using account user can capture the photo of object to be recognized. Within fraction of seconds produced result will be spoken as well as displayed on the screen.

The first one is the firebase which is used as real time database so that real time result can be computed. The image which is captured by camera of user is uploaded to real time database in the format of byte image. Database schema is represented in the Fig. 2.

The schema of users is comprises of unique UID (user id) which is generated according to firebase account of the individual user along with the image which is uploaded at that instant of time multiple requests from different user at same instance of time.

Image which is in the database (firebase) is taken and the results are pushed back into the database. The image which is in the byte format needs to be converted into regular .PNG format so that further proceeding can be done on the image.

Conversion of the image to 299 $\times$ 299 pixel is performed. Before feeding image to Convocational Neural Network Inception V3 model it is necessary to convert linear structure like Array which is vectored form of corresponding image in .rbg format.

Feeding these values directly into a network may lead to numerical overflows. It also turns out that some choices for activation and objective functions are not compatible with all kinds of input. The wrong combination results in a network doing a poor job at learning which is done by pre-processing technique Dimensionality Reduction [4]. Dimensionality Reduction helps in transforming vectored image data into a compressed space with less dimensions, that can be useful to control the amount of loss and it uses as input to CNN. After performing dimensional reduction it is necessity to adequate image to the format the model requires. The image is fed to CNN model for prediction (Fig. 3).

Convolutional neural networks (CNNs) is present high-tech model architecture for image classification tasks. CNN process a sequence of filters to the raw pixel records of

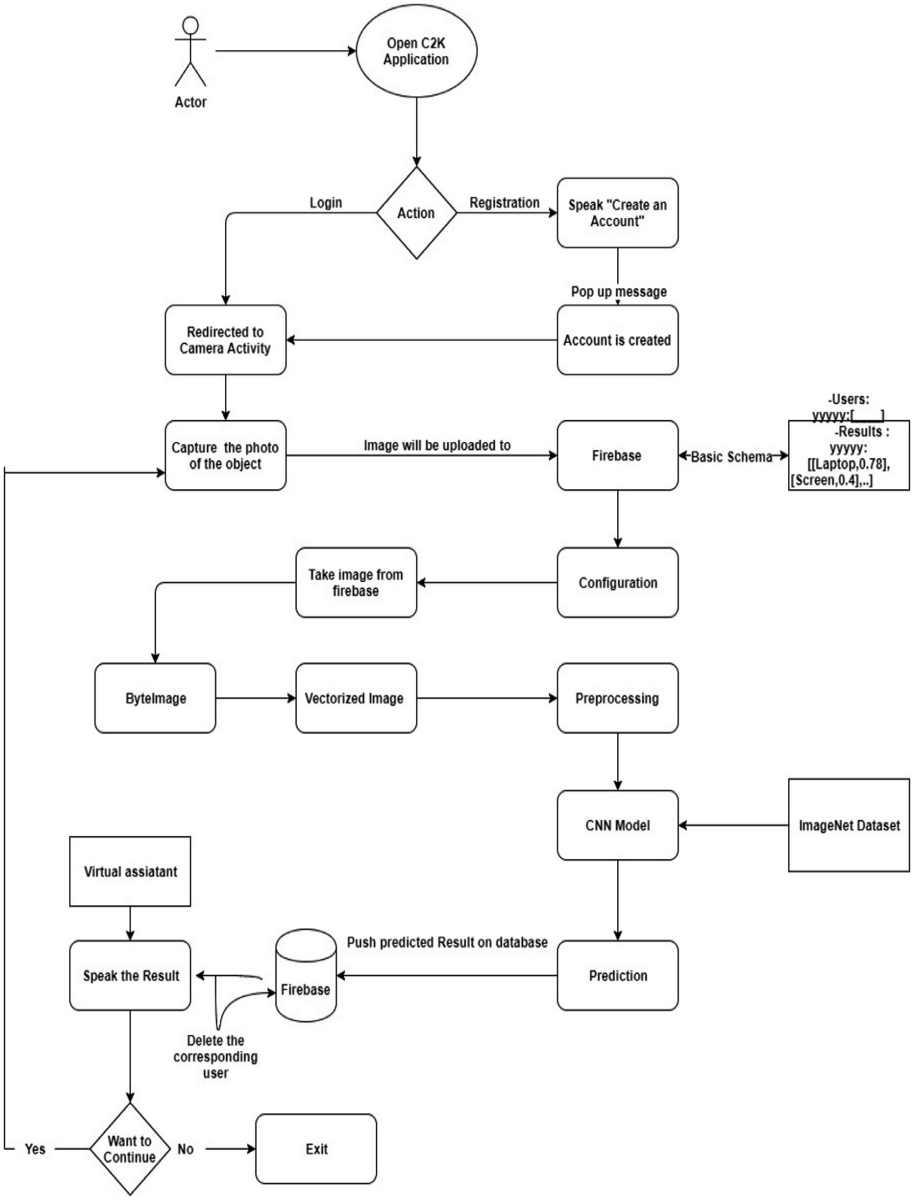

Fig. 1. Flowchart of proposed system

Fig. 2. Real time database schema for users

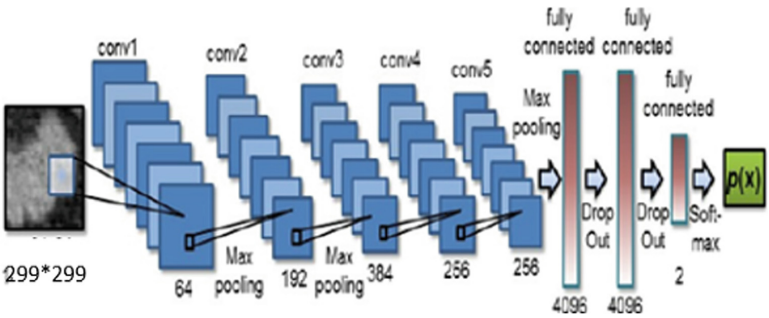

Fig. 3. CNN architecture

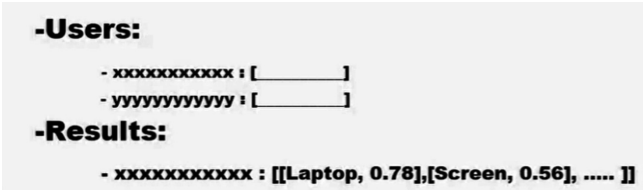

Fig. 4. Real time database schema after result computation

an image to mine and lean high-level features, which is use to classify. CNNs contains three components [5]:

A. Convolutional layers: This layer applies a definite number of convolution filters to the image. For each substitute region, the layer executes a set of mathematical operations to produce a single value in the output feature map. Convolutional layers then typically apply a ReLU initiation function to the output to present nonlinearities into the model.
B. Pooling layers: Pooling layers depressed sample the image data mined by the convolutional layers to shrink the dimensionality of the feature map in order to decline processing time. A frequently used pooling algorithm is max pooling, which mines sub regions of the feature map (e.g., 2 × 2-pixel tiles), keeps their maximum value, and discards all remaining.
C. Dense (fully connected) layers: Dense layers accomplish classification on the features mined by the convolutional layers and down experimented by the pooling

layers. In a dense layer, every node in the layer is linked to every node in the prior layer. After classification results will be in the form of like Fig. 4.

Here 0.78 means the probability of laptop being in the image is 0.78 in the similar fashion 0.56 conveys the probability of screen being in the image is 0.56 so on. The among the computed results the images having top 4 probabilities will be pushed onto the real time database within fraction of seconds for that specified user. Now the modified schema will look like in the Fig. 2, where xxx… is unique UID.

After computed results were pushed onto the real time database the android application will fetch it instantly. As result were fetched from real time data base there is no need to store them for future use. In order to facilitate fast and efficient working of real time database the user along with result are deleted. The android application will show the result of object recognition in audio as well as text format in real time.

4 Experimental Results

For implementing the system, ImageNet dataset is used. The dataset includes 14,197,122 total images in .jpeg form. These images consist of total 21841 non-empty synsets. Other details about the dataset are as [6]:

Total number of non-empty synsets: 21841. Total number of images: 14,197,122. Number of images with leaping box annotations: 1,034,908. Number of synsets with SIFT features: 1000. Number of images with SIFT features: 1.2 million.

The proposed approach for object recognition is based on Convolutional Neural Network which is able to recognize about 21841 objects (classes) in ideal case. The performance of the system is depicted in Table 1. For calculating confusion matrix, the application is tested 60 times. 26 times out of 30, the system recognizes object correctly (Here, correct recognition is considered as the predicted object is equivalent to the class of object in which it falls). It is important to note that the system is able to compute results correctly even when image consists of multiple objects.

Table 1. Confusion matrix for performance of system

		Actual	
		True	False
Predicted	True	26	7
	False	4	23

But when the accuracy of predictions of entire system is concerned, only 60 tests are not enough. According to the online documentation, the model used in this application is able to give accuracy as high as 94.4% in top five predictions. However top one accuracy is bit less i.e., 78.8% [7]. Thus the application flow goes like this. One has to create an account (By speaking “Create an Account”) once when he/she installs the application for the first time. After successful account creation, the application intent redirected to the camera activity. Here user can capture images of surrounding.

The image captured is then converted into Bytes of integer and sent to the firebase database. Because of his/her account on the firebase (which is created earlier), user will get unique UserID/QueryID which is useful for differentiating user requests from each other. Now the continuously running python script fetch the Byte-converted image and convert it into .PNG format image. The converted image is then fed to the CNN model after doing some pre-processing as mentioned in the proposed system. The top-four predictions are computed and results are again pushed back to Android application via Firebase. Such fetched results are been shown on the screen along with voice guidance/pronunciation. Some sample results are shown in Fig. 5.

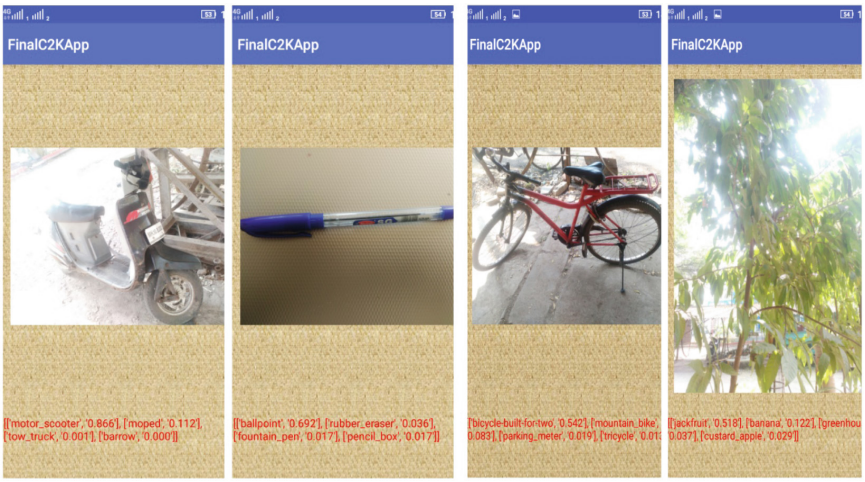

Fig. 5. Sample result

5 Future Scope

The work can be extended further with the various ideas like: Switching from CNN to Recurrent Neural Network (RNN) in order to reduce the computation time and accuracy. Deploying the Object Recognition (server-side scripting) over the wearable smart devices [8] like smart Google Glass, Smart Shoes or Smart Sticks which can result in more usefulness of the application for physically challenged people

Acknowledgements. We sincerely thank to all panel members for their guidance and encouragement in carrying out this work. We also highly indebted to Walchand college of Engineering Sangli for providing necessary information regarding this research and financial support to carry out this work.

References

1. Elisa, M., Giulia, P., Lorenzo, R., Lorenzo, N.: Interactive data collection for deep learning object detectors on humanoid robots. In: 2017 IEEE-RAS 17th International Conference on Humanoid Robotics, pp. 862–868 (2017)
2. Qi, M., Zong, Z.: A new method of moving targets detection and imaging for bistatic SAR. In: 2014 Seventh International Symposium on Computational Intelligence and Design, vol. 2, pp. 224–227 (2014)
3. Zainab, A.: Research blog about various libraries that can be used in deep learning. https://medium.com/mindorks/detection-on-android-using-tensorflow-a3f6fe423349. Accessed 27 July 2017
4. Raj, D., Gupta, A., Tanna, K., Garg, B., Rhee, F.C.H.: Principal component analysis approach in selecting type-1 and type-2 fuzzy membership functions for high-dimensional data
5. Fakhrulddin, A.H., Fei, X., Li, H.: Convolutional neural networks (CNN) based human fall detection on body sensor networks (BSN) sensor data, pp. 1461–1465 (2017)
6. Article showing detailed explanation of ImageNet dataset including various classes and their total counts, etc. http://image-net.org/about-stats
7. Document of keras library giving accuracies of various models. https://keras.io/applications/
8. Delrobaei, M.: Errata to "using wearable technology to generate objective Parkinson's disease dyskinesia severity score: possibilities for home monitoring". IEEE Trans. Neural Syst. Rehabil. Eng. **25**(11), 2214 (2017)