Cloud-Based ImageNet Object Recognition For Mobile Devices

Akram T. Saeed[1], Dan Schonfeld [2]
akram.saeed@uoninevah.edu.iq[1], Dans@uic.edu[2]
Ninevah University[1], University of Illinois at Chicago[2]

Abstract. User reliance on real-time applications is continuously increasing as the use of smartphone devices has tremendously increased in day to day life within the past few years. Smartphone devices computation power has significantly increased as well, however, there are still some scalability, performance challenges and some complications with real-time application such as limited computation capabilities and battery life of mobile devices. In this paper, we propose a cloud-based object recognition through task offloading to a high-speed server. We explore this design extensively and demonstrate a real-time solution for object recognition framework on mobile devices using a Convolution neural network (CNN) leveraging the ImageNet dataset and by optimizing the offloading process to minimize the time and energy needed. This framework use the emerging the Android operating system as a platform to connect with an object recognition server where the CNN deep learning resides and process received images. Using this method will overcome the design limited capacity of mobile devices since object recognition algorithms require high-speed calculations.

Keywords: Cloud computing, Convolutional neural network, Object Recognition.

1 Introduction

As internet access becomes easier and available on almost any location using the cell phone data plans with affordable cost on mobile devices, free Wi-Fi almost around every corner has given the user the ability to reach any required information from any location. This paper is mainly discussing an approach to building a cloud-based communication between a mobile phone and an object recognition server. Due to the rapid improvement of image acquisition devices, high-resolution cameras, Image and object recognition moved out of limited industrial and medical use and entered the mainstream and became widely popular another aspect of the new development is the smartphones are no longer isolated resources devices. They now are built with highly sophisticated technologies with great data connectivity, however, the amount of needed processing to get instant real-time results is often infeasible while running object recognition algorithms on the mobile, and therefore an offloading process is being used to achieve the goal. Another main goal of the project is to demonstrate the power of combining different cloud network APIs with a deep learning module server to assist the user with these real-time applications and build a sophisticated mobile application for android devices which will as the framework that will execute the offloading and retrieval tasks and processed data instantaneously and accurately. Our results show that cloud-based object recognition outperforms local-based frameworks in many performance metrics. The user does not need to understand the internal infrastructure of

IMDC-SDSP 2020, June 28-30, Cyberspace

DOI 10.4108/eai.28-6-2020.2297916

the system to get the needed data. It is as simple as starting the application. some other work [1] showed that local recognition and rendering performance for the object to be recognized is working but only for some scenarios with limited detection algorithms and slightly lower accuracy due to large image dataset required to be stored and a huge amount of labelled contents and the intensive computation power that requires high energy and a lot of time which is about 3 seconds to load and analyze the received image on a Snapdragon 800 mobile processer. Using cloud-based object recognition will compensate for limited mobile capabilities, with the assist of web APIs that provide the mechanism to link the mobile devices with cloud-based server to deliver the object recognition tasks and deliver back useful information. Some challenges will be faced during the offloading process such as network connection delay and object-recognition server processing time. The results of the recognition should be sent back to the mobile device after a moment [2].

2 Machine learning and Object recognition

Machine learning techniques have successfully revolutionized many aspects of the field of computer vision. With the discovery of feature extractions popular methods such as “scale-invariant feature transform (SIFT)” and “Speeded Up Robust Feature (SURF)” complicated tasks such as object recognition and augmented reality have tackled and properly handled, however, more sophisticated techniques are needed to achieve higher accuracy and less latency. Deep learning models and “convolutional neural networks (CNN)” have become a superior method to use for large-scale pattern recognition with a model such as ImageNet where it can outperform humans in terms of recognition capabilities. ImageNet was mined from the internet and was initially built by Google and put together in a large visual dataset. Different methods and techniques are being used in DIP to transform input images. Various criteria must be taken into consideration while selecting the appropriate technique and it significantly depends on what result we want to achieve.

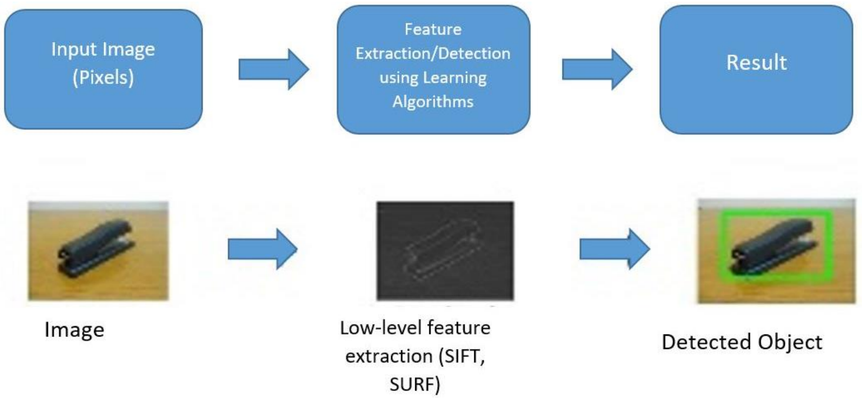

Input Image (Pixels)
Feature Extraction/Detection using Learning Algorithms
Result
Image
Low-level feature extraction (SIFT, SURF)
Detected Object

Fig. 1. The traditional object recognition algorithm

2.1 “Scale Invariant Feature Transform (SIFT)”

Scale-invariant feature transform (SIFT) become [3] a very well-known feature-based algorithm in 2004 after it was introduced by Lowe. A lot of different versions of this algorithm exists in combination with other approaches in order to get rid of its disadvantages. All the algorithms are consist of four major stages [1] but here we are going to discuss the general SIFT. Firstly, all images are searched over all other image locations and all scales. With the difference of Gaussian (DOG) such a process can be achieved easily as shown in figure 2. The left side contains results of the convolution of an input image I with a Gaussian filter GA, which is calculated for all different scales. The difference between the two adjacent scales is DOG. In the following two steps, extrema are localized and the orientation of each maximum is calculated. The Maximum and the minima of the DOG images are computed by comparing the pixel with its 26 neighbors. 8 neighbors in the current scale and 2*9 neighbors in the adjacent scales. [2]. lastly, key points or feature descriptors are presented in the last stage. Each different local feature produces a descriptor that has different information about the gradient magnitude and orientation around a sample point. That is weighted with a Gaussian window and 4x4 sub smaller regions are combined into an orientation histogram. [3]

2.2 “SURF - Speeded-Up Robust Features”

H.Bay, and et al. adopted SIFT as a model for their more advanced speeded up robust feature (SURF) algorithm. As a feature-based algorithm consists of the exact processes as SIFT, but provide different innovations. In the beginning, features have to be specified. SURF uses integral images and the Hessian matrix to detect maxima. The Hessian matrix has an approximation of the convolution of Gaussian second-order derivatives and the image. [4]

2.3 “CNN – Convolutional Neural Network“

It was important to be aware of the various existing technologies in the tech and research field, studying the different technologies to present a different approach. Studying the existing method for object recognition to achieve high accuracy object recognition, CNN (Convolutional Neural Networks) was the selected method after comparing it with traditional CV methods as we will discuss further. Neural networks are known for one special type of recognition, which it can be superior over other types which particularly suitable for object recognition (deep) convolutional neural networks (CNN) concept was partially inspired by the biological human cortex in which one or more hidden layer and convolutional implies the use of convolution layers

Figure 2 below shows an example of the architecture where the Input images are convoluted with a filter to acquire 3D featured maps. A sub-sampling, or sometimes called pooling, degrades the amount of data. This procedure keeps going until a one-dimensional vector, which represents the different classes is acquired.

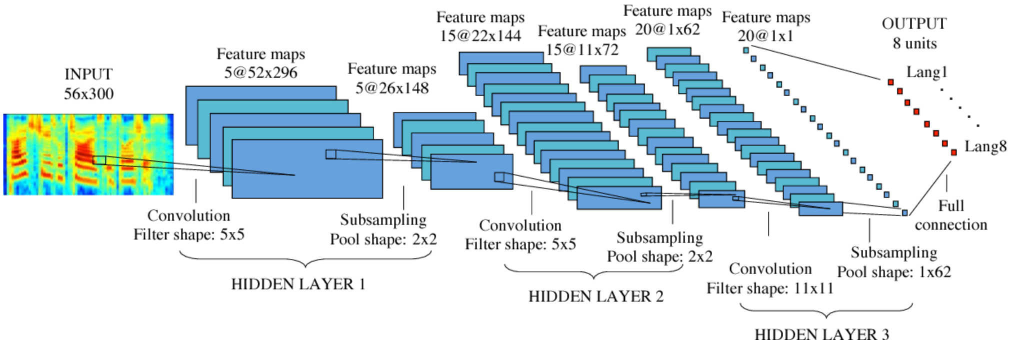


Fig. 2. Convolutional neural network using the subsampling and hidden layers.

Like many existing object recognition algorithms, CNNs require training to balance all weights of neurons. In that phase, various features are extracted. Low-level features contain color, lines, whereas edges and corners belong to mid-level features. High-level features already include class-specific forms or sections. [5]

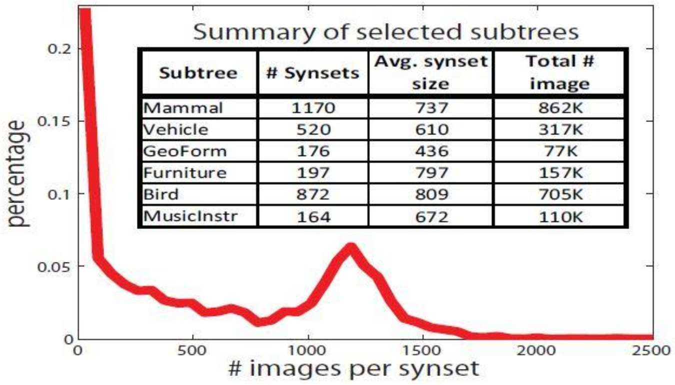
In recent past years, the optimum object recognition algorithms have achieved accuracy rates of about 70 %. During the last few years and the introduction of deep learning methods, new improvements were achieved. Very recent and complex algorithms achieve results far beyond 90 %. Although the principles of NN is way older than the other methods, it still can compete with them in different ways. Especially the progress in the betterment of faster hardware, it is no problem anymore to operate with a huge amount of parameters and equations. Especially taking into consideration the continuous enhancement of CPUs and GPUs, scientists expecting deep neural networks and in particular, CNNs to be the most common way for object recognition in the near future. [6]

3. ImageNet Dataset

“The ImageNet” Dataset is a visual library was built for use in object recognition software research. There are over 14 million [7], images that are manually labeled by the project to indicate what object is captured. ImageNet saves more than 20000 categories with various categories, such as "birds" or "cats", each category is consisting of several hundred images. See below statistics and histogram Fig.3 The database of labeling of third-party image URLs is freemium and now available on the ImageNet website. The saved images are not owned by the ImageNet dataset.

ImageNet is hierarchical structure that is offered by WordNet. In its completion, ImageNet plans is to hold more than 50 million cleanly annotated full-resolution clear images.

.



Subtree	# Synsets	Avg. synset size	Total # image
Mammal	1170	737	862K
Vehicle	520	610	317K
GeoForm	176	436	77K
Furniture	197	797	157K
Bird	872	809	705K
MusicInstr	164	672	110K

Fig.3. The scale of ImageNet. Red curve: Histogram of the number of images per synset [7].

3.1 Training Image Dataset

After downloading the ImageNet training dataset and validation data, they will be stored on the server disk like Training and validation inputs are saved as train.txt and validate.txt as txt listing all files and their labels. Adding more datasets to the current ImageNet images database will increase recognition accuracy significantly. The following examples (figure 4) are obtained results before and after training.

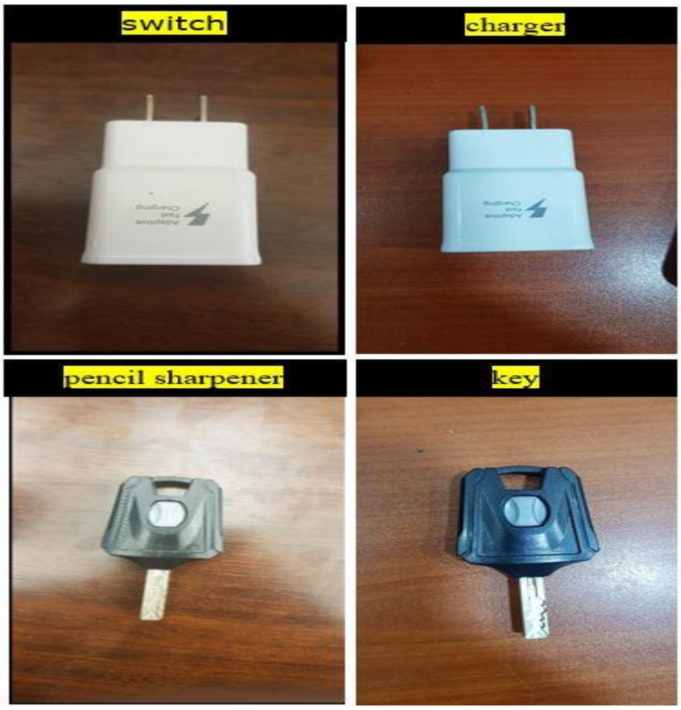


Fig.4. Untrained versus trained ImageNet results.

4 Cloud-Based Object Recognition Server

An HTTP cloud web server and PHP installed to receive the posted image from the mobile app, on the client-side, an image will be uploaded to a temporary storage folder on the server, and the deep learning object recognition module (ImageNet) which currently used will start running once the image is received. Once the image is processed as we discussed earlier, a label of the recognized object will be pushed back and saved on the server automatically as illustrated in figure 5. The system architecture that consists of a mobile device client and the cloud server as following is a short description of the overall procedure:

On the client-side, the server takes the image, compresses it and saves its size and pixels distribution the object recognition CNN in the server extracts feature points within the view once the image is received. Unlike [8], our software does not require the client to enter the area of interest for the recognition to be achieved. Then the result is direct gets sent back to the mobile application, including the label information of the target object.

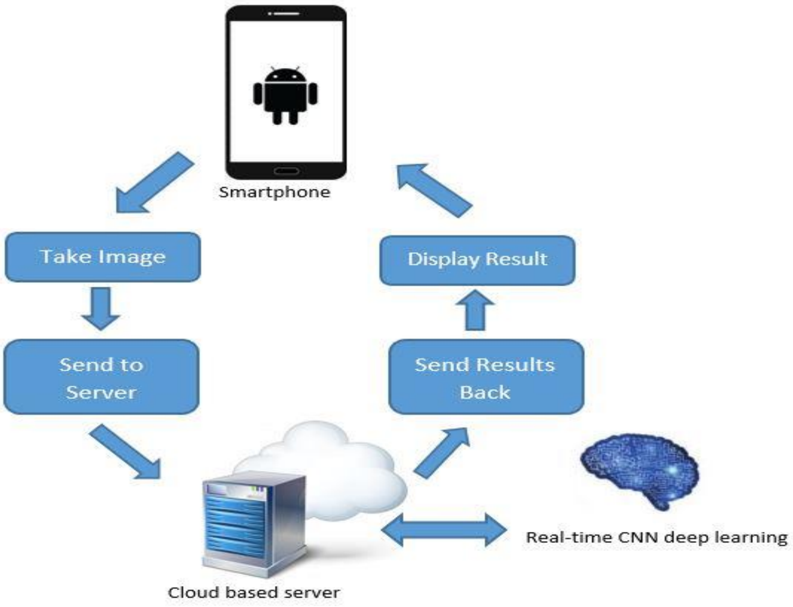


Fig.5. Client-Serverer system overview.

4.1 Network Module (TCP versus UDP Protocol)

The network module is the channel that carries the information between the client device and the server. To minimize the size of transferred data and the overhead of the handshaking process we propose the use of the UDP connection less-protocol as reducing latency through using a faster network protocol is the crucial part, a drawback to the use of UDP could be packet loss. UDP is known to be an unreliable transfer method for important pieces of data, but our system handle packet loss as they happen by ensuring that image sent size and pixels distribution is the same as the image received. Object recognition on the cloud happens each time a new image is sent, and a new recognition result can be received. If a reliable network protocol such as TCP is utilized in this case to transmit the information, the results could slightly take longer as the re-transmit is required for the lost data packets, below tables shows the difference in the time we found when we used two different protocols to send the data to the server for processing.

Table 1. TCP and UDP network protocol comparison.

Image size after compression in (KB)	Upload time through TCP	Upload time through UDP	Result Response time Through TCP	Result Response time Through UDP
2,650	687ms	359ms	10ms	5ms
2,200	491ms	196ms	10ms	6ms
2,233	511ms	206ms	9ms	5ms

4.2 Server Design

In smartphones object recognition frameworks, complicated real-time image processing tasks are almost impossible for the current computational capabilities in real-time on current mobile devices. In the framework we proposed, we use cloud object recognition servers as a processing platform for those large-scale tasks. We used an Intel Xeon Core i7-8950 CPU 3.2 GHz processor and 16 GB of RAM, the OS is Windows 10 and graphic card NVIDA GTX 1080 8GB which allows the mobile to offer a smooth and enjoyable experience. Cloud object recognition latency is a major issue that needs to be handled by the system properly so that the labeling contents can be aligned with the physical objects accurately.

One possible solution to minimize the server latency time is to send the image compressed to the server to calculate the current pose of the image. However, object matching is not a trivial task on mobile, which consumes much time and energy when compared with cloud high-speed processing. Executing object recognition on the server reduces the overall time required significantly see figure 6.

Image with recognized label	**Upload time (after Compression)**	**Analysis Time**	**Total time**
Pin 	359ms	1.878 sec	2.273 sec
Cup 	196ms	1.814 sec	2.01 sec
Mouse 	204ms	1.953 sec	2.157 sec

Fig.6. Upload and analysis time.

5 Conclusion

In this paper, there were some challenges during implementing a CNN and the optimization of the network module that is efficient and reliable to allow to connect it to the ImageNet Dataset. We presented a "cloud-based object recognition framework" which aims at resolving the large-scale patterns of object recognition and real-time processing problems on mobile devices. We proposed an optimized network module and android application running on mobile to assist in the offloading task and mitigate the offloading latency issue so the user will not be able to realize high latency during the object recognition process. Our results showed that the cloud-based app performs sufficiently in terms of smooth object recognition experience and robustness which offer real-time object recognition with less than 2.2 seconds of time needed, low offloading delay, and high performance capable of running the large ImageNet dataset.

References

[1] Zhang, Wenxiao & Lin, Sikun & Hassani Bijarbooneh, Farshid & Cheng, Hao & Hui, Pan. CloudAR: A Cloud-based Framework for Mobile Augmented Reality. (2018)

[2] X. Yang and K. T. Cheng, "Mobile image search: Challenges and methods," in Mobile Cloud Visual Media computing. Springer, 2015, pp. 243–262.

[3] Saeed, Akram T. Location Based Assistant on Android Platform. Diss. 2017.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). In Computer Vision and Image Understanding 110, 2008.
[5] Josip Josifovski. Object recognition: Sift vs convolutional neural networks. University Hamburg - Lecture Intelligent Robotics WS 2015/16, 2015.
[6] NVIDIA. Deep learning on GPS http://ondemand.gputechconf.com/gtc/2015/webinar/deep-learningcourse/intro-to-deep-learning.pdf, mar 2016.
[7] http://www.image-net.org/about-stats.
[8] J. Jung, J. Ha, S.-W. Lee, F. A. Rojas, and H. S. Yang, "Efficient mobile AR technology using scalable recognition and tracking based on server client model," Computers & Graphics, vol. 36, no. 3, pp. 131–139, 2012.