Object Recognition Using a Smartphone

Jeremiah Cheuk-Nam Mai, Ngai Hon Lung and Sham Kwan Nok

Advised by
Professor Yeung Dit-Yan

Introduction

Whenever we look at an image, we may want to know what the image is capturing. However, this is not an easy job for us since there are so many different kind of things in the world with labels we may be unaware of .This is where a computer system that can recognize different objects come in handy. In this project we do a feasibility study on building an object recognition system for Android. The object recognition had to recognize 40 different models of toy cars under various background conditions. To build this system, we implemented and tested various machine learning algorithms on a cropped dataset consisting of 40 different models of toy cars. Due to the complexity of the recognition algorithms and the limited computing power of the Android platform, we had to come up with ways to improve not only the accuracy of the system but also the speed performance of the system.

Objectives

The objective of this project is to develop a smartphone application which uses a built-in camera to capture an image of a toy car. The application will then recognize the toy cars within a reasonable time frame and with high accuracy.

Design

There are three main parts which constitute of our project. The first part is building a Convolution neural network (CNN) classifier for recognizing the class of the car within a cropped image. The second part is applying a detection algorithm for locating the approximate location of the car in an image. The final part is the integration of the first two parts on a smartphone targeting the Android Platform. Here are more details of the three parts:

1. Convolution neural network (CNN)

Image preprocessing:

Before the CNN classifier could be trained, we did certain preprocessing techniques to our dataset in hopes of improving our classifier's result. Methods included rotation, add noise, blurring and zooming as shown below:

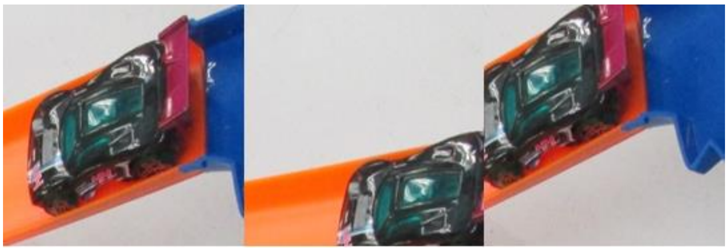

Fig.1. Rotation

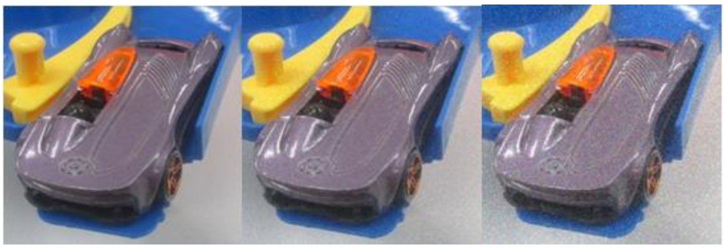

Fig.2. Adding Noise

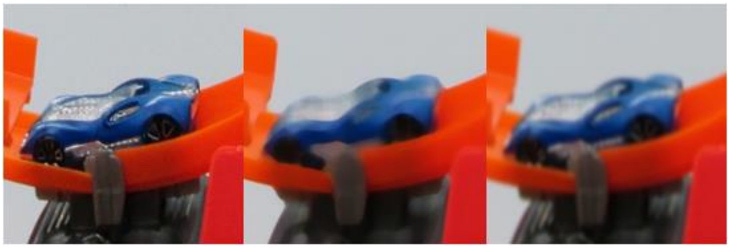

Fig.3. Blurring

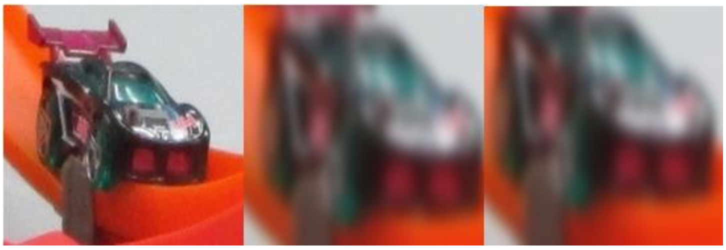

Fig.4. Zoom

CNN classifier:

We used the Caffe framework to fine-tune a pre-trained ILVRC12 network used in "imageNet". Our resulting network architecture is similar to the ILVRC12 CNN architecture except for the last layer, which we modifier the output to accommodate our 40 models of toy cars.

2. Object Detection:

Before doing the car recognition, the image will be processed by applying a feature detection algorithm. We are looking for the feature points we are most interested in, mainly is the toy car, but feature points from background are also included. By adjusting the threshold of the feature detector and number of the best feature which should be retained, most of the background feature can be eliminated. Finally the toy car can be located by finding the region with most feature points. Here are some result images:

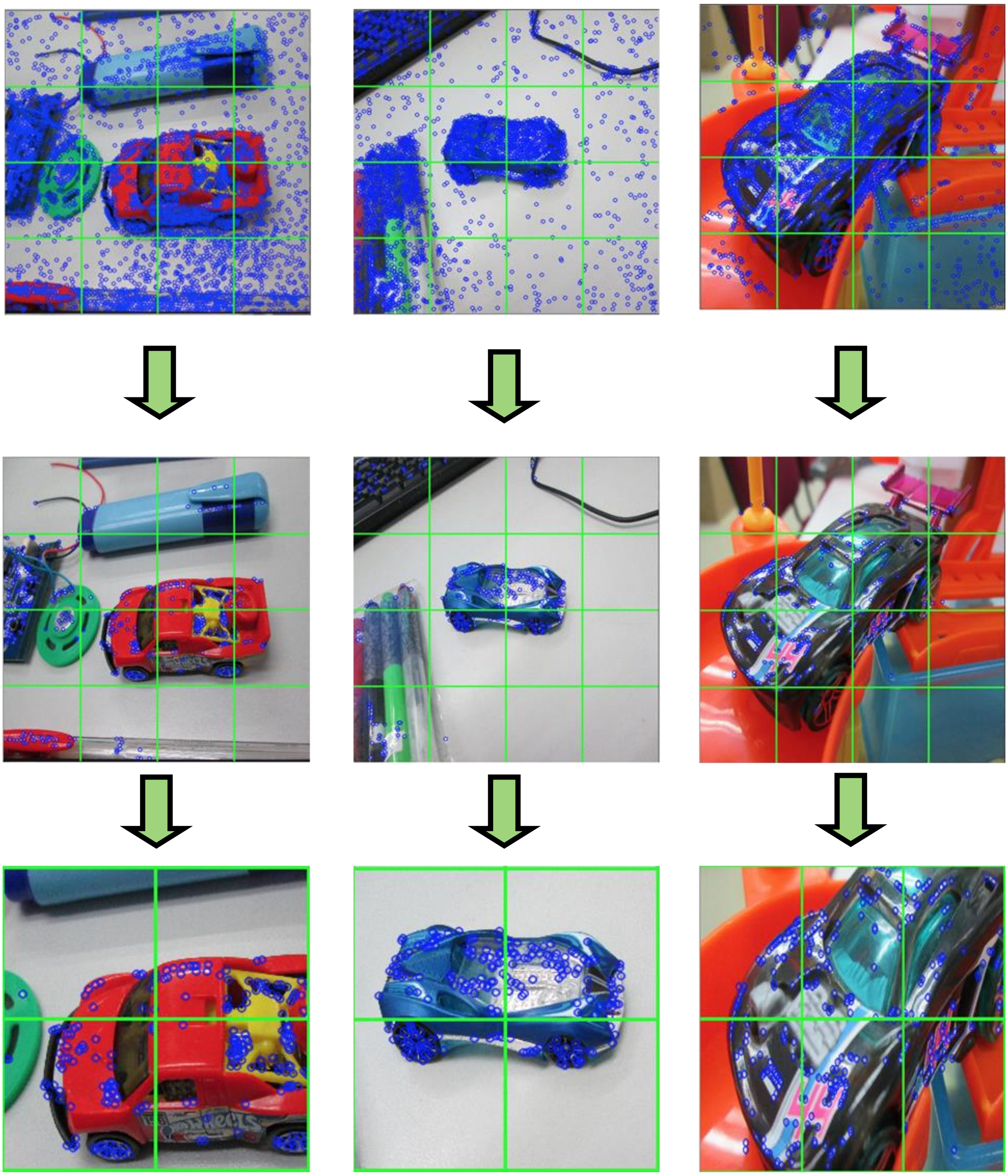

Fig.5. Images with feature points after applying detection algorithm

3. Android Application:

The Android application implements the detection algorithm to find approximate location of the car and then make use of the trained CNN model to perform classification. It aims to provide an easy to use interface, achieve good accuracy and recognition speed. Results are listed in a rank of 5. This allows higher chance of giving users a correct model. The preview screen and result screen are shown on the right:

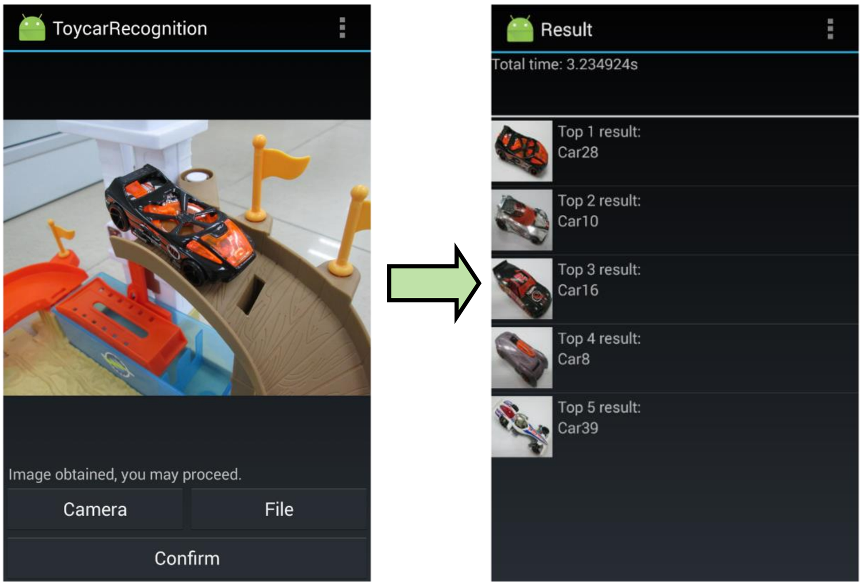

Image obtained, you may proceed.

Total time: 3.234924s

Top 1 result: Car28

Top 2 result: Car10

Top 3 result: Car16

Top 4 result: Car8

Top 5 result: Car39

Fig.6. Preview screen & result screen of the application

Test Results

Raw images of 40 different car models are tested. Top 1 and top 5 results are shown in Table.1. We can see that top 1 result achieved an average of 61% and top 5 is 83%.

Testing on speed shows that even for high resolution images, recognition can finish within 5 seconds.

	Top1	Top5		Top1	Top5		Top1	Top5		Top1	Top5		Top1	Top5
Car1	66%	93%	Car9	60%	75%	Car17	50%	81%	Car25	45%	90%	Car33	83%	94%
Car2	31%	56%	Car10	75%	95%	Car18	63%	73%	Car26	95%	95%	Car34	42%	68%
Car3	31%	63%	Car11	65%	85%	Car19	94%	100%	Car27	80%	95%	Car35	60%	80%
Car4	62%	75%	Car12	66%	88%	Car20	65%	95%	Car28	88%	100%	Car36	50%	70%
Car5	15%	47%	Car13	68%	94%	Car21	45%	70%	Car29	65%	80%	Car37	47%	89%
Car6	36%	57%	Car14	80%	90%	Car22	52%	68%	Car30	44%	100%	Car38	45%	95%
Car7	60%	65%	Car15	90%	95%	Car23	70%	90%	Car31	57%	84%	Car39	90%	100%
Car8	80%	90%	Car16	70%	100%	Car24	55%	80%	Car32	50%	80%	Car40	42%	73%

Table.1. Result for 40 car models

Conclusion

At the end of our project we managed to design and implement a smartphone application which can recognized 40 different models of toy cars. Overall, the project was successful as we managed to implement the recognition system on Android and we were pleased with the speed of recognition on the device. However, we believe our classification accuracy could be improved even further once more powerful Android devices are released.