Machine Learning (Fall 2017) Project

On my honor as a UNT student, I have neither given nor received unauthorized assistance on this work.

Note regarding unauthorized assistance: I would prefer that you learn as much about this subject as possible through your project. Therefore, I encourage you to talk to everyone and anyone who is willing to discuss (or even just hear about) your project. Feel free to involve collaborators somewhat deeply, but in order to learn, ***you will of course need to do a significant amount of work yourself****. I expect all written assignments to be your own work and reflect* ***your*** *understanding of that work. That said, your paper should reflect feedback from your class peer group as well as from me. Please include proper citations for any writing or other intellectual property that came from someone else. Missing citations will be reported to the office of academic integrity and will have a substantial negative impact on your grade.*

Write this paper in the style you would write it for a relevant conference or journal (even if you don't plan to submit it). If you do plan to submit the paper, follow the conference or journal guidelines. Otherwise, **the paper should be approximately 2500 words** (*not including the introduction and prior work section*) – an absolute minimum of 1500 and maximum of 7000 words (not including the sections starting at *Acknowledgements*). All projects are unique – you do not have to include everything in these guidelines, but a good paper will discuss most of these issues.

Domino detection and classification of Domino image Dataset

Isaac Samuel Raj Boddu, Karthik Venkataramana Pemmaraju

Date: 12/7/2017

Author List, with affiliations and email addresses
(Please include everyone you feel has made an *author-worthy* contribution to your project.)

Abstract

A lot of vision based applications like FaceID, gesture control and object detection in real time systems, face challenges in performing to a satisfactory and reliable extent.

Because of such performance issues the advancements in the field of computer vision and similar applications have been suffering.

A lot of recent research work in related areas have demonstrated promising and exciting results lately. Together with Machine Learning techniques, computer vision applications have started to show some spectacular results.

By using deep vision techniques, we hope to identify dominoes in each given image and possibly identify the position and its significance in a given setting, which in turn could be used to identify the state of a game and/or help decide the next move for a user or robot to make progress in the game as a later possible extension to this project.

By using machine learning we can potentially develop models which would make playing dominoes with inanimate players or a machine more enjoyable while not explicitly specifying rules and moves for that game. This will help in improving companionship in adult subjects with declining cognitive abilities or for those in need of companionship.

By developing more generalized algorithms or models to create spatial awareness without the need of additional/specialized hardware could improve the nature of services provided and to reach a wider range of recipients. This could also help save money by reducing the cost of currently expensive alternative solutions for entertainment and medical purpose robots.

Introduction

The project aims at efficiently identifying select objects(Dominoes) on a given background and identify its significance in each game state. We hope to extract enough information to help determine a possible next move, by reading the value based on the dominos position.

Upon successful implementation, we hope to be able to generalize the model/algorithm enough so that it would deem useful in other games or applications.

Gaming and medical groups would be interested in such research applications.

Having such a model could be a huge advantage in the gaming community to develop human-machine interactive games. Likewise, it could mean advancements in the field of robotics which require specialized spatial awareness.

Although there are a lot of new deep learning implementations which perform well for a lot of regular applications, there are no models for specific applications like detecting dominoes.

There is a very high level attempt at detecting the face value of dominoes in a very controlled and clear environment.
https://www.youtube.com/watch?v=1OIDYBZeGgw

There are various state of the art results out there which perform well and there is a constant rush in improvements with the advent of ML techniques. The biggest drawback for previous implementations is that they performed well in controlled environment but would not be able to do well in noisy or real world setting.

The key contributions of this work are:

- *Detect dominoes in noisy conditions*
- *Located dominoes in the image and locate their position*
- *Identify the position/condition of dominoes in an image (Standing/Laying down)*

Background/Preliminaries, Definitions, Data, Examples, Key issues/problems, Problem framework

Dataset used for the project (HiLT Domino Dataset) contains folders GreenTrain and GreenTest which hold the original images included in the domino dataset. These images are of green dominoes on a variety of surfaces, in a variety of lighting conditions, and arranged in a variety of configurations. Annotations were completed by students in the Spring 2017 section of UNT's CSCE 4310/5210 Artificial Intelligence course. New annotations were created as the ones we used initially were not very helpful in training with our model framework.

The images were captured using a NAO robot's bottom camera.

The data set used contains images covering a wide range of positions, orientations and physical state of the dominoes. Also, the dataset was specifically created for the purpose of such machine learning purposes and it has been ensured that dataset is independent and identically distributed. Because of the size of the dataset used, enough care was taken to ensure even distribution of samples for training, validation and test datasets.

Data Set	**Total**	**class A**	**class B**	**class C**	**class D**	**class E**	**class F**	**% of Dominoes**
Training Set	981	206	316	119	110	180	50	64.15958143
Validation Set	103	38	13	8	14	26	4	6.736429039
Test Set	445	82	132	60	49	88	34	29.10398954
Total	1529	326	461	187	461	187	88	99.9999

Data Set	**Total Dominoes**	**class 1**	**class 2**	**class 3**	**class 4**	**class 5**	**class 6**	**class 7**	**class 8**	**class 9**	**class 10**	**class 11**	**class 12**	**class 13**
Training Set	486	32	43	32	44	36	80	50	42	31	34	23	15	24
Test Set	226	11	21	25	25	22	30	18	25	13	11	8	8	8
Validation Set	38	4	5	3	4	3	6	3	3	1	2	0	1	3
Total	750	47	69	60	73	61	116	71	70	45	47	31	24	35

Either here or in the intro, consider including a framework/model for solving the problem, including a diagram depicting the framework. Describe the key aspects of most solutions, cast in a way that enables you to refer back to them throughout the paper, (e.g., so that in the later sections of the paper, you can easily compare and contrast your solution to the state of the art).

Approach / Methods

Our hypothesis was to identify dominoes in each given image. We attempted to capture the position and location of dominoes in an image and to ensure it was generalized well enough to be used by other applications. Because the challenge was clear due to the noisy settings and varying light conditions, it was important to train the model in varying conditions and with different modes. To achieve a decently well performing model, a lot of effort was spent on tuning hyper-parameters and on training on limited hardware.

All the available data was divided into three datasets – Training, Validation and Test.

All the experiments conducted used training dataset for the training phase and the models were trained to fit the training data. After at set checkpoints the performance was tested on a different set of validation set to tune for performance on that set. After many iterations, once we were confident about the model, we used data from test dataset and evaluated the model. No further modifications were applied to the model afterwards and the performance is captured.

The goal was to be able to identify, locate and explain the position of multiple dominoes in a given image. Because of the nature of complexity of the task at hand, a lot of attempts to create a single model to deliver on all the sub-tasks failed in spite of tweaking with model and data distributions. After a lot of planning and design meetings, we had to move to a relatively different approach but the one which gave the most acceptable approach. We had to train two models, one to identify, locate, and explain the position of the domino and another model which would identify, locate and provide the face value of the domino. This dual model approach worked best out of all the approaches attempted.

We also ensured that the data used to teach the model was describing domino features well enough to be able to perform the tasks we set out to do, while at the same time ensured that it generalized well enough to be able to perform in various external conditions to be usable for other applications and purposes. This is particularly important because the previous attempts were in extremely controlled environments and hence not suitable to be used in real world applications.

Regions with Convolutional Neural Networks were used to accomplish the task. We also tried to use other variations of CNN algorithms, but these were better for our task than others during our experiments.

Model name	Speed	COCO mAP	Outputs
ssd_mobilenet_v1_coco	fast	21	Boxes
ssd_inception_v2_coco	fast	24	Boxes
rfcn_resnet101_coco	medium	30	Boxes
faster_rcnn_resnet101_coco	medium	32	Boxes
faster_rcnn_inception_resnet_v2_atrous_coco	slow	37	Boxes

Tensorflow Detection Models

The algorithm clearly learned the edges to identify dominoes although it is not visually obvious, the spatial location of the dominos, isolate each domino, identify the physical state of the domino and attempt to read the face value on each domino detected.

The models were fine tuned to perform for the domino detection task. Tuning the models includes changing the hyperparameters like learning rate, momentum, layers used, nodes in each layer, number of classes, sub-image dimensions, kernel stride and modifying distributions of datasets.

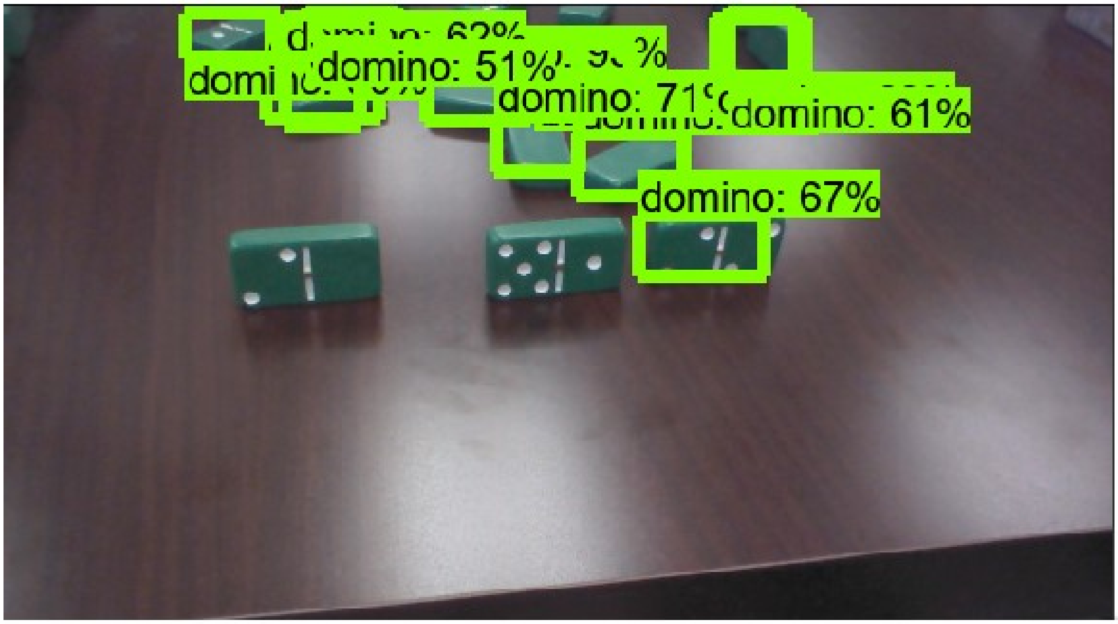
Also include a link to your code (or submit the code separately).

Experiments

Design / Methods

The objective was set right at the beginning and a crude design for a model was drawn at the beginning. There were issues with the setup of environment and configuration issues which persisted for a while. Eventually we got to the point where we got the whole setup ready, but the hardware on our machines did not support the intense computational demands and we sought out to online providers like Google GCP, Paperspace and even our very our Talon. The burden of setting up on different environments discouraged us to try on multiple machines. We got relatively better hardware and started conducting experiments.

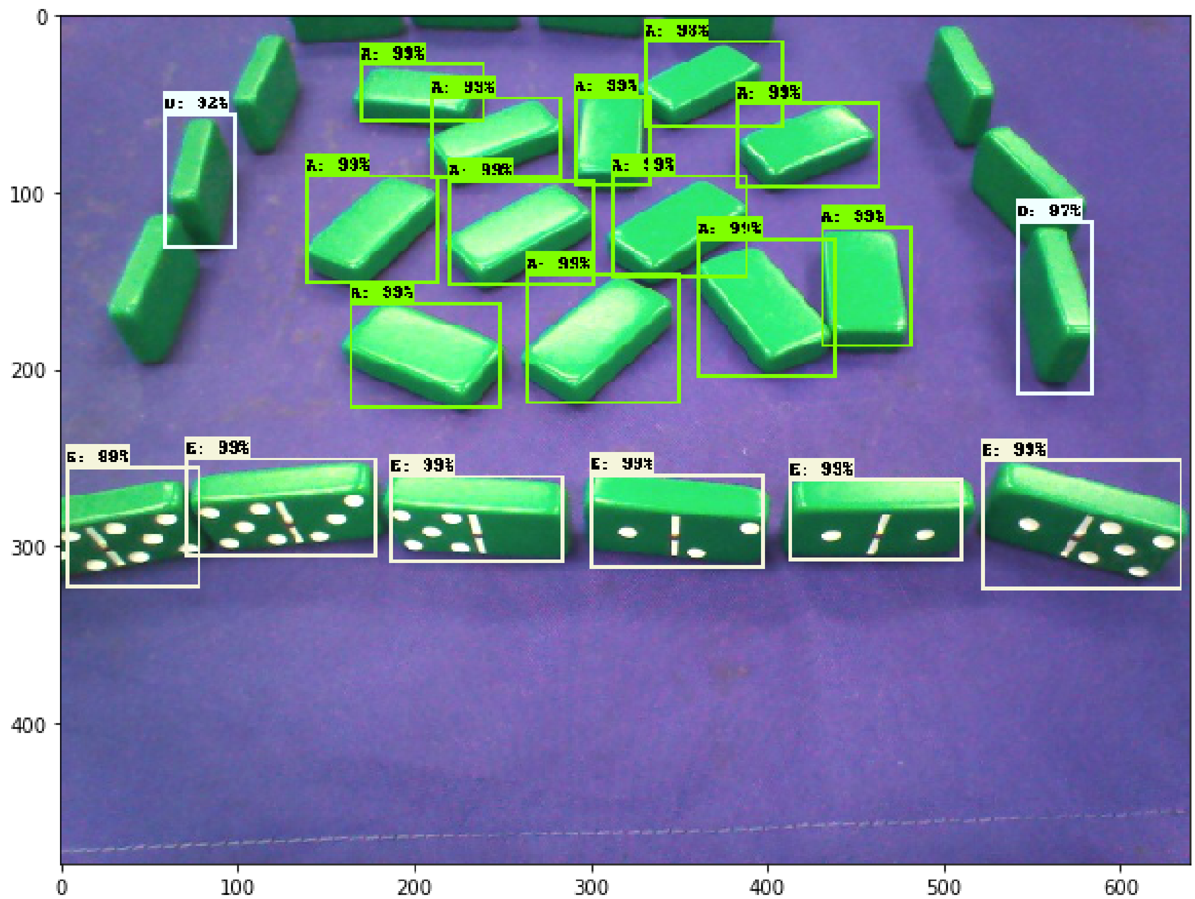
Pretrained models were used as a starting point to help quicken the learning process and a lot of models were used to train on the training dataset. Initial results were bad and unpredictable.

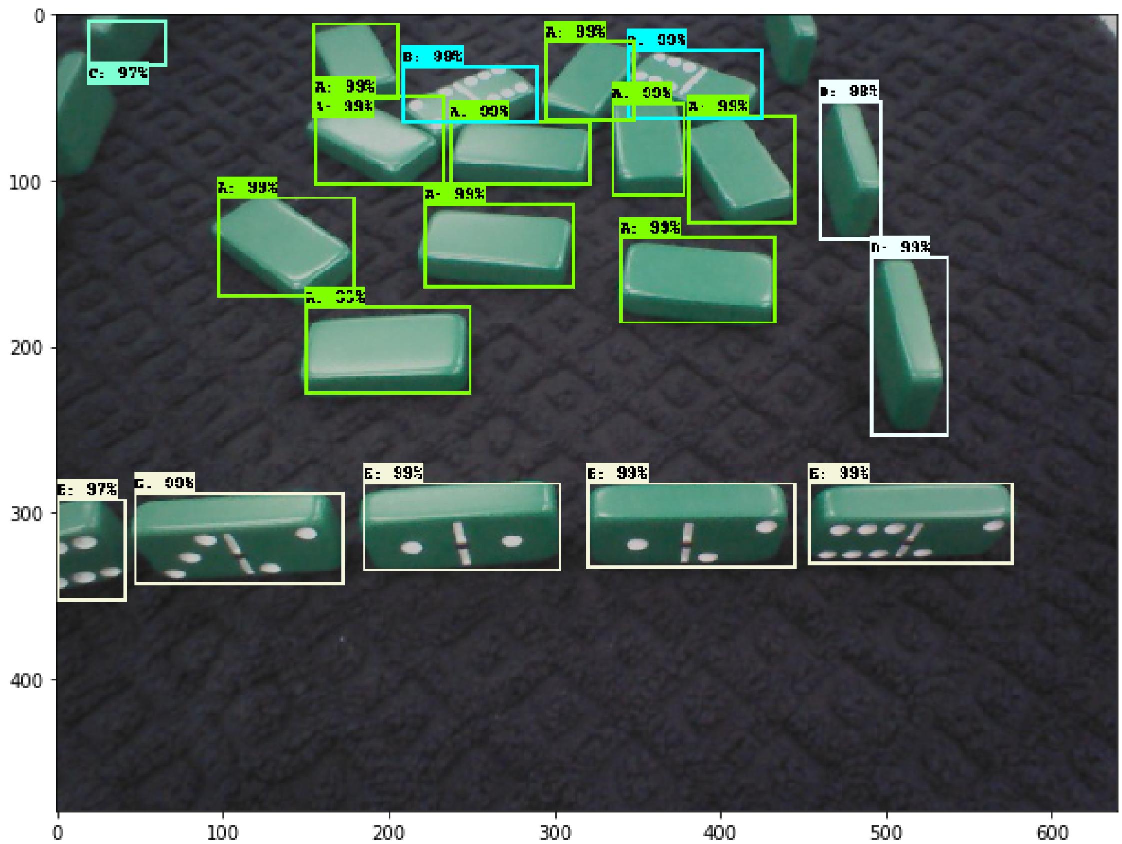


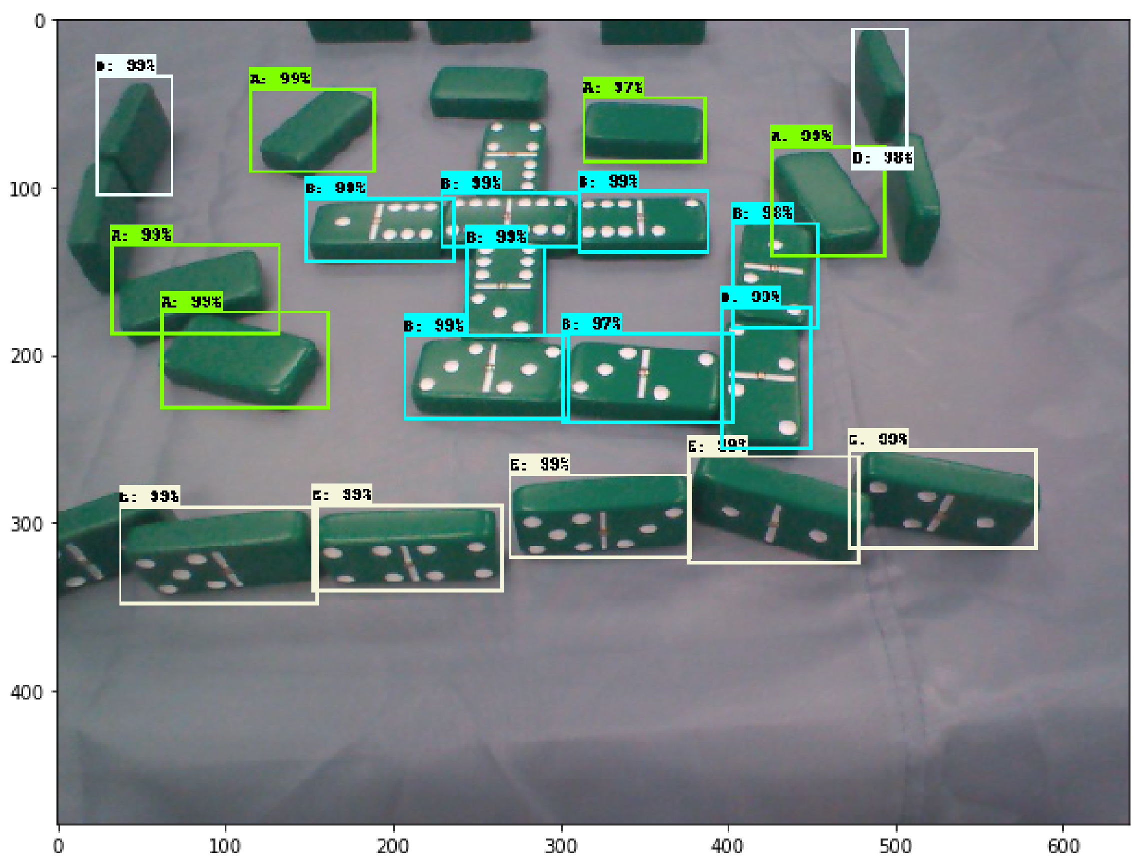
Later experiments include tuning with evaluations conducted on validation dataset and fine tuning the model to get better results. As the models took a long time to train, each modification/adjustment took a long turn around time. Now with our current model we have significantly better results.

Results (tables, figures and 1-3 paragraphs)

Following are the evaluations performed on the test dataset with our current model.


0
100
200
300
400
0
100
200
300
400
500
600
D: 92%
D: 97%


0
100
200
300
400
0
100
200
300
400
500
600


0
100
200
300
400
0
100
200
300
400
500
600

The encoding used in the training and results is as follows:

Description	Code
Laying and Facing down	A
Laying and Facing Up	B
Standing and Toward left	C
Standing and towards right	D
Standing and facing Camera	E
Standing and facing away from camera	F

The source files and data can be found here:

Image pips –

https://drive.google.com/open?id=1EVlge23N4FHRklr2L5K6lt1eqZtKKhc0

Image classes –

https://drive.google.com/open?id=1S3-C-sr4PosHWT1YIEi8zUdTfD4Y33AS

There are no available performance metrics for such and application for baseline or state of the art comparisons. However, the baseline could just be a random guess of one of the classes, but that would be very low as the joint probability of randomly guessing the image.

Discussion

The model resulting from this project is a novel addition as there are no pretrained models available to detect dominoes and identify the position and its physical state. So there are no clear comparison that could be drawn with other models. However the standalone evaluation metrics highlight that the model performs with a good degree of accuracy and recall. We can learn the architectural complexity in building a new model to be able to classify and learn complex features like the physical stance and face values for lesser common objects like dominoes.

The promising results from this project can be extended to detect the play state in game of dominoes. Even further extension could be to process all the information generated and learn a possible move in a given board state to progress in the game. This can also be easily extended for other applications and customized/tuned to the requirements.

Conclusion (2-3 paragraphs)

Some of the challenges in object detection with respect to performing under noisy environments and deal with finer details like counting face-value in images with multiple similar objects has been addressed in this project. Appropriate use of training, validation and test datasets ensured the model did not overfit nor give erroneous performance. The model also generalized well as witnessed in testing with examples with noisy backgrounds to a reasonable extent.

Acknowledgments

We would like to acknowledge our professor Dr. Rodney Nielson for all his insights and guidance all along the way. Also, the TA Namratha for her timely assistance and pointers. Special mention to our dear friend Pavan for his contribution in annotating parts of the data for this project and the encouragement from all the peers and friends and can never give google enough credit.

References ...

Tutorial:

https://pythonprogramming.net/introduction-use-tensorflow-object-detection-api-tutorial/

API
https://github.com/tensorflow/models/tree/master/research/object_detection

https://towardsdatascience.com/how-to-train-your-own-object-detector-with-tensorflows-object-detector-api-bec72ecfe1d9

Code Acknowledgements:

XML TO CSV - https://github.com/datitran/raccoon_dataset/blob/master/xml_to_csv.py

CSV TO TFRECORD - https://github.com/datitran/raccoon_dataset/blob/master/generate_tfrecord.py

Image Annotation - labelImg https://github.com/tzutalin/labelImg