Binary Image Classification: Manmade and Natural

A Study with CALTECH 101 Dataset

Sandra Sajeev

Carnegie Mellon University

Submitted 5 May 2018

Abstract

This paper presents a study on binary image classification with semi-complex categories. Though the classification of natural versus manmade seems simple at first, it is further complicated by backgrounds and technicalities of what makes an object considered natural or man made. The techniques used to extract the features are based on classical image classification techniques. At the same time, present day techniques such as modified Bag of Words and pooling are utilized. The resulting model helps highlight which features are the best for improving classification performance. This insights from this paper could be applied for the first layer of a convolutional neural network.

Introduction

Image classification has become a hot topic with recent advances in computer vision. The use of convolutional neural networks has greatly increased the accuracy of many image learning models. For this project, classical approaches to machine learning and images were explored, while still employing some aspects of recent work. The task of classifying man made versus natural objects can be a difficult binary tasks even for humans. For instance, if a person where given a painting of a butterfly, one could argue that the butterfly is the object and that is a natural objects. On the other hand a painting is created by a human, and this can be classified as natural. It broadly depends on the definition of what is manmade and what is natural. For the purposes of this project, natural and man made objects are determined by the object in the foreground and what is represents. For instance, if there was a paper crane, it would be identified as natural since cranes as a concept are from the natural world. These types of complex classification problems show up many times in computer vision, where it can be difficult for even a human to identify a particular object. On a larger scale, object classification is highly important in applications such as autonomous driving, facial recognition, image search, etc.

Most of the recent research has separated from conventional methods of image processing. This paper deals with a problem that is particular complex for a binary classification of images that are either manmade or natural. There is another layer of complexity, because some of the natural objects are captured in a manmade

Related Work

Bag of Words came up as a popular method of image classification before the introduction of ConvNets. The paper, “Understanding bag-of-words model, a statistical framework”, helped describe the algorithm for bag of words when applied to images. As described by the paper, Bag of Words involves 4 main steps. The first step involves the creation of a dictionary of words by applying filters and feature descriptor to an image. A feature descriptor will get the key points in an image using an algorithm. One popular one is the SIFT feature descriptor. Next, you will apply different filters to the images and start to build a dictionary with all the images in your test set who have undergone filtering and keypoint extraction . Then each image in the test set can be represented as a histogram. This project didn’t follow the bag of words procedure exactly, but it did incorporate aspects of it. For instance, k-means clustering was used for the colors in each image. In addition, a dictionary of features for each image was created. A histogram of words was not calculated, because, it can be computationally expensive.

The paper, “Beyond Bag of Words, Spatial Words” also made use of the Caltech 101 image dataset. They used multiple word histograms to identify different natural scene categories from Caltech 101. They had a slightly similar objective of identification of certain natural scene categories. This paper went one step further and had more than just a binary classification. They randomly picked areas of the image to get visual words from and built multiple visual histograms for an image (Lazebnik, Schmid, & Ponce). The performance values for recognition of 15 categories was 64.6 for strong features (Lazebnik, Schmid, & Ponce).

Data Setup

The data that was used for the experiment was from CalTech 101, an image dataset of 8000+ images grouped in 101 categories. For the purposes of this project, the binary classification of manmade or natural was added on all of the labeled images. There was a 50-50 distribution of images that were manmade and natural. The data was split into development, test, train, and final test. The breakdowns of each category are below. Due to the size of dataset, a distinct test and train class was created rather than relying on cross validation.

Table 1: Dataset Breakdown

Dataset	Number of Instances
Development	532
Train	3729
Test	3196
Final Test Set	1066
Total	8523

The class value that I decided to focus on was the designation manmade versus natural. This category was manually created based on the labeled images from CalTech 101. An arff file and a csv was generated for for each of the sets, train, test, development, and final test and put a random assortment of images in each, while still ensuring that the 50-50 distribution of manmade to natural existed in all of the sets. A series of features were extracted from all of the images using various image processing techniques described below.

Table 2: Feature Extraction

Feature	Description	Instances
K-means Colors	Clusters the colors of the image into k numbers using clustering, separated by rgb channel	12
K-means foreground colors	Clusters of colors extracted from foreground of the image using k means, separated by rgb channel	9
Key points	Number of features derived from SIFT algorithm which identifies meaningful points in an image	1
K-means Edge Filtered Dictionary	Made a dictionary for each word by first computing SIFT features, using 10 of the SIFT features to calculate 10 key points. Then I gather the filter responses at these points for the Laplacian, Vertical Sobel, Horizontal Sobel, Gaussian	5
Fourier Transform	Applied a fast fourier transform to the image with a convolution, then pooled the results into a 1x36 array	36
Gabor Filter	Applied a Gabor filter to the image and then pooled the results into a 1x36 array	36
Fourier Transform Foreground	Applied a fast fourier transform to the foreground image with a convolution, then pooled the results into a 1x36 array	36
	Total:	182

Experimentation

The baseline experiment consisted of having just the number of key points from the SIFT feature detector. This provided really suboptimal performance. The model was trained with SVM with the train set and was tested on the test set. With this model, an accuracy of 65.33% and a Kappa value of 29.67% was achieved. The Kappa value reveals that random chance has a large role in the accuracy value achieved. The confusion matrix for this experiment is below.

Table 3: Baseline Results

Actual/Predict	Manmade	Natural
Manmade	1316	359
Natural	749	772

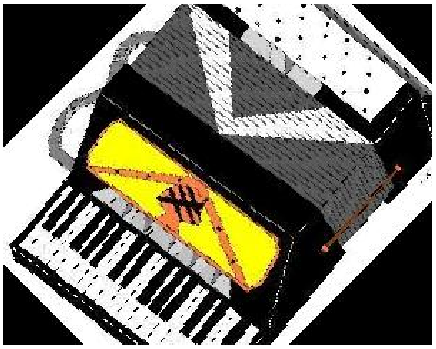
One interesting error analysis occurred when Fast Fourier Transform features were added the to the feature space. This helped increase the Kappa performance by almost 6%. The feature space before this round of error analysis consisted of 4-k-means clustered colors, 3 k-means foreground clustering and 50 k-means clustering of a combination of sobel filters, laplacian, gaussian blur, and the bilateral filter. In addition, the number of SIFT features each image has was also included. LightSide was used to extract column features.

At first, a model was trained on the train set. The support vector machines was the learning algorithm, because it is frequently used in the image classification tasks. For this baseline test, a Kappa value of around 44.91% and an overall accuracy was 72.59% were achieved. In addition, 180 more manmade images were misclassified as natural. Then for error analysis, a model was trained on the train set, and tested on the dev set.

Similar values for the Accuracy/Kappa values were achieved as compared to the original baseline model. Once again there were more manmade images that were misclassified. Some of the misclassified instances are described below.

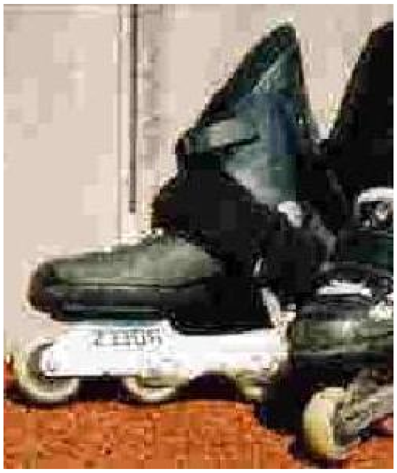
Many of the natural images that were misclassified were mostly drawings. The model seems to be having a hard time determining whether a drawing of an animal is actually representing a natural object. Most of the manmade images that were misclassified were real images, but the noise of other things in the foreground is messing with the results. In addition some images were rotated.

Figure 1: Manmade Classified as Natural



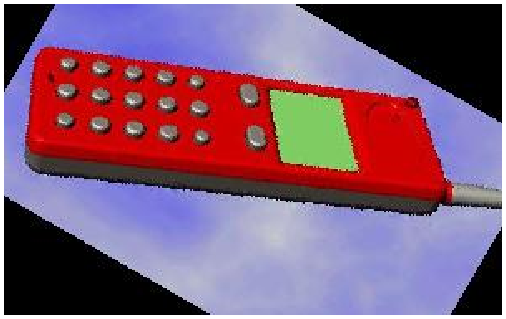
This image may have been misclassified in part due to the fact that it is oriented in a weird direction

Figure 2: Manmade Classified as Natural



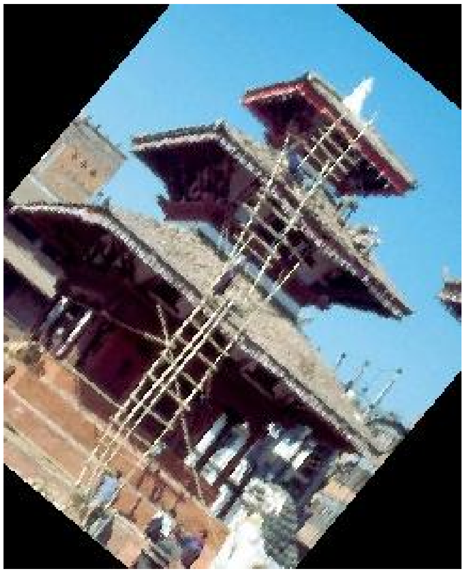
It is not clear whether this image is an illustration or if it is real. Therefore, FFT, may provide insights about the type of image as well as helping identify that this is a manmade image

Figure 3: Manmade Classified as Natural



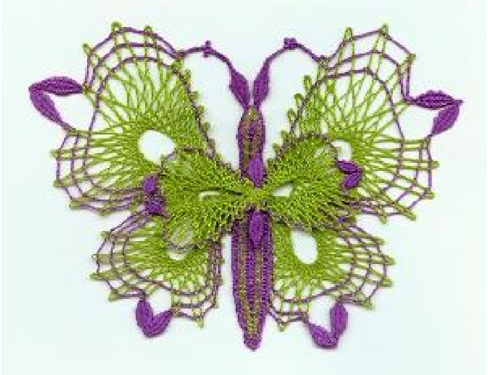
This image has colors that could be interpreted as natural. In addition, it’s orientation may be confounding the model as well. The soft gradient of colors in the background may be confounding the model as well.

Figure 4: Manmade Classified as Natural



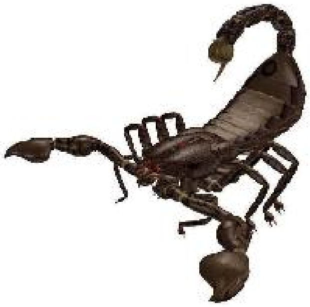
This image was most likely misclassified partly due to its orientation and the blue sky. The orientation could be confounding the clustered edge information.

Figure 5: Natural Classified as Manmade



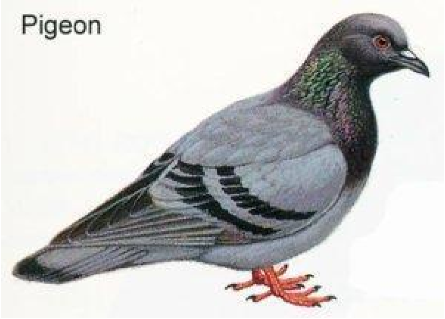
This image is of a butterfly, but it is an abstract representation, with colors that typically do not show up in nature. In addition the texture of the image is off.

Figure 6: Natural Classified as Manmade



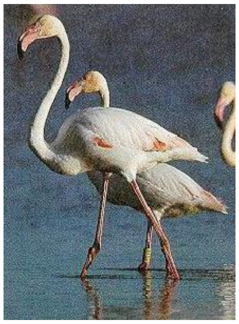
This image was misclassified because it looks like an illustration of a scorpion. The image is also slightly blurred, possibly confounding the model even more.

Figure 7: Natural Classified as Manmade



This is a realistic illustration of a pigeon. There is a vibrance in this image due to it being created by humans. The colors are not exactly like the ones found in nature.

Figure 8: Natural Classified as Manmade



This is a stylized filtered image of flamings. It could have been real if it was more detailed. There are some semi sharp color transitions in comparison to a more natural image.

To fix this problem, he Fourier Transform (FFT) of each image and the DC offset was computed. The DC offset of the Fourier transform is an average of all the signals in the image. Images that are illustrations will generally have a higher FFT values since there is more sharpness in the color changes. The FFT will also allow the model to identify the directionality of the image, since some of the misclassified manmade images were rotated.

The numpy FFT function was used to compute to Fourier Transform. Afterwords, the resulting image was shifted so that the DC offset of the FFT was centered in the middle. Thena logarithmic function was applied to the image to further see the impact of different frequencies. The results were pooled and then reshaped into a 1x36 vector to put 36 new features into the model.

After, this the new feature space was tested by building a model with SVM (Support Vector Machines). This model had an accuracy of 75.22% and a Kappa of 50.25%

Table 4: Confusion Matrix for FFT Model

Act/Predict	Manmade	Natural
Manmade	1302	373
Natural	419	1102

A performance increase of 6% for Kappa and 3% for overall accuracy was acquired. A t-test was conducted to see if there were significant results based on the current model and the one prior to the addition of the FFT features. A p value of approximately 0 was computed. Therefore, there was a significant improvement on the model.

This was an intriguing error analysis because it combined aspects of signal processing to the problem at hand. The fact that mapping the location and intensity of the fourier transform magnitude could reveal insightful information about the image for the model was surprising.

Tuning

As the feature space was built, the model was primarily tested on one machine learning algorithm, the Support Vector Machine. This algorithm was used, because it has proven effective in many machine learning tasks, especially those related to images. The default SVM algorithm with Normalization and LibLinear on LightSide was used. With tuning, I was trying determine if the model could be improved by changing the parameters. The chart below shows the parameters that have been tuned for SMO. The complexity parameter was tuned to see whether a hyperplane with a larger distance constraints would would allow different patterns emerge. The the exponent in a Normalized Poly Kernel was also, to determine if there was an aspect of the data that had nonlinearity in it.

Table 5: Tuning Parameters Tested

Algorithm
SVM: Normalize Class Values, LibLinear (Default)
SMO Complexity Param (c = 1.0)
SMO Complexity Param (c = 3.0)
SMO Complexity Param (c = 10.0)
SMO Normalized Poly Kernel Exponent (exp = 0.0)
SMO Normalized Poly Kernel Exponent (exp = 2.0)
SMO Normalized Poly Kernel Exponent (exp = 3.0)

After doing tuning, the best results were achieved using SMO with a complexity of 3.0. To see if the tuning was significant compared to the default tuning parameters for the algorithm, I used a paired-t test, and got a value of 0.2952 which is greater than 0.05. Thus tuning will not give improved performance.

Final Evaluation

In order to conduct the final test, the train and test sets were combined. This resulted in a final training set of 6925 images. Then SVM with LibLinear was applied to train the model. With the final evaluation on the final test set, which consisted of 1066 images, an accuracy of 73.39% and a Kappa value of 48.52% was achieved. A total of 281 images were misclassified. This is inline with the results acquired during training of the model.

Table 6: Final Confusion Matrix

Actual/Predict	Manmade	Natural
Manmade	437	122
Natural	151	365

Future Work

This project was a great introduction into the challenges of machine learning principles and the methodology to follow for classification tasks. Machine learning is both an art and a science. There is a level of creativity needed when trying to improve your model from the features you choose to the algorithm that is applied for learning task. The next step in enhancing this model would be implementing a convolutional neural network. Within the scope of this project, I was able to identify a good set of filters and techniques that would by applicable to the first and second layers of convolution. The design of this neural network can have a significant positive impact on the overall accuracy of the model. For future work, it would be interesting to pursue classification and testing of the model with images from other image datasets or integrate into an app to test it with images one may take on a cell phone. This way, we can identify if there is any inherent bias towards the Caltech 101 image dataset. This idea of biased datasets is a big problem in computer vision now where models perform really well on the dataset they are trained on, but not at well for everyday scenarios. Regardless, the project helped revealed the complexity of image classification problems. In addition, it helped develop a methodology to do machine learning.

References

- Lazebnik, S., Schmid, C., & Ponce, J. (n.d.). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR06). doi:10.1109/cvpr.2006.68
- L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004
- OpenCV-Python Tutorials. (n.d.). Retrieved from https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html
- Zhang, Y., Jin, R., & Zhou, Z. (2010). Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics, 1(1-4), 43-52. doi:10.1007/s13042-010-0001-0