Deep learning for annotating ecological images: an example using Optical Character Recognition

Ben Weinstein1

1Department of Fisheries and Wildlife, Marine Mammal Institute, Oregon State University, 2030 Marine Science Drive, Newport, OR 97365, USA

# Abstract

1. Image capture is playing an increasing role in ecological analysis and biodiversity monitoring. The growth of image capture is limited by the significant time investment for annotating images with metadata.
2. Machine learning tools for image classification are growing in power and complexity, but are largely unavailable to ecologists without significant technical expertise.
3. Google CloudML is a new machine learning service from Google that combines easy access to convolutional neural networks and a robust data environment for massive parallelization.
4. I retrained a machine learning classifier to recognize text within ecological images captured during time-lapse video recordings of hummingbird-flower interactions.
5. The classifier showed greater than 98% accuracy and increased the speed of image review and annotation at minimal costs. The future of ecological image analysis will combine text annotation, species recognition and human collaboration for ecological image annotation at unprecedented scales.

Keywords: Google, character recognition, cloud computing, eco-informatics, image analysis

# Introduction

To monitor biodiversity, ecologists and conservation managers need high quality temporal and spatial data on animal presence, movement, and behavior. The high costs, complex logistics, and significant expertise of human observation studies limit the ability to gather ecological data. Ecologists are increasingly turning towards greater automation to facilitate biodiversity monitoring (Pimm *et al.* 2015). The decreased costs of image capture and the growing library of analysis tools makes automated biodiversity monitoring closer than ever before (Pennekamp & Schtickzelle 2013; Dell *et al.* 2014; Troscianko & Stevens 2015).

A major obstacle to the growth of image-based ecological analysis is the speed of scoring and extracting information from images. Annotating images with metadata, such as time and species identity, requires significant labor investment. The emerging field of computer vision can decrease the time for image annotation, increase consistency among annotators, as well as engage less experienced observers in biodiversity monitoring (Swanson *et al.* 2016). Computer vision is a field of image-based computer science that uses image pixels to mimic human perception based on image characteristics, shape and sequence.

While computer vision has made incredible strides in a variety of fields, its growth in ecology has been slowed by a lack of access to high level algorithms. Computer vision publications are often short, terse, and filled with jargon. Given the potentially lucrative opportunities for new algorithms, source code is rarely made available. The challenge lies in adapting existing tools to meet the technical ability and budget limitations of ecologists.

The recent unveiling of the Google Cloud Machine Learning platform (Google CloudML) could be a quantum leap in access to machine learning tools. Released in 2016, Google CloudML gives users access to a robust web service to retrain models using Google’s popular TensorFlow software. TensorFlow is a computational graph algorithm that represents mathematical operations as nodes and stores data in multidimensional arrays (‘tensors’). For image analysis, the main model components are a set of images with known labels, a model to compute a goodness of fit score, and a set of rules on updating model parameters during training. The model is then run for a fixed number of iterations, updating the model weights based on the minimizing the error in predicting a withheld portion of the training dataset (‘testing data’). Known colloquially as deep learning, neural networks differ from traditional machine learning tools because features are not apriori used to describe each class, rather the model teaches itself which features best delineate the training and testing data (LeCun *et al.* 2015). In particular, convolutional neural networks reduce the dimensionality of images through repeated downscaling and filtering of image pixels to create unique combination of image features. For a rigorous treatment of convolutional neural networks for image analysis see LeCun *et al.* (2015). For most ecologists, the beauty of the recent improvements is that the black box of neural networks needs not be delved too deeply.

While there have been technical applications of deep learning for ecology (Chen *et al.* 2014; Gomez *et al.* 2016), it is the availability of highly sophisticated tools for average users which holds revolutionary potential. Rather than building a model from scratch for each application, users can retrain pre-built models to add new image classes. Known as transfer learning, this approach uses the strengths of the underlying architecture, but adds flexibility for specialized problems. This greatly reduces the time and expertise needed to implement image analysis solutions. Through transfer learning, ecologists with ample labeled data can teach models new classes and develop specific solutions for their taxa and location.

## A sample problem: extracting metadata from wildlife images

In 2013, my colleagues and I began placing time-lapse cameras in the montane cloud forests of Ecuador to monitor hummingbird-plant interactions. Ecological networks, and plant-pollinator networks in particular, are chronically under-sampled due to the rarity of many interacting partners and the seasonal pattern of flower phenology. By using cameras that turned on at 6am, turned off at 6pm, and took 1 photo per second, we collected large volumes of video data. We currently have twenty cameras generating nearly 100 GB of video data per month, with plans of increasing to >50 cameras across multiple sites. The videos are screened using computer vision software to find important candidate motion events, which reduces the total number of reviewed frames by >90% (Weinstein 2015). These candidate frames are then reviewed by a human to record species identity, the time, date, and behavior of individuals (Fig. 1). Manual review of frames still is a large bottleneck in generating datasets and limits the scalability of our approach.

Using Google CloudML, my aim was to retrain an existing convolutional neural network to identify the printed characters in the images. This task is known as optical character recognition (OCR), and is a well-researched application of machine learning. For comparison, I analyzed the same dataset with Tesseract OCR, a commonly used open-source software. By evaluating the accuracy, obstacles and costs of each approach, I tested of the efficacy of these tools for ecological image analysis.

# Materials and Methods

Neural networks are a series of mathematical layers that connect combination of predictors to create a pathway from input data to output prediction (Fig. 2). Input data pass through the first hidden layer and receives a new value:

where Z2 is the 2nd layer in the network and W1 is the weight of the connection between the input data and the first hidden layer. The data is given a specific notation X, rather than Z1, to denote that it is a fixed quantity. The support for this pathway through the network is called the activity (a), and is calculated at Z2:

Common activation functions include sigmoid:

or rectifier functions:

The same process is repeated for each layer of hidden network.

The last step is to compute the predicted value given the network weights at the final hidden layer.

Given this generic structure, training the model involves finding the values of Wn which maximize the predictive value of for our data X. At each training step, new weights are chosen to increase predictive value. These changes cascades through the network, with all connected weights updated to reflect new values farther up the network. This updating process continues for a preset number of steps, optimally until each new step produces no increase in predictive value.

This conceptual explanation intentionally glosses over the rich quantitative complexity of finding optimal model parameters and structure. The number of layers, the form of the activation function, and the connections among layers are all active areas of research. For additional information on neural networks in ecology see Gomez *et al.* (2016); Marburg & Bigham (2016); Qin *et al.* (2016).

## Model Training

To generate a training dataset of numbers and letters, I wrote a python script to identify the location of letters within existing annotated images. This script split letters based on their outline and processed the images to increase size and clarity (Fig. 2). All images were captured using a Plotwatcher Pro camera (Day 6), and therefore had the camera ID, date, and time in same position within the image. I then manually coded the known letters based on existing image metadata that had been collected earlier in the project. I split the 7511 character images into 85% training and 15% testing data, with a maximum number of 300 images per class. From my existing stock of images I was able to recover digits 0-9, semicolon, forward slash, and the letters (A, C , E , F , H , L , M , O , P , R , T , W , Y). The remaining letters were not found in available images, and were not included as potential classes.

I ran the model training for 1000 steps, minimizing the cross-entropy loss at each step:

where the loss is the sum of the input training images (x(i)) labeled with class (y(i)), with an activation score of the neural network at each node (a). Through iterative updating of the model weights, the cross-entropy is reduced based on the success of predicting the withheld portion of the training data. To determine if the model has had sufficient time for training, I ran training until the rate of improvement in predictive accuracy reached a plateau and did not yield better classification with additional steps. All source code can be found at the Github repository online (<https://bw4sz.github.io/MeerkatReader/>) (Appendix 1).

I compared the performance of my neural network with Tesseract OCR, a highly regarded open-source tool for transcribing text from scanned documents. The same input letters used to evaluate the neural network were run through Tesseract with default language set to English and page segmentation turned off to reflect identifying a single character (flag, --ps 10 in command line interface). Tesseract was run from pytesseract python module (see source code available on the [github](https://bw4sz.github.io/MeerkatReader/)).

# Results

The neural network proved to be relatively fast, reasonably inexpensive and highly accurate (Table 1) (Fig. 3). The overall accuracy rate was 98.3%, with the majority of misclassifications relating to poor parsing based on the python pre-processing. Running with multiple machines in parallel, modeling train took 9.5 minutes, and prediction of the 738 images took 14.1 minutes. Of the few errors in predicting characters, most were single events among highly similar letters in the image font (1 versus 7, P versus R). Overall there were no systemic problems with character recognition using the trained model.

Tesseract OCR performed poorly, with an average accurate rate of 53% (Fig. 4). Errors were distributed widely over character classes, with systemic errors in misidentifying both letters and numbers. These errors were often difficult to interpret, with strongly different letters resulting in similar predictions. For example, P, A, and 6 were all commonly predicted to be capital I, despite obvious visual differences among images.

# Discussion

Image classification through machine learning has the potential to revolutionize biodiversity monitoring (Beijbom *et al.* 2012; Van Horn *et al.* 2015). Researchers can spend hundreds of hours reviewing video footage, or sorting through thousands of camera trap photos (Swanson *et al.* 2015; Weinstein 2015; Gaglio *et al.* 2016). Automated image annotation will act in concert with new data gathering techniques to increase the scale and efficiency of ecological monitoring (He *et al.* 2016). This analysis is the first test of the cloud machine learning services provided by Google for ecological image analysis. With robust documentation, and a growing user community, and low costs, ecologists will greatly benefit from further involvement with these tools.

The sample problem was chosen because it solved a pressing data management problem, and could be tested against existing optical character recognition tools. The trained machine learning model eclipsed the off-shelf performance of Tesseract OCR for character recognition. This is not meant to disparage Tesseract, which was designed for text scanning and natural language processing, rather than image recognition. While the neural network results are impressive, they are not surprising in the sense that once a model has been correctly trained, there should be no variance in the way which characters are printed on an image. In more challenging tasks image classification tasks, it is of paramount importance that training data represent the full breadth of potential image features. For example, if I was attempting to distinguish between two species of woodpeckers, and the species were always photographed in different habitats, there would be a danger in the model learning the difference in the image background, rather than the target bird, yielding an overfit model that would have poor predictive performance in new photos (Van Horn *et al.* 2015).

From optical character recognition to species level identification, there is little known about the limits of these developing tools. Past neural network applications have relied on users with specialized knowledge and access to high performance hardware (Chen *et al.* 2014). Past work has yielded impressive results in species level taxonomy in a wide array of taxa, including savannah animals (88.9% accuracy, Gomez *et al.* 2016), benthic invertebrates (89.0%, Marburg & Bigham 2016), and coral (88.9% accuracy, Beijbom *et al.* 2012)). The next step is to systemically review the conditions for when machine learning tools will be successful in ecological images. What is the level of phenotypic differentiation needed to train neural networks to identify animals within images? What are the optimal conditions for image capture and pre-processing? Where can we leverage the natural excitement of citizen scientists to improve deep learning results? This final question could have added benefit of engaging the public in scientific image analysis, yielding both high quality data and conservation outreach (Kosmala *et al.* 2016). Further exploration of these tools will help push ecology towards greater automation and increased sampling in space and time.

# Acknowledgements

The author acknowledges no conflict of interest. I want to thank the engineers at Google Cloud Platform for their encouragement and assistance.

# Literature Cited

Beijbom, O., Edmunds, P.J., J., Kline, D.I., Mitchell, B.G., Kriegman, D., Kline, D.I., Mitchell, B.G. & Kriegman, D. (2012). Automated annotation of coral reef survey images. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1170–1177.

Chen, G., Han, T.X., He, Z., Kays, R. & Forrester, T. (2014). Deep convolutional neural network based species recognition for wild animal monitoring. *2014 IEEE International Conference on Image Processing (ICIP)*, 858–862.

Dell, A.I., Bender, J.A., Branson, K., Couzin, I.D., de Polavieja, G.G., Noldus, L.P.J.J., Pérez-Escudero, A., Perona, P., Straw, A.D., Wikelski, M. & Brose, U. (2014). Automated image-based tracking and its application in ecology. *Trends in Ecology and Evolution*, **29**, 417–428.

Gaglio, D., Cook, T.R., Connan, M., Ryan, P.G. & Sherley, R.B. (2016). Dietary studies in birds: Testing a non-invasive method using digital photography in seabirds. *Methods in Ecology and Evolution*.

Gomez, A., Diez, G., Salazar, A. & Diaz, A. (2016). Animal Identification in Low Quality Camera-Trap Images Using Very Deep Convolutional Neural Networks and Confidence Thresholds. *Advances in Visual Computing: 12th International Symposium, ISVC 2016, Las Vegas, NV, USA, December 12-14, 2016, Proceedings, Part I*, pp. 747–756. Springer International Publishing.

He, Z., Kays, R., Zhang, Z., Ning, G., Huang, C., Han, T.X., Millspaugh, J., Forrester, T. & McShea, W. (2016). Visual Informatics Tools for Supporting Large-Scale Collaborative Wildlife Monitoring with Citizen Scientists. *IEEE Circuits and Systems Magazine*, **16**, 73–86.

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P. & Belongie, S. (2015). Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 595–604.

Kosmala, M., Wiggins, A., Swanson, A. & Simmons, B. (2016). Assessing data quality in citizen science - preprint. *Frontiers in Ecology and the Environment*, **14**, 551–560.

LeCun, Y., Bengio, Y., Hinton, G., Y., L., Y., B. & G., H. (2015). Deep learning. *Nature*, **521**, 436–444.

Marburg, A. & Bigham, K. (2016). Deep Learning for Benthic Fauna Identification. *OCEANS 2016 MTS/IEEE Monterey*, 1–5.

Pennekamp, F. & Schtickzelle, N. (2013). Implementing image analysis in laboratory-based experimental systems for ecology and evolution: a hands-on guide. *Methods in Ecology and Evolution*, **4**, 483–492.

Pimm, S.L., Alibhai, S., Bergl, R., Dehgan, A., Giri, C., Jewell, Z., Joppa, L., Kays, R. & Loarie, S. (2015). Emerging Technologies to Conserve Biodiversity. *Trends in Ecology & Evolution*, **30**, 685–696.

Qin, H., Li, X., Liang, J., Peng, Y. & Zhang, C. (2016). DeepFish: Accurate underwater live fish recognition with a deep architecture. *Neurocomputing*, **187**, 49–58.

Swanson, A., Kosmala, M., Lintott, C. & Packer, C. (2016). A generalized approach for producing, quantifying, and validating citizen science data from wildlife images. *Conservation Biology*, **30**, 520–531.

Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A. & Packer, C. (2015). Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Scientific Data*, **2**, 150026.

Troscianko, J. & Stevens, M. (2015). Image calibration and analysis toolbox - a free software suite for objectively measuring reflectance, colour and pattern. *Methods in Ecology and Evolution*, **6**, 1320–1331.

Weinstein, B.G. (2015). MotionMeerkat: Integrating motion video detection and ecological monitoring. *Methods in Ecology and Evolution*, **6**, 357–362.

Figures



Figure 1. A sample image of a female Violet-tailed Sylph (*Aglaiocercus coelestis*) visiting a *Guzmania jaramilloi*. At the bottom of each image is a camera ID, date and time stamp.

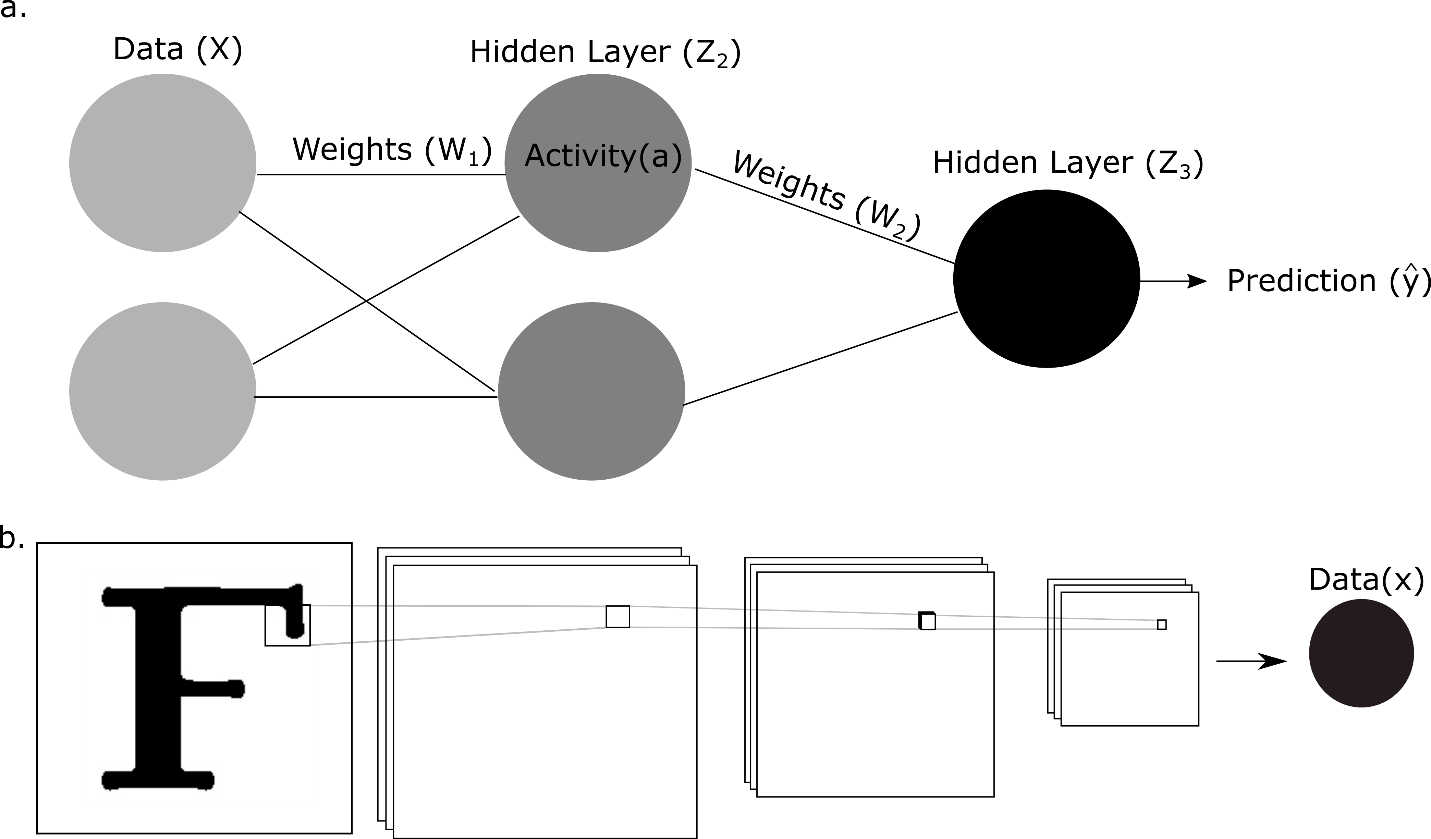


Figure 2. Conceptual overview of a neural network for machine learning prediction: a) a generic deep learning structure, input data passes through hidden layers, called nodes, to create pathways from predictors to prediction. The activation score at each of these nodes is used to estimate model weights, which sum to create a fully connected network. In many deep learning applications, there will be many layers of nodes to create combinations of input predictors. b) Example image of class “F” illustrating the pixel convolutions for downsampling and pooling image features to generate input predictors.

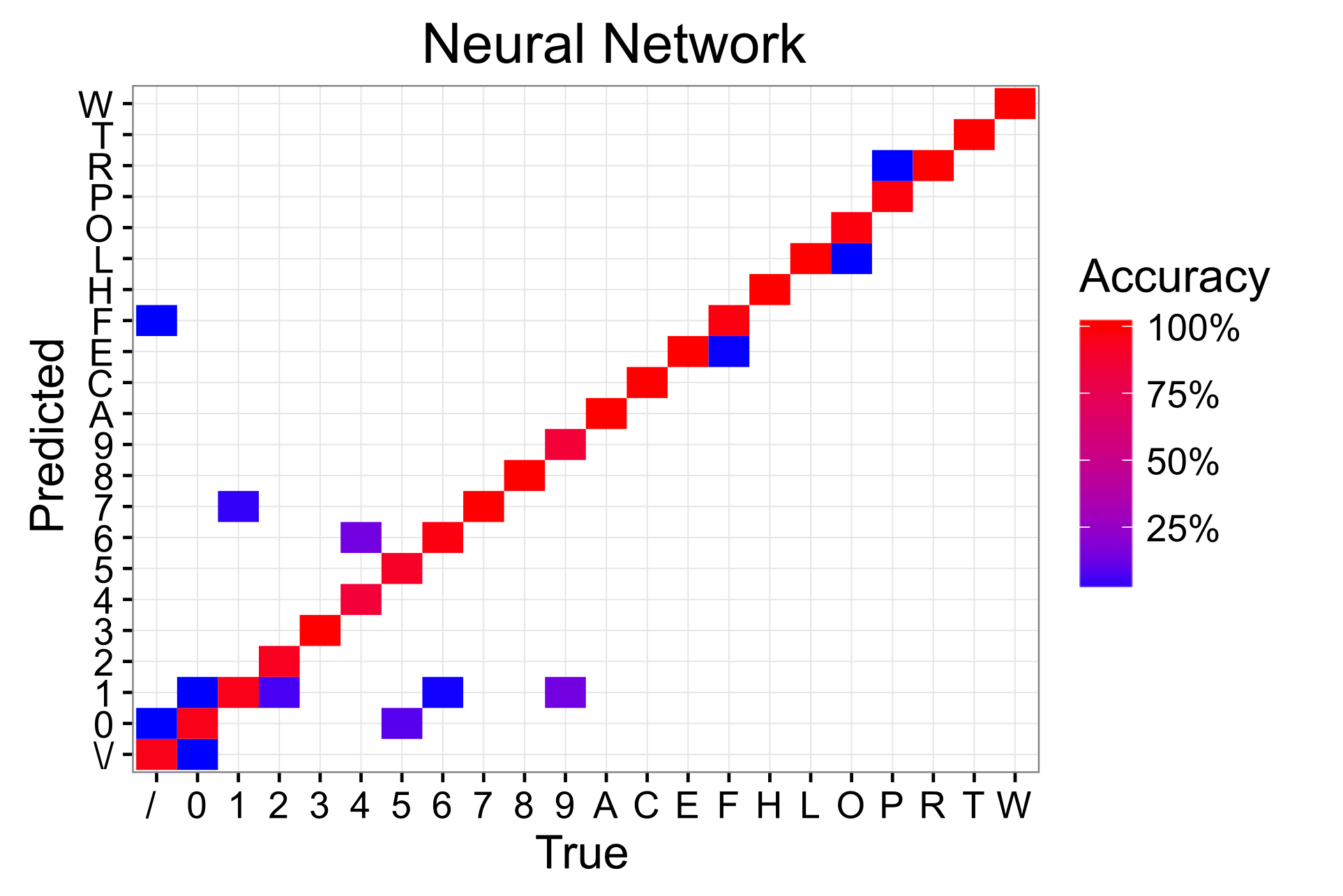


Figure 3. Confusion matrix for the neural network trained on Google’s CloudML web service. The known true label is compared to the predicted label for each of the 738 testing images.

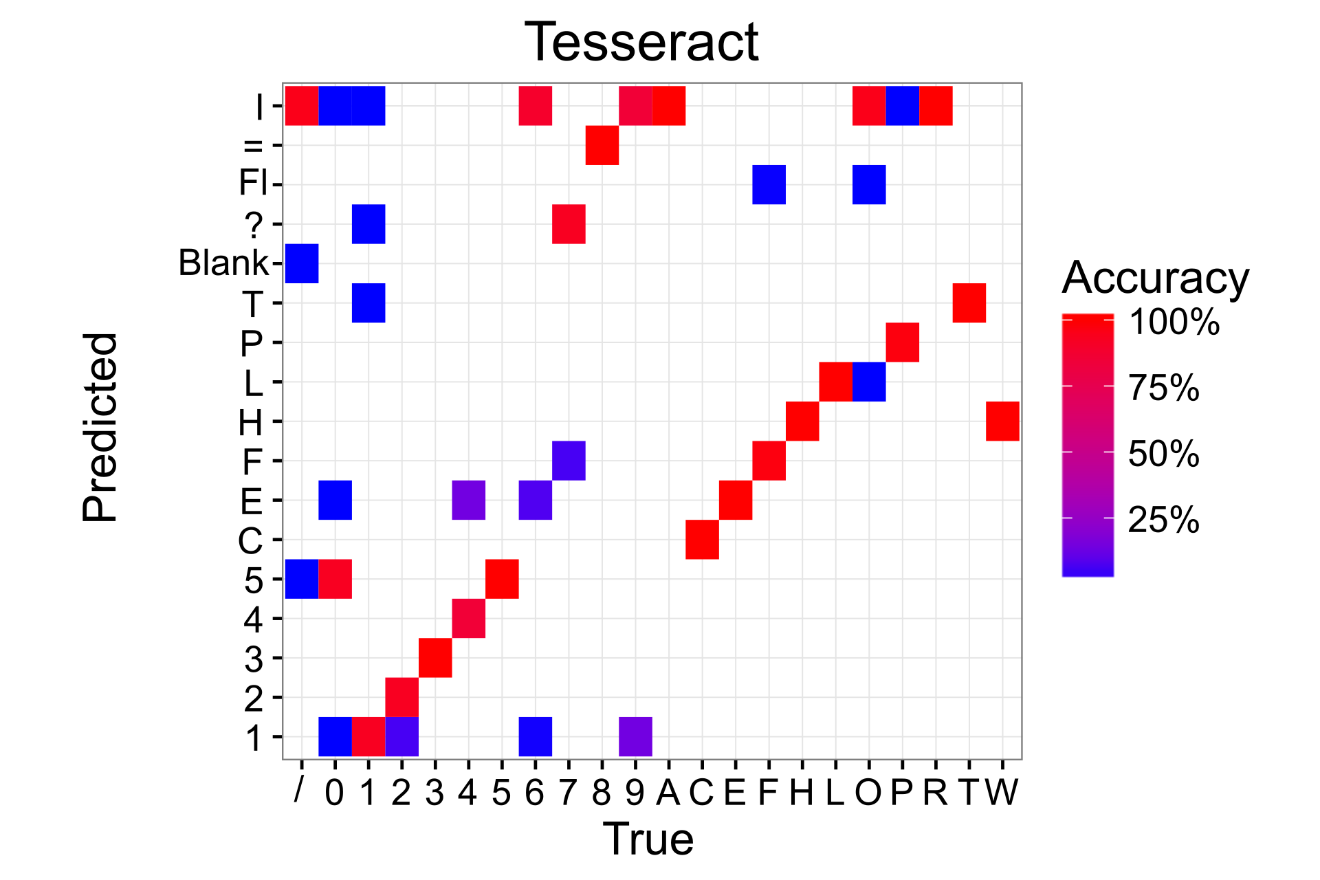


Figure 4. Confusion matrix for Tesseract OCR, a commonly used tool for optical character recognition. The known character label is compared to the predicted character label for each of the 738 testing images.

Table 1. Performance for the neural network trained with Google CloudML and the Tesseract OCR model for predicting characters from images. Cost includes the computational time for training, prediction, and storage of image data on Google Cloud Storage buckets. All computation was run on default Google Cloud compute instances with 1 CPU and 3.75 GB of virtual memory.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Training Time | Prediction Time (per Image) | Accuracy | Cost |
| Neural Network | 9.5 min | 2 to 3 seconds | 98.3% | $38.75 |
| Tesseract OCR | - | < 1 second | 53.0% | - |

Appendix 1

Using the Google Cloud Vision Text Detection API

Users familiar with the Google Platform may be point out that Google offers a cloud-based scene text detection service that allows users to directly upload images and returned processed text. The service is designed for identifying text in real-world images (house numbers, street signs, etc.), and did not perform well in this test case. For the individual extracted letters the service almost never returned any meaningful results. Most returns calls were empty. Feeding the original image into the service achieved more tangible results, but they were not on par with the text detection analysis based on a trained neural network model or tesseract OCR. The high performance of the neural network rendered further investigation of scene text detection unnecessary.