Using Machine Learning with Tensor Flow That Uses K-Means-Clustering To Detect Threat Objects

Maanas Peri

Eastlake High School.

Research Problem

BACKGROUND

- The United States has more mass shootings than any other country (**8,942 total incidents of gun violence in 2018 up to March 2nd**)
- Between 1982 and 2011, a mass shooting occurred roughly once **every 200 days**. However, between 2011 and 2014 that rate has accelerated to **every 64 days**
- **List of Recent Mass Shootings excluding casualties**
  - Stone Man Douglas High School Florida shooting Feb 2018 left **17** kids and adults dead
  - Las Vegas Shooting in 2017 where **58** people were killed
  - Orlando Night club shooting in 2016 where **49** people were killed
  - Sandy Hook Elementary School shooting in 2012 where **27** innocent kids were killed

PROBLEM

- Currently there is no single device that can track a weapon
- People are almost, always unaware of threat weapons in their vicinity
- Delayed detection of a mass public shooter exponentially increases the risk to human life.
- Most of the detection technology is both expensive and/or harmful to human body (X-Rays, Infrared Rays, Millimeter Wave Scanners)

CRITERIA FOR SOLUTION

- Create an application that can classify (non-concealed) firearms quickly and accurately
- Classify firearms in 3 categories Pistol, Rifle, Revolver.

PROPOSED SOLUTION

- Implement artificial intelligence by deploying machine learning algorithms from TensorFlow to detect firearms. This is done using K-Means Clustering.
- Ex: This application can be used as a security device in schools to detect a threat object.

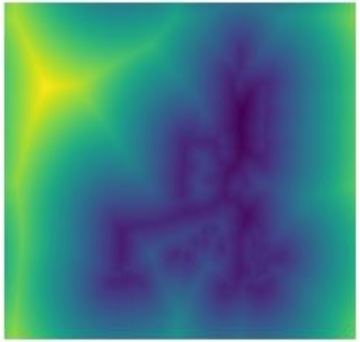
CONSTRAINTS

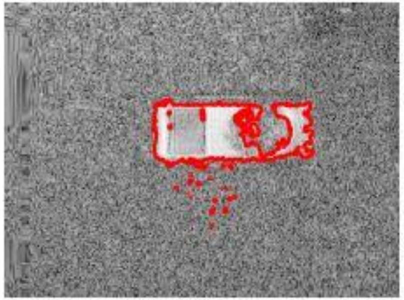
- Terahertz cameras were expensive and the data set for the images was limited.
- Chanvese Active Contour Algorithms using needed processing time to identify an object which could not be classified easily

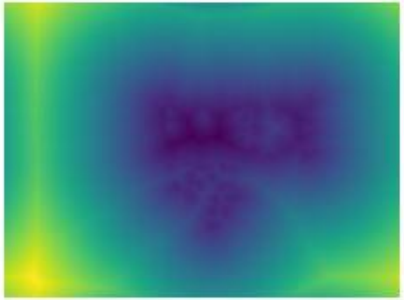
Explored Alternative Solutions

- Terahertz images can be captured when objects are concealed. These rays are non-ionizing and are harmless to humans.
- The images have low (signal to noise) ratio and normal algorithms cannot identify the objects.
- Terahertz is a research topic at MIT and the devices that produce these images are very expensive (to the tune of 100 thousand USD)
- Chan Vese contour algorithm was used to identify the objects, however the training data set is very hard to get and classification of these objects is not easy.
- Hence the project was moved to leverage a more open source and less expensive classification on Tensorflow.

Terahertz THz Images

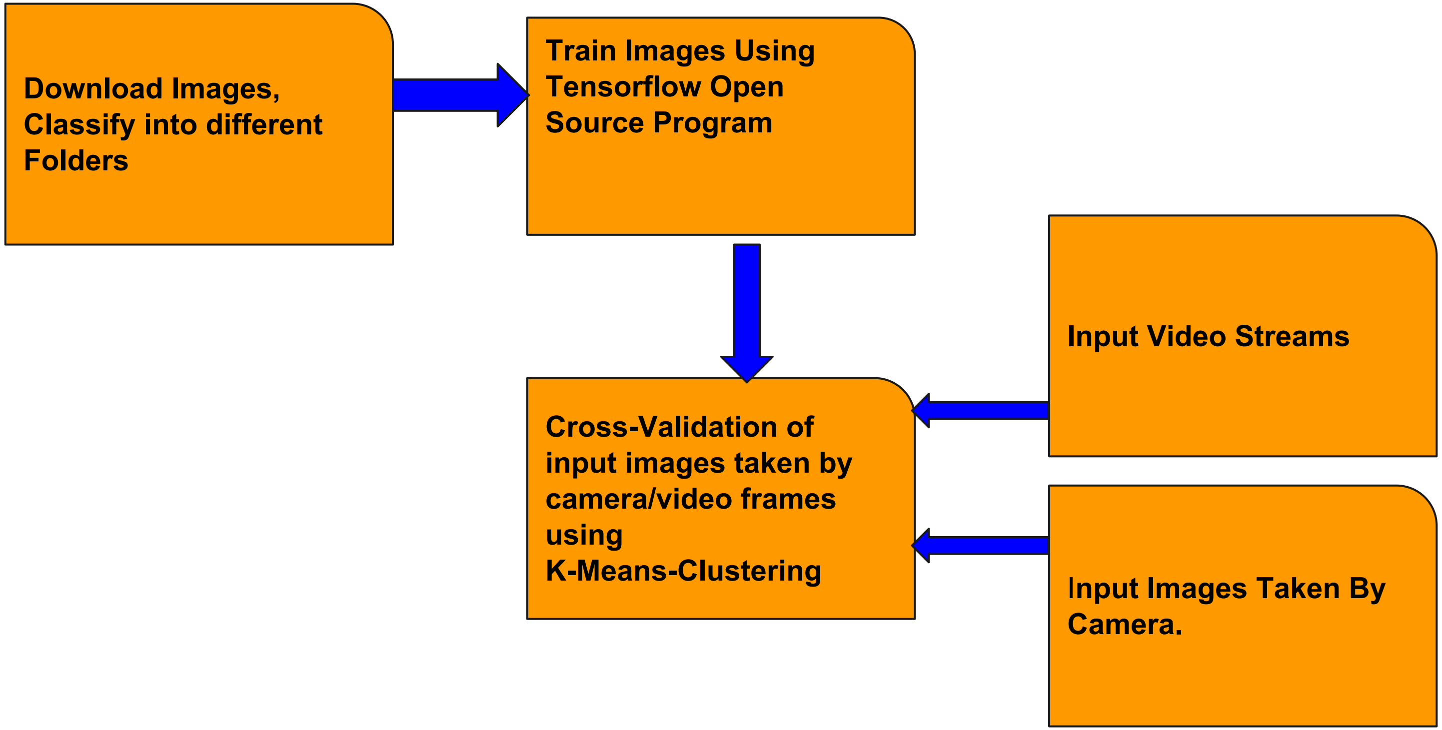






Design Methodology

FLOW CHART


Download Images, Classify into different Folders
Train Images Using Tensorflow Open Source Program
Input Video Streams
Cross-Validation of input images taken by camera/video frames using K-Means-Clustering
Input Images Taken By Camera.

TRAINING DATA SET AND TENSORFLOW

- ❖ To train the algorithm using Tensorflow, all the Guns category data was downloaded from IMFDB and Google sites. http://www.imfdb.org/wiki/Category:Gun
- ❖ Tensor flow Master was downloaded from Github https://github.com/tensorflow/tensorflow

TRAINING DATA PREPARATION

Dataset was split into

60% Training Data

20% Cross Validation

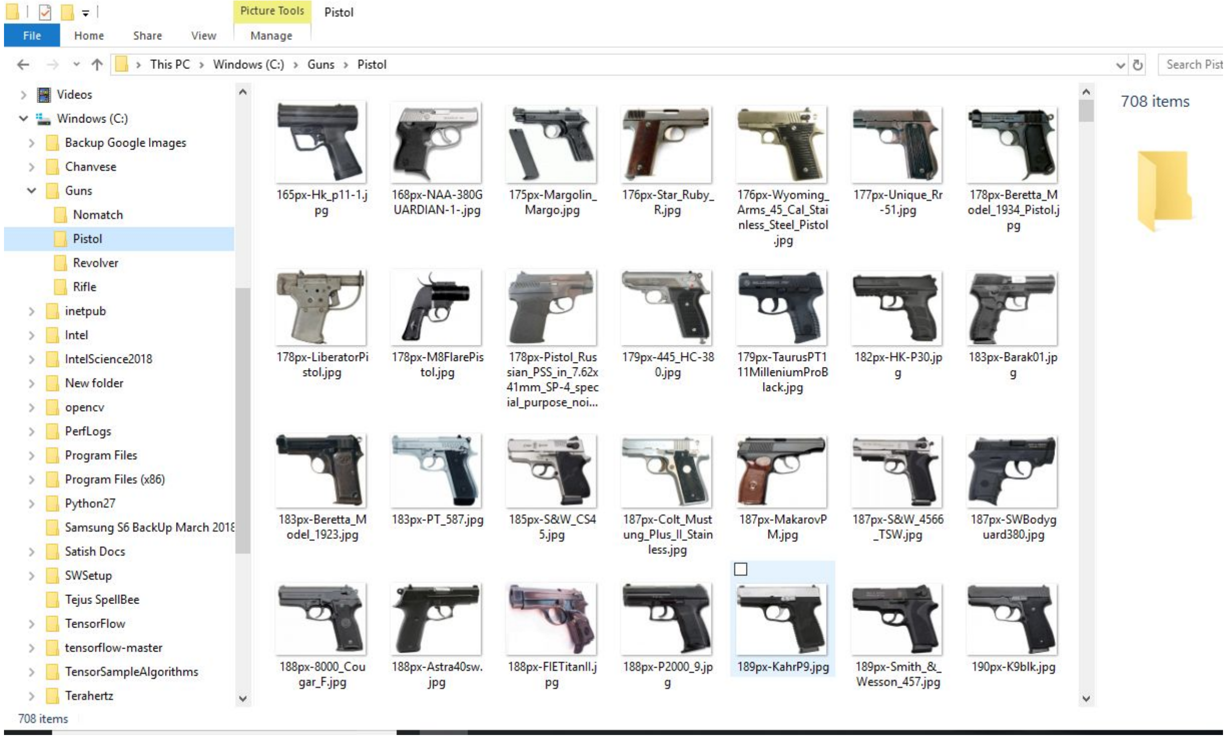
20% Hidden Test for final validation

Our dataset totalled up to 2,856 images for training

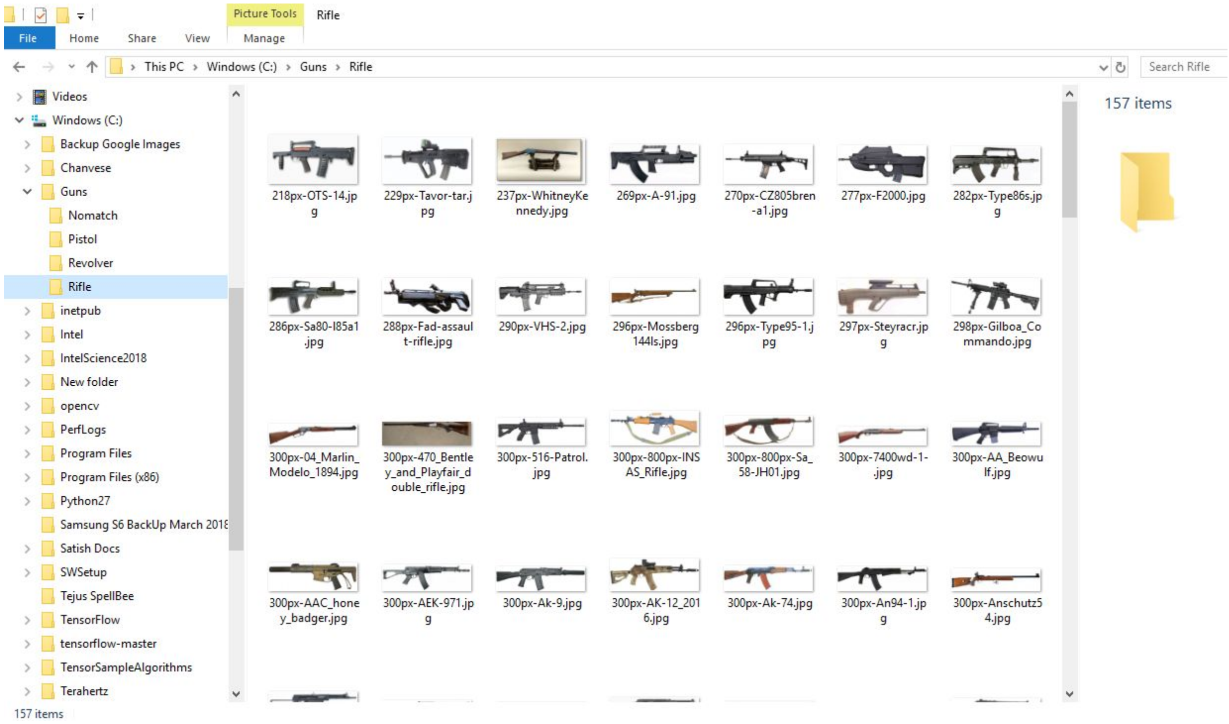
EXECUTION AND TESTING OF TENSOR FLOW

- ❖ Trained the Algorithm with the data downloaded.
  - ➢ c:\tensorflow-master>python tensorflow/examples/image_retraining/retrain.py --image_dir c:\Guns
- ❖ Run the Target Object against the training data set.
  - ➢ python tensorflow/examples/label_image/label_image.py --graph=/tmp/output_graph.pb --labels=/tmp/output_labels.txt --input_layer=Mul --output_layer=final_result --input_mean=128 --input_std=128 --image=C:/chanvese/Gun2.jpg

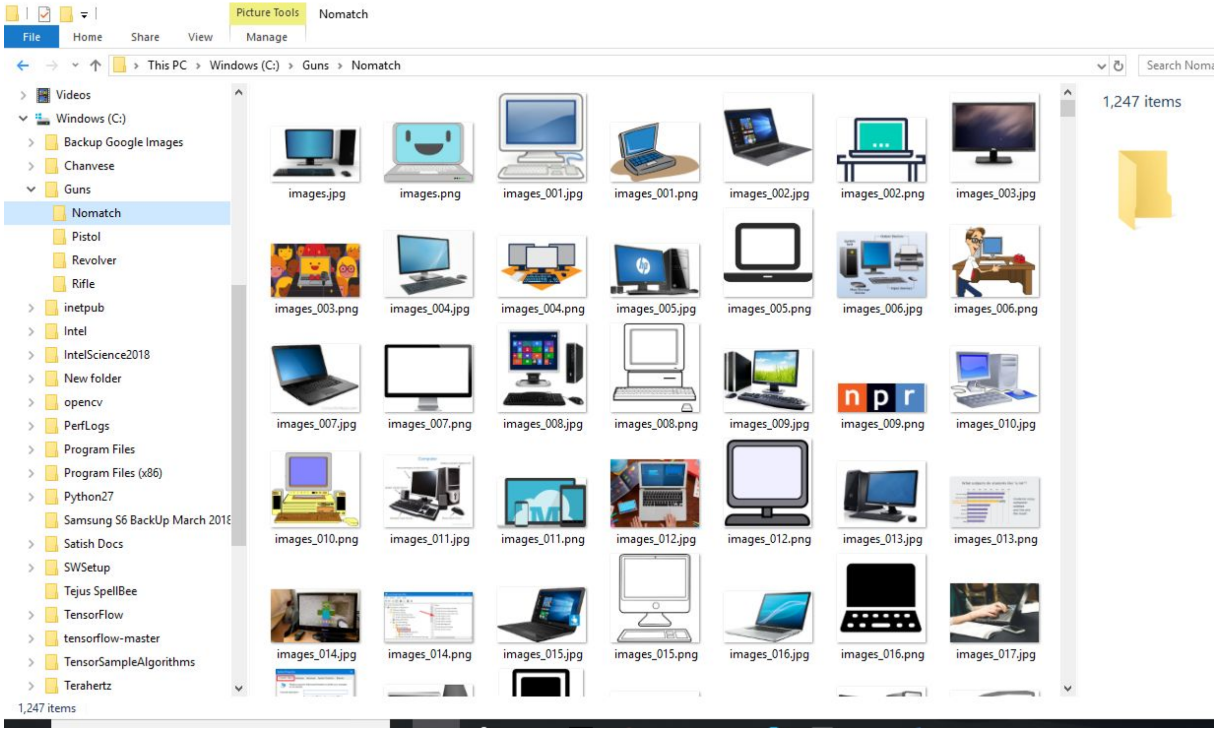
TESTING DATA SET 1



TESTING DATA SET 2



TESTING DATA SET 3



RESULTS

Image Name	Revolver %	Pistol %	Rifle %	No Match %
ManwithKnife	0.5%	0.05%	0%	99%
iPod	3.9%	1.7%	0%	94%
Revolver	98.12%	1.8%	0%	0%
Grandma	0%	0.2%	0.017%	99.76%
JamesBond	28%	41%	7.7%	22%
Maanas	0.5%	0.1%	0%	98%
ManwithGun1	53%	45%	0%	1%

Conclusion

IMPACT AND BENEFITS

- Low cost IOT device to classify threat objects
- App can be mounted on drones to create a secure virtual boundary
- Can be extended to Video streams by using existing infrastructure like surveillance cameras
- Future K-Means clustering optimization

NEXT PHASE

- Put my algorithm in a docker container
- Port the application into android, iOS devices
- Capture all the video streams split them into frames and classify
- Mount a simple android device in front of my school and classify threat objects
- Possibility to extend the data model to identify personnel traffic (student vs non-student) by building student profile at my school

Concepts

TENSOR FLOW

- ★ TensorFlow™ is an open source software library for numerical computation using data flow graphs
- ★ Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them
- ★ The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API
- ★ TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research

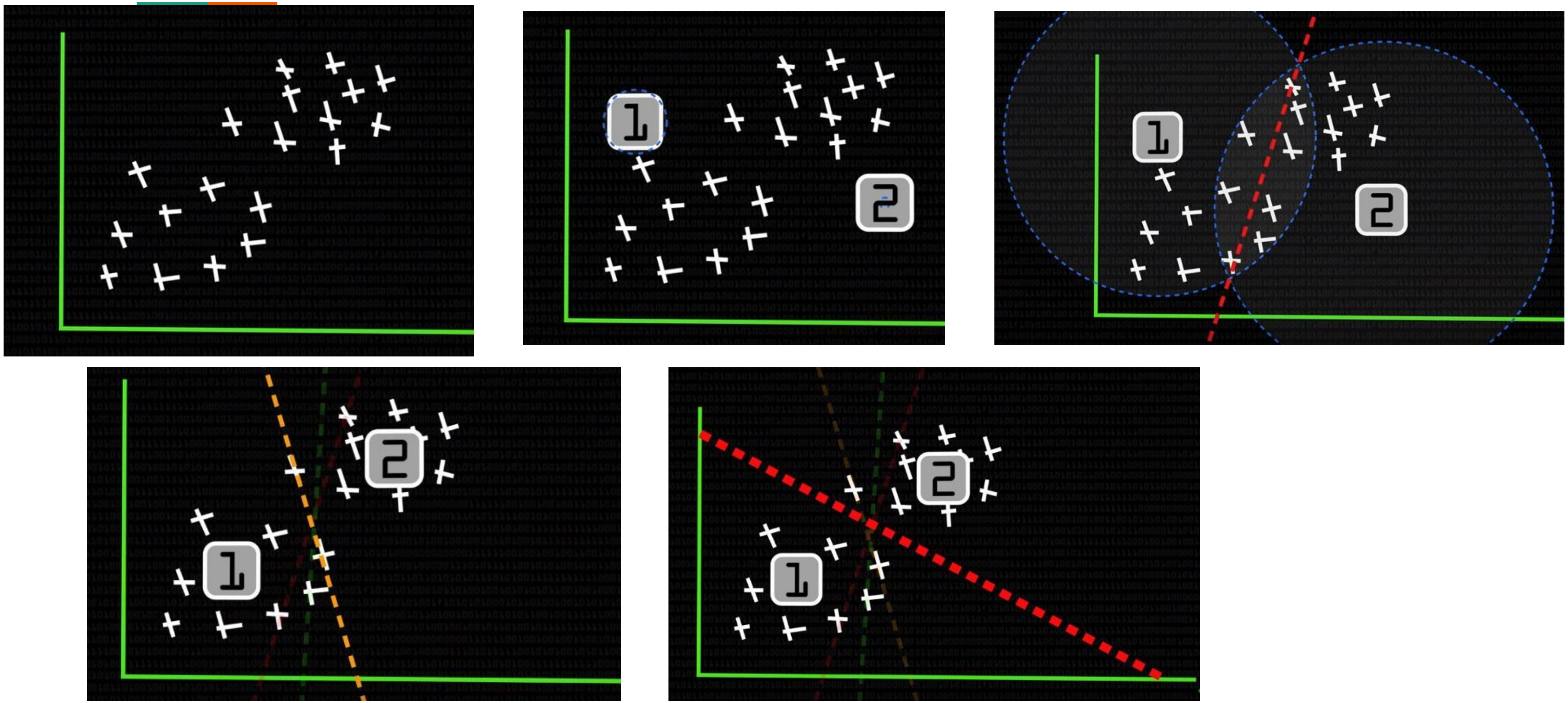
K-MEANS CLUSTERING

- K-means clustering is used when you have unlabeled data (i.e., data without defined categories or groups)
- The goal of this algorithm is to find groups in the data, with the number of groups represented by the variable K
- The centroids gradually move towards concentrated areas of similar data (using Euclidean Geometry), in order to discover a consistent pattern
- **Our k-value was set to 4. (Pistol, Rifle, Revolver, No-Match)**
- The algorithm works iteratively to categorize each data point to one of the categories above based on feature similarity

K-MEANS CLUSTERING PROCEDURE

1. Randomly select k data objects from dataset D as initial clusters
2. Calculate the distance between each data object and all k cluster centres
3. Assign data object the centroid
4. For each cluster, recalculate the cluster center by finding the average value (hence the name, K-Means-Clustering)
5. Repeat until there is no change in the centroid movement

K-MEANS CLUSTERING SEGMENTATION



OPTIMIZATION OF K-MEANS CLUSTERING

- ★ Objective: Reduce the extra time to calculate the distance from each cluster's center in each iteration.
- ★ Theory: If the distance of a data object from a new cluster center is smaller than the distance of the data object from the previously standard K-Means-Clustering model, then it SHOULD belong to the same cluster. This means there is no need to calculate the distance of the data object from other cluster center.