Classification of TERC Photos: A Space Odyssey

TERC1: Dimitri Makrigiorgos, Neela Kaushik, Sarah Ferry
dmak1112@bu.edu, nkaushik@bu.edu, ferrys@bu.edu

Project Task

The objective of this project is to provide the TERC Windows on Earth team with a program that can analyze the images from the ISS and automatically tag them with 12 basic tags with a high level of accuracy. TERC receives thousands of images every day but are only able to manually tag less than 1% of these images. A machine learning, object recognition implementation that achieves high tagging accuracy would greatly facilitate their efforts to build a searchable database of images for research and education purposes.

Related Work

Convolutional neural networks are the standard machine learning approach for image recognition. Additionally, there exist pre-trained image recognition neural networks such as ResNet50 or Inception V3 that are often used as a starting point and then fine-tuned to fit unique models. [1]

Approach

The neural network architecture we initially used is based on an out-of-the-box ResNet50 implementation pre-trained on ImageNet. [2] The architecture is implemented in Python using the Keras library built on top of TensorFlow. We freeze the standard top layers for ResNet50 and replace the output layer with a new layer of density twelve to account for the twelve categories we are interested in classifying (Figure 1). The classification model we employ uses the sigmoid activation function and a binary cross-entropy loss defined as:

$$L(p) = -(y\, log(p) + (1-y)\, log(1-p))$$

Given an image, the model outputs a probability for each of the 12 tags. For each of the tags, if the probability is higher than the predetermined threshold equal to 0.5, then the model predicts that the tag exists (i.e. assigns it a 1, else assigns it a 0). The final output is a .csv file that contains each image name and its predicted tags.

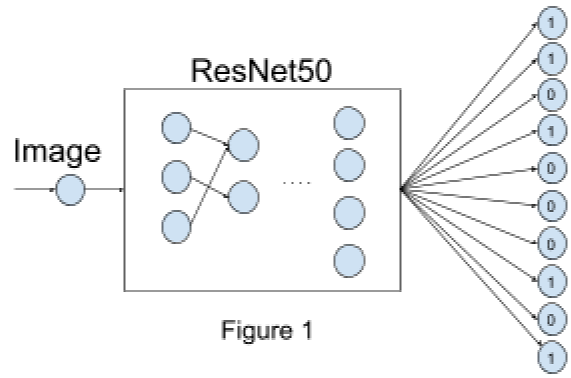

Figure 1

Dataset and Metric

The training data consists of 8,130 images provided by the project partner. Each image is resized to 224 x 224. Images are shuffled and randomly assigned to training, validation and testing sets at a 60/20/20 split. Image labels are obtained from the EXIF metadata and stored in a .csv file using a categorical encoding scheme 1=tag exists, 0=tag does not exist for each of 12 tags ["Volcano", "Sunrise Sunset", "ISS Structure", "Stars", "Night", "Aurora", "Movie", "Day", "Moon", "Inside ISS", "Dock Undock", "Cupola"] for each image. Images and their labels are vectorized and passed into the neural network for processing. For example, if we had an image with tags Night and Aurora, we would extract those tags from the image metadata, and in the csv file the resulting entry would be [0,0,0,0,1,1,0,0,0,0,0,0], where the 1 represents that the tag exists and the 0 represents the tag does not exist.

Our Project Partner gave us a baseline goal of 80% accuracy for each tag. We hope to show that our method can achieve accuracy higher than 80% achieved by the baseline method for each individual tag.

We will also use binary accuracy and the F1 score to measure the success of our model. We hope to show that our model can achieve 90% accuracy using both of these metrics. Binary accuracy is a metric predefined in the Keras package that reports the mean accuracy over all labels. The F1 metric is a popular accuracy score in classification tasks that takes into account both the model's precision and recall. Precision is the true positive rate, which is defined as:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall is the fraction of tags that we identify as positives in the image, defined as:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

F1 is the harmonic mean of precision and recall, defined as:

$$F1 = 2\,\frac{Precision * Recall}{Precision + Recall}$$

Preliminary results

We trained on 4877 samples and tested on the validation set of 1625 images. Our batch size is 16 and we run 10 training epochs. At present, this model yields the accuracies tabulated below:

	Accuracy	
	Binary	F1
Epoch 1/10	0.9254	0.4735
Epoch 2/10	0.949	0.7031
Epoch3/10	0.9629	0.7872
Epoch 4/10	0.9873	0.9313
Epoch 5/10	0.9893	0.9427
Epoch 6/10	0.9902	0.9474
Epoch 7/10	0.9909	0.9509
Epoch 8/10	0.991	0.9522
Epoch 9/10	0.9912	0.9532
Epoch 10/10	0.9915	0.9549

Tag	Accuracy
Volcano	0.9945
Sunrise Sunset	0.9920
ISS Structure	0.9865
Stars	0.9938
Night	0.9908
Aurora	0.9994
Movie	0.9815
Day	0.9809
Moon	0.9982
Inside ISS	0.9938
Dock Undock	0.9908
Cupola	0.9963

We obtain very high accuracies when using both metrics (>0.99 and >0.95, respectively). Additionally, our model predicts every tag correctly for an image (i.e. with no false negatives or false positives) in 0.9243 of the validation samples.

Timeline and Roles

Task	Deadline	Lead
Insert tags back into EXIF Metadata	12/03/17	Sarah Ferry
Generate results for different learning rates and layers to optimize accuracy	12/03/17	Neela Kaushik
Create a trained model that can be used by the project partner	12/07/17	Dimitri Makrigiorgos
Prepare report and presentation	12/12/17	All

References

[1] Gardner, D., & Nichols, D. (2017, July 02). *Multi-label Classification of Satellite Images with Deep Learning* (Publication). Retrieved http://cs231n.stanford.edu/reports/2017/pdfs/908.pdf

[2] Yu, F. (2016, October 03). A Comprehensive guide to Fine-tuning Deep Learning Models in Keras (Part I). Retrieved from https://flyyufelix.github.io/2016/10/03/fine-tuning-in-keras-part1.html