Going Deeper: Depth Image Classification via simulated SPAD array images

Abhi Jadhav
Department of BME
Duke University
Durham, NC 27708
aj246@duke.edu

Dhanasekar Sundararaman
Department of ECE
Duke University
Durham, NC 27708
ds448@duke.edu

Jason Liu
Department of BME
Duke University
Durham, NC 27708
jl532@duke.edu

Abstract

Single-Photon Avalanche Diodes (SPAD) are affordable photodetectors, capable of collecting fast low-energy events, due to their single photon sensitivity. This makes them very suitable for depth based imaging systems, while maintaining high temporal resolution [2]. In this work, we aim to simulate SPAD-based imaging by using existing RGBD datasets as the ground truth and a modified VGG architecture for classification. We used 5556 images from 21 classes of household objects from the RGB-D Kinect Object Database from the University of Washington CS Department [1]. In the best case scenario, we get great performance in classification of up to 91.26% on the training data and 91.19% on the test data. When implementing non-ideal physical conditions with the introduction of random noise, the training accuracy dropped by 1% while the test accuracy dropped by 4%.

1 Introduction

1.1 General 3D depth imaging information

Three dimensional (3D) depth imaging is a rapidly growing field of research. Interest in color-depth (RGB-D) camera sensors skyrocketed after the November 2010 release of the comparably low-cost, consumer-grade Microsoft Kinect RGB-D system [2]. Modern 3D Imaging technologies can be categorized by three classes: 1) Stereo Vision, 2) Structured-Light, and 3) Time-Of-Flight (TOF). Stereo vision exploits the same parallax effect that our two human eyes affords us. Structured-Light projects a known pattern on the scanned field of view, a camera observes the distorted projected pattern, and the distortions are processed for depth. Finally, time-of-flight imaging uses a modulated light source with a known pulse or continuous wave and a fast photo-sensor that can measure the phase shift of the reflected light [3]. Of these three options, TOF has significant potential for widespread use as a 3D depth imaging modality due to its low cost, fast response time, medium depth accuracy, scalability, and low software complexity. This potential is reflected by a major increase in research activity in the field, increasing number of publications year to year, growth of RGB-D datasets available, and development of higher performance TOF sensors [4]. Given the availability of RGB-D datasets, the development of TOF sensors, and our training from BME590: Machining Learning in Imaging at Duke University by Professor Horstmeyer, in this paper we evaluate a machine learning

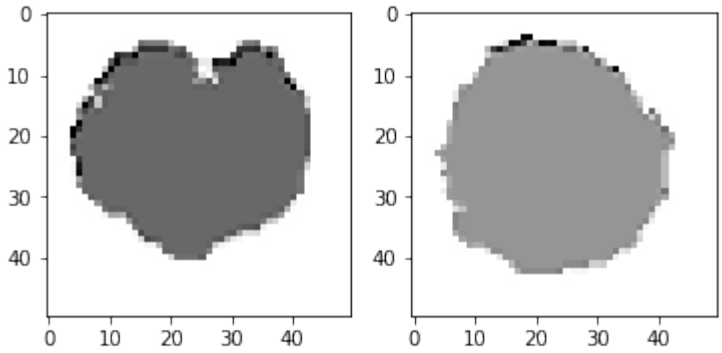

Figure 1: Depth image of an apple(left) and a lemon(right)

approach to determining the minimum viable physical parameters needed within a TOF sensor system to classify objects.

1.2 Single-Photon Avalanche Diodes as a TOF sensor for depth imaging

We anticipated the use of Single-Photon Avalanche Diodes (SPAD) as the specific TOF sensor. Their affordability, single-photon sensitivity, and speed are suitable for TOF sensing, and have been seen in recent research for TOF sensing. There are limitations to their use that we explore in this paper. Specifically, SPAD arrays are currently limited by their small sensor pixel dimensions, noise characteristics, and ranging issues. The images used in our dataset are not generated by a SPAD array. Rather, they were generated by a Kinect Structured Light 3D imaging system.

2 Methods

2.1 Harvesting RGB-D Datasets and preparing for machine learning

Though there are many online lists of RGB-D datasets, many are no longer hosted, poorly annotated, poorly formatted, or contained extraneous depth images not relevant to our specific application [5]. This is interesting, however– seeing the various applications of depth imaging by the many types of data out there. There are generally categorized into several classes of image data for several applications of depth imaging: scenes, videos, poses, meshes, segmentation, maps, semantics, multiclass images, bounding boxes, and more. From the sea of these incompatible datasets, we found the RGB-D Kinect Object Database from the University of Washington, which contains 51 categories of 300 objects. The dataset was generated by rotating each of these 300 objects, recording RGB-D data from a kinect device, recording every 5th frame, and manually cropping the images to only include the image area. We were able to download the evaluation dataset, and using a Python script to go through all of the directories of the data, we collated 5556 images from 21 classes.

2.2 Machine Learning Approach

2.2.1 Description of the data

The data consists of 21 classes namely 'apple', 'ball', 'banana', 'bowl', 'calculator', 'cell', 'flashlight', 'food', 'garlic', 'instant', 'kleenex', 'lemon' ,'lime', 'orange', 'plate', 'pliers', 'potato', 'scissors', 'shampoo', 'tomato', and 'water'. The depth images of all these objects totally number to 5556. Two sample depth images representing an apple and a lemon respectively is shown in Figure 1. The classes are imbalanced and the data is split into train and test in a 90:10 ratio.

2.2.2 Preprocessing

The raw RGB-D images obtained were inconsistent in resolution, shape, and depth resolution. The smallest image was 50 x 50 while the largest one was almost 200 X 200. These inconsistencies

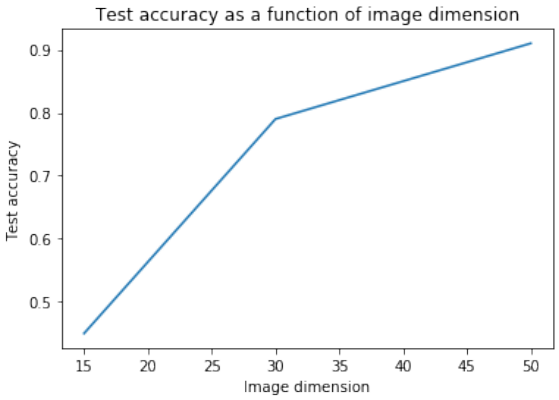

Figure 2: Objects classification Test accuracy as a function of image dimensions

in the data are not suitable for running classifier models, so we had to scrub and format the images. All the images were converted to a uniform resolution of 50 x 50 using resizing and padding. 50 x 50 pixel dimensions were chosen is because it proved to be the lowest dimension possible without comprising much on classifier accuracy. Figure 2 illustrates that images with much lower dimensions such as 15 x 15 and 30 x 30 fail to provide enough information to the CNN classifier model to achieve good accuracy.

2.2.3 Analysis using CNN models:

Two CNN models were initially chosen, wherein we describe them as configuration A and B. Configuration A only has dense layers connected with each other and a softmax function in the last layer. Configuration B nearly follows a VGGnet architecture with a 5x5 2D convolution layer followed by max pooling repeated a few times. There are 2-3 fully connected layers followed by a softmax function in the last layer.

First, a two-class classification problem was set up between images of apples and oranges. Both the configurations were tested on this problem. Config A failed to get more than 60 percent accuracy on this problem as it lacked convolutions and pooling while Config B almost got near 100 percent accuracy. Then the number of classes was increased as Config B was able to achieve near perfect classification accuracy. The number of classes was increased to 5 and then later to 21. The results, consisting of train and test accuracy, as well as other evaluation metrics such as confusion matrix, is discussed in detail in the results section.

2.3 Modeling Imperfections of SPAD

Despite all the benefits of the SPAD array, there are limitations that we needed to model. For example, noise in SPAD-generated images limits the resolution and quality of depth images. This noise is due to a variety of things like errant light, heat pixels, dead pixels, or more, leading to false depth pixels (voxels), and thus potentially poor classification. To simulate noise, we randomly modifed pixel values of input images in the train as well as test data and observed the CNN model's classification performance. As expected, training accuracy dropped by 1% and test accuracy dropped by 4% when random noise is introduced on 1000 pixel values in the train data and on 100 values in the test data (0.008% noise).

3 Results

The results section shows the accuracies obtained with the two different configurations of CNN mentioned in section 2.2. The table shows the accuracies of the two models of different classification problems. Model B seems to outperform A as it has many more of convolutions and pooling operations within the network.

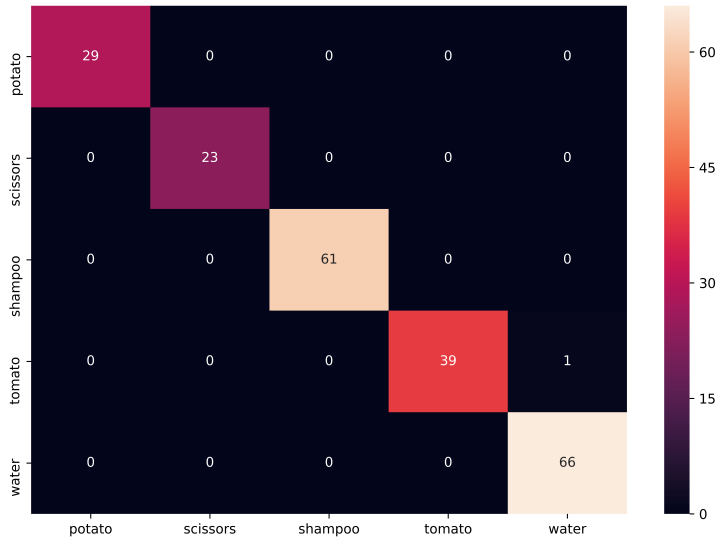

Figure 3: Confusion Matrix for the 5-class classification problem

Figure 3 and 4 shows the confusion matrix of the model B when applied on the 5-class and 21-class problem respectively.

Model	Problem	Train accuracy	Test accuracy
A	2-class classification (Apples vs Oranges)	54%	53%
B	2-class classification (Apples vs Oranges)	100%	99.9%
B	5-class classification	99.92%	99.54%
B	21-class classification	91.26%	91.19%

4 Discussion

4.1 Limitations to the data used

As a whole, our study into depth image processing is, like many other studies involving machine learning, limited by the data. We originally set out to fully simulate SPAD array technology using many high quality depth images. However, such data proved to be very difficult to find, and the dataset we were able to find and engage had very poor depth resolution (all images had 3-4 layers of depth to each image). were very obviously poor in quality (edges were ill-defined, and certain portions of the image were not resolved properly), and also low in quantity (only 5556 images). Given these limitations, we were still able to demonstrate basic classification of 21 objects by the depth channel alone with >90% accuracy, which shows promise for the future of depth imaging and classification with just depth sensors alone. The code we put forth here serves as a strong framework to build from, once such higher quality images exist.

4.2 Proposed future work

We were able to implement a framework simulating the core physical properties needed in a TOF sensor: sensor pixel resolution and sensor noise characteristics. That being said, we acknowledge a major limitation to our study: the inability to fully simulate all physical parameters of a SPAD array. To address this, we have laid out what is left to be done in this project: The remaining elements needed to fully simulate a SPAD array would be to model either time-gating or integration methods

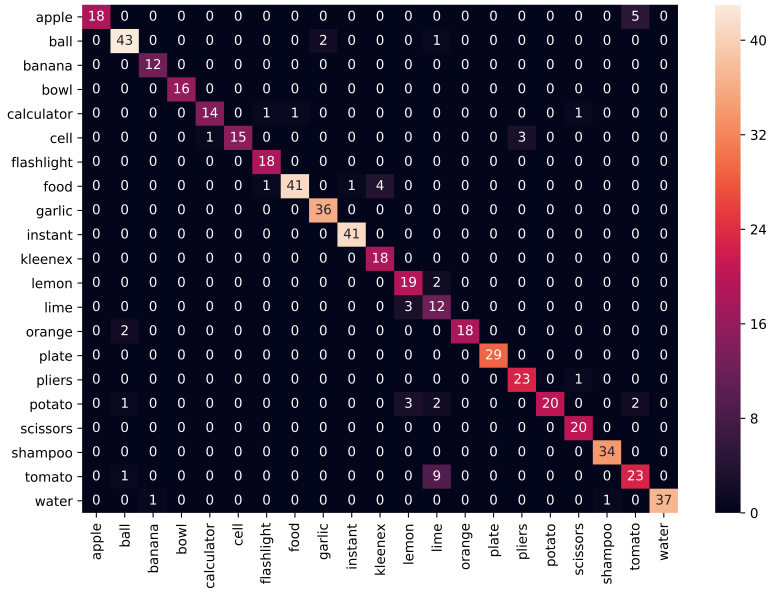

Figure 4: Confusion Matrix for the 21-class classification problem

for 3D ranging, which would have involved "slicing" high quality depth images into channels to feed into the network. More specifically on the physical parameters, we would need to simulate excitation pulses or continuous waves with near infrared light, observe the reflected and transient components of the light, and use the traditional TOF ranging equations as listed in a recent 2017 paper [7]. After this, however, our existing code can add in random noise and downsample to test the minimally viable physical characteristics needed within a SPAD array.

5 Conclusion

As a whole, we were able to demonstrate basic image classification from depth images from an existing dataset, finding minimum pixel dimensions and relative noise characteristics within a depth imaging sensor. We were limited by the quality of publicly available depth imaging datasets, but were able to work with what we found to achieve >90% classification accuracy. We were able to simulate lower resolution sensors and noise, which our system was very sensitive to. We have learned a lot about depth imaging, the growth of the field, and machine learning approaches to designing imaging systems.

References

[1] https://rgbd-dataset.cs.washington.edu/

[2] K. Litomisky, “Consumer RGB-D Cameras and their Applications," University of California, Riverside. (2012).

[3] L. Li, “Time-of-Flight Camera - An Introduction," Texas Instruments. Sensing Solutions. (2014).

[4] A. Kolb, E. Barth, R. Koch, R. Larsen, “Time-Of-Flight Cameras in Computer Graphics," Computer Graphics Forum. (2010)

[5] http://www.michaelfirman.co.uk/RGBDdatasets/

[6] Y. LeCun, Y. Bengio and G. Hinton, “Deep Learning," Nature **521**, 436–444 (2015).

[7] Y. He, H. Liang, Y. Zou, J. He, J. Yang. “Depth Errors Analysis and Correction for Time-of-Flight (TOF) Cameras. Sensors. (2017)