On Safari with TensorFlow: Assisting Tourism in Rural Southern Africa using Machine Learning

Laurie Butgereit
School of ICT
Nelson Mandela University
Laurie.butgereit@mandela.ac.za

Laura Martinus
Embedded Intelligence Systems
Meraka Institute, CSIR
LMartinus@csir.co.za

***Abstract*—Tourism is a major contributor to employment in southern Africa and a major contributor to gross domestic products of many southern African countries. One of the major tourist attractions in many southern African countries is the wild animals. Major national parks such as Etosha in Namibia and Central Kalahari in Botswana often have rangers available to assist tourists on their game safaris by recognising animals and describing their habitats. Many of the smaller reserves, however, do not have the luxury of rangers available to tourists. At such smaller reserves, tourists are left on their own to recognise the various animals. This paper describes the use of Google's TensorFlow to create an image recogniser trained for southern African mammals. The recogniser was embedded in an Android mobile app and could then assist tourists at smaller reserves.**

Keywords— machine learning, image recognition, tourism, tensorflow, android

I. Introduction

In its 2017 Economic Impact report on South Africa, the World Travel and Tourism Council (WTTC) reported that the direct contribution of travel and tourism to the South African gross domestic product (GDP) was USD 8.7 billion (ZAR 127.9 billion) or 3% of South Africa's GDP for 2016 and that value was expected to rise to USD 13.9 billion (ZAR204.4 billion) in by 2027. In addition, travel and tourism directly supported 716,500 jobs in 2016 (4.6% of total employment) and that value was expected to rise to over one million jobs by 2027 [1].

These values are echoed in the entire SADC (Southern African Development Community) region with the direct contribution of travel and tourism to the GDP of the SADC region being USD 17.4 billion or 3% of the GDP of the region with this value expecting to rise to USD 28.9 billion by 2027. Travel and tourism directly supported 2.5 million jobs in the SADC region in 2016 and that value was expected to rise to just under 4 million jobs by 2027 [2].

One of the many tourist draw cards in South Africa and the greater SADC region is the abundance of wild animals including the Big Five - lion, leopard, rhinoceros, elephant and buffalo. In addition to the Big Five, the region has an abundance of zebra, giraffe, cheetah, hippopotamus, and a wide variety of antelope.

Many of the larger national parks such as Kruger and Pilanesburg in South Africa and Chobe and Central Kalahari in Botswana offer safaris or game drives led by professional game rangers. These professional game rangers point out and describe the animals that the group encounters.

Many of the smaller national parks, however, do not have the luxury of professional game rangers in attendance and visitors are left on their own to identify the animals they encounter. Areas around these smaller national parks are often neglected by tourists. In addition, many of these national parks do not have cell phone connectivity or they have poor connectivity. In such cases, a self-contained mobile app which does not require online facilities could assist visitors during their self-driven game drives. In addition, if the mobile app were designed properly, different recognition models could be installed specifically for mammals, or specifically for birds, or insects, etc.

It is important to note that the goal of this research is not to replace professional game rangers. The goal of this research is to see if an app could assist tourists at the less known game reserves (where there are no professional rangers available because of economic reasons).

This research described in this paper investigated the following:

1. How many images are required to adequately train a model for identifying African mammals?
2. Would the size of the model and the size of the mobile app be suitable for mid-range mobile devices?
3. Would the mobile app work adequately with stand-alone images?
4. Would the mobile app work adequately in a game drive or safari environment?

This paper has the following structure. Section II provides background on machine learning. Design Science Research (DSR) was used for this project and DSR is described in Section III. This project was effectively divided into two sub-projects: one to create a model and one to do the recognition. This is explained in Section IV. DSR is an iterative research methodology and a number of iterations are described in

Sections V through VII. The results and conclusion can be found in Sections VIII and IX respectively.

II. Literature Review

Various image recognition techniques could be used for the creation of a classification system for African animals.

A. Template Matching

A simple technique for image recognition is template matching in which a template which represents each of the classifications is stored. Each new input image is matched against the all the templates and the best matching template then results in the classification of the image.

B. HMMs and ANNs

Hidden Markov Models (HMM) can be used for image classification. For each of the image classifications a statistical model is created into which the features of the input image are entered [3]. Each model outputs the probability that the input image matches the classification and the highest probability is chosen as the classification of the image.

Artificial neural networks (ANN) are often used in image classification. ANNs are composed of neurons. The connections between the neurons have an associated weight which can be fine-tuned through the training of the network. An adequately trained ANN can then accurately classify images according to their features [4].

A common downside to most of these image recognition systems is that the pixels of the image are not input. These techniques rather take the extracted features of the image as input. This requires a feature extraction method to be developed in addition to the classification system. Convolutional neural networks (CNN) are considered to be one of the most accurate methods in image classification as instead of taking a vector of image's features as input it takes the images as input [5]. A disadvantage of CNNs, however, is that a large amount of data is necessary in order to adequately train the network [6].

C. CNNs

A CNN is well adapted to be used in image classification and was designed in a way that simulates human vision [7].

CNNs convert the input image's pixel values to the final classification through convolutional, pooling, and fully connected layers, in various arrangements [6]. The convolutional and fully connected layers have associated parameters (weights and biases) while pooling layers do not.

The input neurons to the CNN is the two dimensional array of pixel intensities, usually the three colour channels red, green, and blue (RGB) [8]. These neurons are connected to a hidden layer, but unlike in a standard ANN, not every input neuron is connected to every hidden neuron. Small regions of the input neurons are connected to a hidden neuron as depicted in Illustration 1. This small region of input neurons is referred to as the local receptive field of each hidden neuron [6]. Each of the connections between the local receptive field neurons and the hidden layer neuron has a weight which it needs to learn and each of the hidden neurons has an overall bias which it needs to learn in order to be able to analyse the local receptive field it connects to [9]. The output of the neuron is computed by calculating the dot product of the weights and the local receptive field followed by a non-linearity [9]

Each local receptive field is shifted by one pixel to form the next field as can be seen in Figure 1.

This first hidden layer detects one feature, meaning that all of the neurons in one hidden layer detect the same feature. By having one layer detecting the same feature in all the local receptive fields is one of the reasons why CNNs are well adapted to image recognition [6]. It would still be possible for the CNN to recognise for example a picture of a lion whether the lion was placed in the top right of the image or in the bottom left.

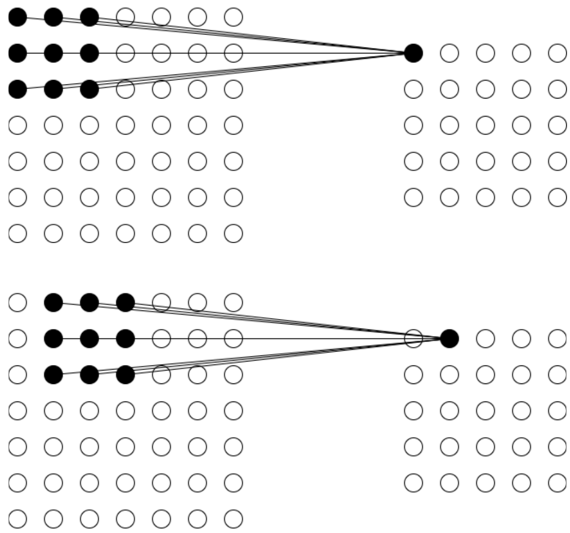

Fig 1: Formation of local receptive field

Pooling layers simplify the output information of the convolutional layers. The pooling layers essentially create a condensed feature map from the convolutional layer [8]. The reason is to reduce the spatial size in order to reduce the number of parameters in the network [6]. Various methods of doing this pooling exist, however the most commonly used method is max pooling using a form of 2x2 filters with a stride of 2 for the depth of the volume [9]. A fully connected layer then connects all the neurons from the pooled layers into a final layer of output neurons.

Various open source libraries are available for creating machine learning algorithms and deep neural networks. TensorFlow if one of these libraries developed by Google and makes it simpler to design, build, and train models. With

TensorFlow, neural networks, and specifically for this project, CNNs, can be created without the need for knowledge about the mathematics behind the networks.

III. Design Science Research

Design Science Research (DSR) is an iterative research paradigm for solving important real-world problems. DSR consists of five steps awareness, suggestion, development, evaluation and conclusion [10]-[13].

The awareness step occurs when researchers become aware of a problem or situation which needs improving. In this particular case, both authors enjoy the African bush and associated game drives and have travelled widely in rural Africa. In conversations with other visitors at the national parks, it became clear that many visitors do not know the names of many of the African mammals.

The suggestion step occurs when possible solutions to the problem are presented. In the case of this research, the second author was using TensorFlow for a medical research application and suggested that TensorFlow would be a possible solution to recognising African mammals.

The development and evaluation steps are typically iterated numerous times until the results are satisfactory.

This project was, in effect, broken into two sub-projects which could run partially concurrently. This is explained in Section IV.

Sections V through VII describe the various iterations with respect to training the model, creating a classifier, and the mobile development.

The final results and the final conclusion steps are documented in Sections VIII and IX.

IV. Two sub-projects

Google has already released an Inception model as a demonstration of how TensorFlow could be used to do image recognition. The Inception model recognises approximately one thousand objects including a number of African animals, animals from other continent, musical instruments, office equipment, etc.

The first sub-project was to learn to create a model which was syntactically compatible with the Inception model but for use with only African mammals. The second sub-project which could run concurrently was to learn to do image recognition on an Android device using the Inception model.

Once the first sub-project started generating its own model which was compatible with Inception, then the Inception model was removed from the second sub-project and replaced with the real African mammal model.

V. Training Iterations

Initially a simple CNN was created using TensorFlow consisting of layers as indicated in Table 1.

This CNN classified 10 types of animals. The system was initially trained with 50 images per animal, obtained from flickr.com. This system performed poorly because a disadvantage of CNNs is that they need to be trained with a large amount of data in order to provide accurate results.

Table 1: Layers of the CNN

Layer	Receptive Field Size	Feature Maps	FC Nodes	Dimensions
Input	-	-	-	128x128x3
Conv	5x5	32	-	128x128x32
Pooling	2x2	-	-	64x64x32
Conv	5x5	64	-	64x64x64
Pooling	2x2	-	-	32x32x64
Conv	5x5	64	-	32x32x64
Pooling	2x2	-	-	16x16x64
Fully Connect	-	-	1024	1024x1
Output	-	-	10	10x1

The system was improved by using 300 images per animal, with a batch size of 50, and with 5000 iterations. This system performed with an accuracy of 56%.

Due to constraints on downloading training images from flicker.com further images were downloaded from Image-Net, which allows access to thousands of images.

Due to the access to large amounts of training data it was decided to upgrade the system from classifying 10 animals to classifying 36 animals, with each animal having a couple hundred available training images. This system was trained with a batch size of 100, and with 10,000 iterations. This system performed with an accuracy of 65%.

To further improve the system it was decided to rather make use of a pre-trained model and fine tune it to our training set rather than use the model which we created.

VI. Pretrained Model Iterations

By using Google's Inception model and retraining it for our training data the accuracy of the system increased significantly as the Inception model was designed for recognizing 1000 items. To tune the model to the system's requirements an additional layer was added to the Inception model. This layer was a fully connected layer connecting all of the previous output neurons to 36 output neurons, one for each animal to be classified. The system was then trained with a batch size of 100 and 16000 iterations. This system performed with an accuracy of 88%.

This system provided high accuracy in recognizing 36 different African animals. The downside to this system however was that the size of the trained model was roughly 100 MB. This is too large to facilitate easy download to a mobile device.

The MobileNet model was fine-tuned and trained in the same manner as the Inception model. This model consists of less layers than the Inception model, meaning less parameters need to be stored. The trained MobileNet model performed with an accuracy of 85% and is 17 MB in size. This is a slight reduction in accuracy for a significant reduction in size.

VII. Recogniser Iterations

The Android development was conducted concurrently to the various training iterations. There were three iterations as indicated

1. The first iteration of the Android app merely used the demonstration Google Inception model and associated libraries. This would ensure that the Android app could recognize objects which were in the Google demonstration Inception model. This was done completely independently of the training iterations described in Section V

2. The second iteration used the first specific model for this research project as described in Section V. This iteration produced an extremely large apk of approximately 130 Megabytes.

3. The third iteration used the MobileNet model. This apk was 35 Megabytes.

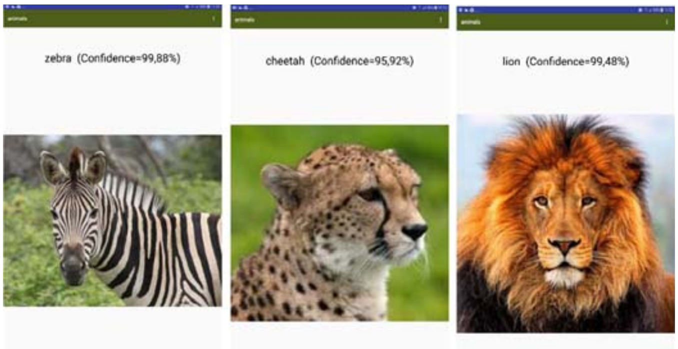

Fig 2: Android app

Three examples of the Android app recognising images can be seen in Fig. 2

VIII. Results

As described in Section IV, there were two sub-projects in this research project: 1) to create a TensorFlow image recogniser for South African mammals, 2) to embed this model in an Android app which tourists could use when a game ranger was not available to accompany them on a game drive. The results of the sub-project #1 were presented in Sections V and VI.

This section only presents results of testing the Android app. The system was tested in three ways: 1) the model itself was tested, 2) the Android app including the model was tested with downloaded images, 3) and the Android app was tested live on a game drive at Pilanesburg in South Africa.

Test #1 was conducted by testing the trained model with 20% of images which were excluded from the training process.

Test #2 (with downloaded images) was conducted by giving the app on a tablet to a person and allowing them to download any images for South African mammals from the internet and run the images through the app.

Test #3 (on a live game drive) was conducted at Pilanesburg Game Reserve in North West province of South Africa.

Test #1 resulted in an accuracy of 85%. This was calculated at the end of the training process. Tests #2 and #3 indicated that the Android app worked well with isolated images of one animal. This was especially true of close up photographs. Iconic photographs of individual elephants, lions, zebras, and giraffes were recognised with 100% accuracy. In addition, animals which are often confused by human beings (such as impala and springbok, or leopard and cheetah, or wild dog and spotted hyena, or kudu and nyala) were correctly identified in isolated images.

However, many animals stay in groups or herds such as impala. In addition, many different types of animals congregate together such as zebra and wildebeest. Some animals are often a distance from the road (or photographer) such as a leopard on a distant rock or a pride of lions under a tree. In such cases the Android app either failed to recognise the animals or misrecognised it.

A subsequent release of the recogniser (outside the scope of this research) took the image and broke it up to smaller tiles and ran the image recognition algorithms on the individual tiles.

IX. Conclusion

Tourism is a major contribution to many African economies. Major national parks attract tourists which, in turn, then purchase petrol, rent accommodation, utilize restaurants, and employ rangers. Some of the smaller parks and game reserves, however, do not have rangers available to assist tourists on their safaris. This research looks at the use of an image recogniser embedded in an Android app to assist tourists at the smaller reserves where no rangers were available.

Images were obtained from fickr.com and image-net.org in order to train a model using Google's TensorFlow. The model was then embedded in the Android app. Testing included

1. Testing of the image recogniser alone

2. Testing of the Android app with images downloaded from the Internet

3. Testing of the Android app during a game drive safari

As explained in the Results Section (Section VIII), there were good and excellent results for tests #1 and #2. In the case of safari (game drive) situations, however, many different types of animals are either congregated together or are quite a distance from the photographer. Additional work is invited in such situations.

Acknowledgments

The authors thank Image-Net (http://image-net.org) for providing thumb nails and URLS of images which were used for training the final model. In addition, the authors thank the

people who contributed their photographs to flickr.com under a Creative Commons license.

References

[1] R. Turner and E. Freiermuth, "Travel and tourism economic impact 2017 South Africa," World Travel and Tourism Council, 2017.

[2] R. Turner and E. Freiermuth, "Travel and tourism economic impact 2017 SADC," World Travel and Tourism Council, 2017.

[3] L. Rabiner and B. Juang, "An introduction to hidden Markov models," *Ieee Assp Magazine,* vol. 3, *(1),* pp. 4-16, 1986.

[4] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst. Man Cybern.,* vol. 21, *(3),* pp. 660-674, 1991.

[5] H.C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R.M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Imaging,* vol. 35, *(5),* pp. 1285-1298, 2016.

[6] N. Michael, "Neural networks and deep learning," 2015.

[7] S.C.B. Lo, H.P. Chan, J.S. Lin, H. Li, M.T. Freedman, and S.K. Mun, "Artificial convolution neural network for medical image pattern recognition," *Neural Networks,* vol. 8, *(7),* pp. 1201-1214, 1995.

[8] O. Abdel-Hamid, A.R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 22, *(10),* pp. 1533-1545, 2014.

[9] F. Li, A. Karpathy and J. Johnson, "CS231n: Convolutional neural networks for visual recognition," *University Lecture,* 2015.

[10] V. Vaisnavi and W. Kuechler, "Design Science research in information systems," vol. June 23, 2013, 2013.

[11] A. R. Hevner, "The Three Cycle View of Design Science Research," *Scandinavian Journal of Information Systems,* vol. 19, *(2),* pp. 87-92, 2007.

[12] V. Vaishnavi and W. Kuechler, *Design Science Research Methods and Patterns: Innovating Information and Communication Technology.* 2007.

[13] R.H. Von Alan, S.T. March, J. Park, and S. Ram, "Design science in information systems research," *Management Information Systems Quarterly,* vol. 28, *(1),* pp. 75-106, 2004.