Classifying Geographical Features in Astrophotography

Andrew Chen
University of Texas Rio Grande Valley
Department of Computer Science
Edinburg, Texas
andrew.chen01@utrgv.edu

Paulina Varela
University of Texas Rio Grande Valley
Department of Computer Science
Edinburg, Texas
paulina.varela01@utrgv.edu

***Abstract*—The Gateway to Astronaut Photography of Earth is a database that archives all photographs and related data from or about space since 1961. There are millions of photos in this database, many of which require annotation. Proper annotation is vital for an archive to have effective searching and sorting. This paper will explore the recent work in the field of computer vision and propose a deep learning model to help classify geographical features in astrophotography.**

I. Introduction

Image processing has always been a hard problem to digitalize. Is it possible use a computer to see what humans see? The brain gives humans the capacity to recognize text and read or differentiate between a monkey and another human. While these tasks are fathomable to humans, it has been a difficult problem to solve using computers. However, in the recent years, there has been major progress in the field of computer vision with the rediscovery of deep learning methods. Groundbreaking work with deep convolutional neural networks has shown that, with the current potential available from computer hardware, computers have matched or exceeded human performance in some domains of visual recognition tasks. As humans, we are able to subconsciously process images at any given point all day. How is it possible for us to see, label, classify, and recognize patterns? The primary visual pathway is the communication between our eyes and our brain that allows for us to see and observe the environment around us. Our eyes have receptors that receive stimuli from light bouncing from objects and use that as the input image or visuals to be processed. While this seems natural to us, there is an incredibly deep and complex hierarchical network of neurons under the hood that handles processing of our vision, allowing us to recognize, label, and classify objects. There is also an inherent method of learning involved. Our brains learn by example about the things that they havent encountered before. Eventually, the brain recognizes patterns in what the eyes receive. If the brain is able to learning, then with sufficient training, it will be capable of forming models or schemas of the environment.

The architecture of neural networks are modeled after the human brain. This research dates back to the 1950s and 1960s where D.H Hubel and T.N Wiesal were able to study and model the brain of mammals [2]. They were able to demonstrate that the neurons in the visual cortex of a cat and monkey responded directly to the perceived environment. Hubel and Wiesal discovered that there were two types of visual neuron cells, simple cells (S cells) and complex cells (C cells) [2]. In vision, there is the notion of a receptive field of a single sensory neuron. This field highlights the region of retina which activates the neuron. To keep the explanation within the scope of this survey, they were able to see that the hierarchal structure of how the neurons operates played a significant role in the brains ability to process images. Fukushima was the first to implement a neural network, which he coined as the Neocognitron, that was inspired by the hierarchical structure and concept of simple and complex cells. This was the first neural network capable of learning how to recognize patterns and objects.

The first convolutional neural network was pioneered by Le Cun, Bengio, Bottou, and Haffner for their work with LeNet-5. Their network shed some light on CNNs, a special kind of multi-layer neural networks that, like most neural networks, are trained with some form of back-propagation. CNNs are engineered to, with minimal preprocessing, take an image input and recognize visual patterns. The group produced LeNet-5 which was able to take in images of hand-written numbers and classify each digit.

LeNet-5 architecture is fundamental in this field because it provided insight about images and their features. Their methods showed that image features are embedded within the entire image and that its possible to use convolutions to learn and find these features in all possible locations of the image. The group was able to revolutionize the area of deep learning with their network. Its features include using convolution to extract spatial features, non-linear transformations in the form of tanh or sigmoids, sparse connection matrix between layers to compensate for high computation costs, subsampling using spatial average of maps, and a convolutional layer consisting of 3 layers: convolution, pooling, non-linearity.

The research was conducted around 1998, so there werent any powerful GPUs and CPUs were still slow. This forced the group to architect the model to save parameters and computation to compensate for the lack of potential in hardware at the time. The field of deep learning fell silent as most research made little progress. It was not until movements like the spread of smartphones that everyone had access to the web and an affordable quality camera. With a computer in everyones pockets, more and more data has spread and become readily available. Computer power has increased significantly with the recent advancements in computer architecture and hardware. CPUs have become faster and more powerful. Since GPUs have become widely available, they have also been adopted as a general-purpose computing instruments. The trends in increased data and computation are what allowed for the field of deep learning and neural networks to boom again.

For the past six decades, government space agencies have been sending astronauts to space. While in space, astronauts conduct various experiments and gather as much data as they can about the type of environment they are in. For the past six decades, they have also been manning astronauts with powerful cameras that are able to capture images of space, including detailed images of earth, as well as the moon. As the years went, the camera equipment aboard the International Space Station became more powerful, allowing for more detailed imagery. While astronauts were taking images on the scale of a thousand to five thousand per mission, they were not recording what the images were, nor what they included, in terms of geographical features.

As this database grew in size, the manpower and annotation need to make it a successful and useful archive became more exponential. In 2014, there were almost two million images in the database [9]. While there is no released number of the current number of images in the database, the database continues to grow. While there are some people working on annotating the images, describing the contents, the location, and various geographical features, the whole process is tedious and in need a efficient and effective solution to improve the time and accuracy it takes to categorize an image. Doing so would help the images be useful and of relevance to people as well as researchers in need of spatial imagery. Whether it be just to admire the Earth from space or to analyze the vegetation, climate change, or witness and report natural disasters, organizing the data is the first step to a more useful archive that will be of value and importance to many people.

II. Deep Learning using Convolutional Neural Networks

The basic mechanism behind a neural network is taking an input through hidden layers. These layers are columns of neurons which are fully connected to every neuron in the preceding layer. After the input goes through all the layers, it will reach a last fully connected layer where the output will be predicted. The final classifier layer contains classification scores for all possible predictions. To summarize, neural networks receive a vector in parameters as input, take the input through a series of hidden layers of neurons that are fully connected to all neurons in previous layers, and then through a final fully connected layer containing scores for each classification.

Convolutional neural networks are similar to regular neural networks like the one discussed above. The layers contain neurons that have weights and can learn. The neurons in each layer receive inputs from the preceding layer, performs the dot product, followed by an optional non-linearity. The main architectural differences of a CNN vs a regular neural network comes from the fact that CNNs are explicitly designed to take an image as the input. This allows for the design of the network to be geared towards certain desirable properties which can reduce the amount of parameters needed and allow for have an efficient feed forward function.

An image input does not scale well with traditional neural networks. The image can be represented as a matrix of pixels with the dimensions height and width. A black and white image can be represented with an additional dimension to consider the grayscale for each pixel, giving the input a 3x3x1 size. On the other hand, a full colored image can be represented with an additional dimension to consider each of the red, green, and blue values of the pixels, giving the input a size of 3x3x3. Regular neural networks have trouble compensating for the scale of the image input size. The CIFAR-10 dataset contains images that are only 32x32x3 in size so the first hidden layer of fully-connected neurons in a typical neural network would have 3072 weights which may seem trivial, but an RBG image with 256x256 width and height would produce 196,608 weights. If a network needs to consider more neurons, the number of parameters of input for each layer would increase significantly. To summarize, the connectivity of a traditional neural network can be costly for image processing and the size of parameters would result in overfitting.

Convolutional neural network architectures are geared towards handling the nature of an images input. A CNNs layers have a 3-dimensional structure to consider: width, height, and depth. Rather than having all the neurons connect to the neurons in the preceding layer, only a small region is connected. By the time the input reaches the final layer, it will have been transformed into a single vector of scores for the possible classification or prediction. Each layer between the input and the final output layer will be a series of hidden layers that have these basic components: convolutional layer, rectified linear unit (ReLU) layer, or a pooling layer. The two main tasks that are done by the layers of a CNN are: using hidden layers to extract features and using the final layer as a classifier.

- Hidden Layers
  - Convolutional Layer: finds the output for each neuron which is connected to a small region in the input layer by computing the dot product of their weights and the region they are connected to from the input layer neurons.
  - Rectified Linear Unit (ReLU) Layer: applies an elementwise activation function
  - Pooling Layer: uses the spatial dimensions and applies a downsampling operation
- Fully Connected Layer
  - The final layer of the network. It is responsible for

computing the scores amongst the different possible classifications. Using the CIFAR-10 dataset as an example, the output of this layer would be in the size 1x1x10. The depth is 10 because there are only 10 possible classifications in CIFAR-10.

These neural networks can extract features through a series of convolutions, the main building block of a CNN. The formal definition of a convolution is producing a 3rd function from the combining and merging two sets of information via dot product. In the context of these networks, convolutions result in feature maps which are produced by running a filter through the input data. This filter is inspired by our neuron cells by acting as a receptive field with a typical size of 3x3. The filter runs and slides through the input matrix, performs matrix multiplication, sums the results, and adds it to the outgoing feature map. Several convolutions are performed on the input using different filters. Each of these filters results in a different feature map which is then added together in the final output layer. Like traditional neural networks, the output of the hidden layers must be non-linear. This non-linearity is achieved by utilizing an activation function such as ReLU. A stride is used to determine how big of a slide a convolutional filter should make. They are typically 1, so the filter slides pixel by pixel. It is possible to increase your stride size to slide a larger interval to achieve less overlap. The size of a outputted feature map is smaller than the input size so, to prevent the feature map from shrinking as it goes through the layers, padding is utilized. Pooling layers are useful to shorten training time and to consider overfitting. Pooling layers are commonly found in between convolutional layers. They are placed to reduce the dimensions of the size of parameters and can therefore help with reducing computation needed. A common method of pooling is max pooling, which is the best way to decrease the feature map size and still consider the most significant information by keeping only the maximum value in each window.

Classification is handled by the last few layers. After an input goes through every convolutional and pooling layer, it is classified by going through one or more fully connected layers. The data is converted from 3 dimensions to 1 dimension for classification. The final layer is also similar to a regular neural network as they are fully connected and have full connections to all the activations in the previous layer. While a CNN is also trained using backpropagation or gradient descent like a typical neural network, the convolution operations add a few degrees of complexity.

To summarize, CNNs have proven to be effective for image recognition tasks. The basics of CNN architecture have been described above. The remainder of this survey will be dedicated towards discussing new CNN connectivity patterns and new applications in research over the past few years.

A. *AlexNet*

The first neural network implemented on a GPU was successfully done by Dan Cladiu Ciresan and Jurgen Schidhuber with an NVIDIA GTX 280.

The biggest breakthrough in deep learning and image processing since LeNet-5 in 1998 was the 2012 Deep Neural Network AlexNet submitted for the ImageNet classification challenge. AlexNet was responsible for popularizing, since LeNet-5 from 1998, the field of deep learning with its results. Krizhevsky and his group was able to take the same insights that created LeNet-5 and explore them in a much deeper and wider neural network. The main motivation was to train a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into 1000 different classes [3]. They were able to achieve top-1 and top-5 error rates of 37.5% and 17%, which were significantly stronger than the previous state-of-the-art results. AlexNet contained about 60 million parameters and 650,000 neurons. The network consisted of 5 convolutional layers with a max pooling layer after the 1st, 2nd, 5th convolution, followed by 3 fully-connected layers with a final softmax of 1000. They also utilized some data augmentation methods such as translations, reflections, and patch extractions, to increase variability of the training images and compensate for overfitting. Krizhevsky et al. were also able to incorporate their dropout method to address the networks substantial overfitting issue. The group utilized batch stochastic gradient descent for to train the network. AlexNet was implemented on two Nvidia GTX 580s which took up to 6 days to train.

B. *VGGNet*

The Visual Geometry Group at oxford wanted to provide further insights towards themes explored from AlexNet. Simonyan and Zisserman were motivated to investigate the effects of network depth on the performance. They were able to successfully implement a convolutional neural network with depth up to 16-19 layers [1]. This deep network, with the use of small 3x3 filters, was able to push state-of-the-art-results. Their work was able to place first in localization and second in classification. VGGNet primarily used 3x3 convolutional filters as opposed to AlexNets 11x11. The authors explain that the motivation behind this is that a 5x5 receptive field can be simulated while still retaining the advantages of using smaller filters, such as reducing the number of parameters, by combining two layers of 3x3 convolutional filters together. A 7x7 receptive field can be simulated by combining 3 consecutive 3x3 convolutional filters. Another benefit of this architecture is that having multiple convolutional layers allows for multiple ReLU layers instead of just one. The group utilized scale jittering as a method for data augmentation and trained the network using batch gradient descent [1]. Their network was trained on 4 Nvidia Titan Black GPUs which took a total of 2-3 weeks.

VGGNet architecture was relatively simple but its biggest contribution was the insight it provided on network depth. To conclude, the authors discovered the notion that CNNs need to have sufficient depth to work effectively with the hierarchical representation of visual data.

C. ResNet

With the advent of VGGNet, researchers began to look towards increasing depth to achieve better performance. Driven by the notion of depth, they wanted to answer the question: Is learning better networks as easy as stacking more layers? The Microsoft Asia research team was able to break through the ceiling and successfully implement a neural network with 152 layers [4]. Their significantly deep network architecture was able to win and place new records in classification, detection, and localization in the ILSVRC 2015 competition. The network was able to achieve an error rate of 3.6% which is lower than the human error rate which is around 5-10% depending on the persons skill and expertise.

The main idea of ResNet is behind their residual learning framework which was motivated by an attempt to ease training of networks that are substantially deeper than those used previously [4]. The building block of this neural network is the notion of a residual block. An input has to pass through a series of layers which involve convolution, ReLU, followed by another convolution, resulting in F(x). This input is then added and combined with the original input to achieve H(x) = F(x) + x. The addition allowed for effective backpropagation as the gradient flowed with ease since the addition operates resulted in a distributed gradient. The authors realized that it is easier to optimize the residual mapping than to optimize the original, unreferenced mapping [4].

ResNets biggest contribution is that they were able to implement a neural network that outperformed humans. ResNet is the residual counterpart of classification neural network that forms by adding shortcut connections to the plain VGGNet networks. The identity shortcut follows two cases. If the input and output have congruent dimensions, they can directly jump to that layer. Otherwise, when the dimensions increase, there are two paths: Introduce no extra parameters with no extra zero entries for padding or match dimensions using 1x1 convolutions. For either of the two cases, shortcuts move across feature maps with a stride of 2 [4]. The group was able to implement a bottleneck architecture to address the considerable amount of time required to train the network. The residual function in each residual block is equipped with a series of 3 convolutional layers: 1x1, 3x3, 1x1. Each 1x1 is responsible for reducing then restoring the dimensions as the input moves in between the layers. The identity shortcuts were designed to be parameter free to consider time complexity and model size. [4] Combined with the bottleneck architecture, the group successfully implemented a deep residual learning framework that bested human error rates.

D. FCN

Long et al. took the classification networks like VGGNet, ResNet, or AlexNet and crafted them to exceed state-of-the-art results for semantic segmentation tasks. They successfully implemented a network, trained end-to-end, that was fully convolutional and could take an input of arbitrary size and produce correspondingly-sized output with efficient inference and learning [6]. By doing so, they adapted the classification convolutional neural networks into fully convolutional networks, creating a model for segmentation tasks.

E. DenseNet

DenseNet are relatively recent, but Huang et al. were able to offer an interesting perspective on neural networks. The group realized that the recent trend of networks with increasing layers highlighted the motivation for exploring different methods of connectivity. They also saw, with ResNet, that a key characteristic of these successful deep networks is shortcutting an input from an early layer to a later layer. Huang et al. designed a densely connected convolutional neural network which has condensed models, is easy to train and parameter efficient. Rather than exploring a deep or wide architecture, they combined feature maps through concatenation in contrast to summation by a process described as feature reuse [5]. This allowed for improved efficiency and increased the variation from input to subsequent layers. The network also consists of bottlenecking, inspired by ResNet, where the input is introduced with a 1x1 convolution before reaching a 3x3 convolution.

While the error rates of the DenseNet are on par with ResNets, the architecture was able to reach these results with significantly less parameters and computation. The groups contribution was inspiring the community to consider different architectures and connectivity patterns for convolutional neural networks.

III.

IV. Proposed Framework

Based on the previous breakthrough work in using deep learning for computer vision, this proposed framework utilizes a CNN to perform binary classification of satellite images. The framework's proposed model is shown below. It has two convolutional layers that are followed by max pooling. Each layer will run 128 filters of 3x3 size through the input for feature extraction.

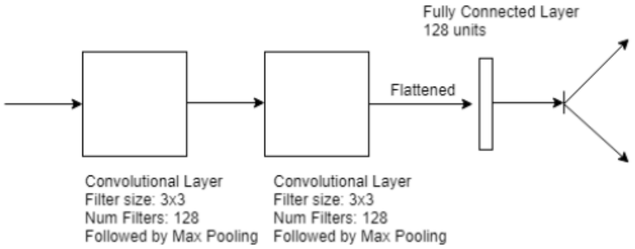

Fig. 1. Background subtraction flow

V. Data and Training

The training data is obtained by extracting already annotated images from the Gateway to Astronaut Photography of Earth. Because the model is binary classification, the initial dataset will have two classes of images: positive and negative. The positive sample are images that primarily contain a mountain, while the negative images do are images that do not contain mountains (i.e. cities). The resolutions for these images vary in resolutions anywhere from 492x515 to 4400x4600 pixels. Some of these are extremely high resolution images which provide a lot of details and information. To consider the lack of computational power, the data was resized to 256x256x3.

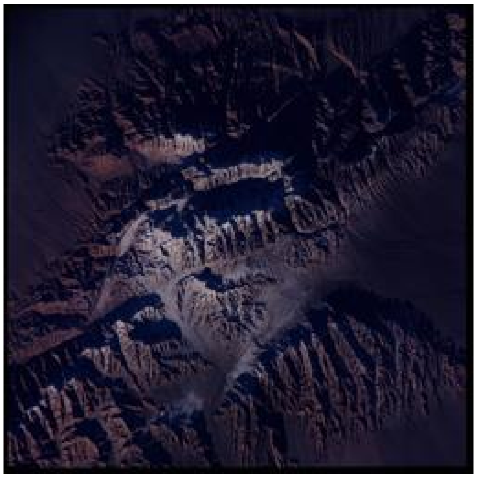

Fig. 2. Example of positive data (mountains)

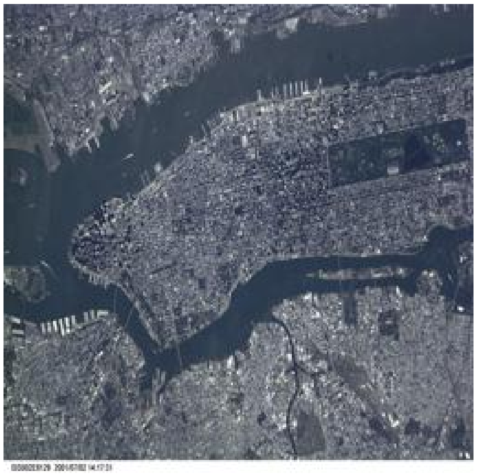

Fig. 3. Example of negative data (city)

Before performing large scale classification, the model is assessed in small scale classification. The initial training set consists of 500 positive and 500 negative samples. The training set was also used as the validation set. There were 100 postive and negative samples in the testing set. The optimizer used was stochastic gradient descent with a learning rate of 0.01. The best training performance is given below. Because the initial dataset was small scale, the batch sizes and steps per iteration were set at 32 each. The loss and accuracy began to degrade after 5 epochs. This is likely due to the relatively high learning rate.

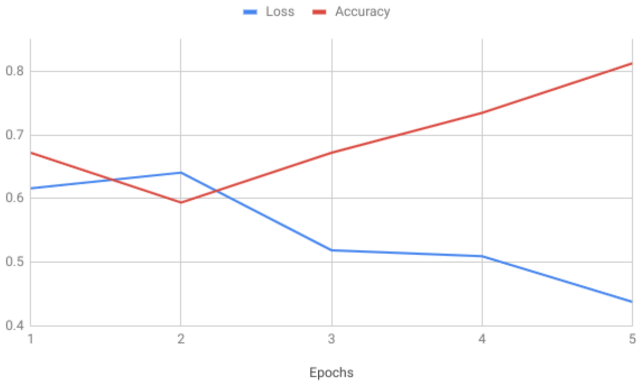

Fig. 4. Proposed model's training performance

VI. Experimental Results

The model was tested on the 100 positive and negative images. It was able to correctly classify 63/100 positive images and 87/100 negative images. The model was able to achieve an overall accuracy of 75%.

TABLE I
Performance on Test Set

Label	**Accuracy**
Positive	63%
Negative	87%

VII. Future Work

The next step for this project is to try satellite image semantic segmentation. We want to do instance based segmentation to find each and every feature in the image. This was our initial plan but we need sufficient hardware and computation to achieve this. Obtaining the dataset for training is also a challenge as hand labeling pixel-wise segmentation is too labor intensive.

VIII. conclusion

We successfully trained a model to classify whether or not an image contains mountains. We deployed this model to a web app and created a dashboard for users to upload satellite image and return classification results. Our results show us that it is viable to train a model to help with the large scale annotation process of astrophotography for the Gateway to Astronaut Photography of Earth.

References

[1] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556. 2014.

[2] Hubel, David H., and Torsten N. Wiesel. Brain and Visual Perception: the Story of a 25-Year Collaboration. Oxford University Press, 2005.

[3] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” In Advances in neural information processing systems, pp. 1097-1105. 2012.

[4] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ”Deep residual learning for image recognition.” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778. 2016.

[5] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. ”Densely Connected Convolutional Networks.” In IEEE conference on computer vision and pattern recognition, vol. 1, no. 2, p. 3. 2017.

[6] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. ”Fully convolutional networks for semantic segmentation.” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431-3440. 2015.

[7] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. ”U-net: Convolutional networks for biomedical image segmentation.” In International Conference on medical image computing and computer-assisted intervention, pp. 234-241. Springer, Cham, 2015.

[8] Wang, Qi, Junyu Gao, and Yuan Yuan. ”A joint convolutional neural networks and context transfer for street scenes labeling.” IEEE Transactions on Intelligent Transportation Systems, 2017.

[9] Dickerson, Kelly. NASA Wants You to Help Sort Astronaut Photos of Earth at Night. Space.com, Future US, Inc., 5 Sept. 2014, www.space.com/27023-nasa-astronaut-photos-earth-at-night.html.