Logo Detection Using Machine Learning versus Affine Scale-Invariant Feature Transform

Matias Lopez [†∂], Lucia Martos[†], Alex Zaldastani[†]
[†] Department of Electrical Engineering, Duke University, Durham, NC [∂] Department of Biomedical Engineering, Duke University, Durham, NC

Duke UNIVERSITY | ELECTRICAL & COMPUTER ENGINEERING

➢ Abstract

Our goal is to allow quick and efficient logo recognition in video. We work closely with a company that aims to aggregate data about product placement through logo detection. We seek to better the detection algorithm with two separate attempts. The first is to use MATLAB powered enhancing algorithms learned in class to clean the input image into an Affine Scale-Invariant Feature Transform (ASIFT) algorithm. The second is making machine-learning classifiers using the best possible technique, we used an algorithm named "YOLO." The success of our first attempt with enhancement algorithms was minimal due to the noise that came with the sharpened image. The second attempt has promise but did not get a great enough dataset to be immediately successful.

➢ Objectives

- Improve the matching that Affine Scale-Invariant Feature Transform (ASIFT) has while only focusing on augmenting the input on the algorithm
- Create an enhancement algorithm that increased the matching within ASIFT
- Choose the best approach to training classifiers with machine learning techniques
- Allow the chosen best approach to successfully build a classifier.

➢ Conclusions

- Enhancement algorithms created too much noise to benefit ASIFT matching
- Noise is suspected to come from image compression artifacts.
- YOLO is a solid approach to training classifiers due to its speed in exchange of accuracy
- YOLO has promise must more work must be done to get a stronger dataset

➢ References

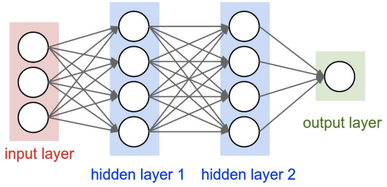
[1] "Image and Video Processing: From Mars to Hollywood with a Stop at the Hospital - Duke University." Coursera. Duke University, n.d. Web. 18 Apr. 2017.
[2] AlexeyAB. "AlexeyAB/darknet." GitHub. N.p., 16 Apr. 2017. Web. 18 Apr. 2017.
[3] "Dataset: FlickrLogos-32." FlickrLogos-32 Dataset (FlickrLogos). N.p., n.d. Web. 18 Apr. 2017.
[4] Google Groups. Google, n.d. Web. 18 Apr. 2017.
[5] "Machine Learning - Stanford University." Coursera. Stanford University, n.d. Web. 18 Apr. 2017.
[6] Nlintz. "Nlintz/TensorFlow-Tutorials." GitHub. N.p., 03 Apr. 2017. Web. 18 Apr. 2017.
[7]"Partial Differential Equations | TensorFlow." TensorFlow. N.p., n.d. Web. 18 Apr. 2017.
[8] The PASCAL Object Recognition Database Collection. N.p., n.d. Web. 18 Apr. 2017.
[9] The PASCAL Visual Object Classes Homepage. N.p., n.d. Web. 18 Apr. 2017.
[10]Redmon, Joseph. "Darknet: Open Source Neural Networks in C." Darknet: Open Source Neural Networks in C. N.p., n.d. Web. 18 Apr. 2017.
[11] Redmon, Joseph. Train a Classifier on CIFAR-10. N.p., n.d. Web. 18 Apr. 2017.
[12] Redmon, Joseph. YOLO: Real-Time Object Detection. N.p., n.d. Web. 18 Apr. 2017.
[13] "Start Training YOLO with Our Own Data." Guanghan Ning's Blog. N.p., 19 Oct. 2016. Web. 18 Apr. 2017.
[14] Thtrieu. "Thtrieu/darkflow." GitHub. N.p., 12 Apr. 2017. Web. 18 Apr. 2017.

➢ Acknowledgements

This work was supported by Duke's ECE590: Image and Video Processing. Dr. Sapiro was a great help in pointing us the correct directions. This project also could not have been accomplished without the structure and guidance of the IVI team.

➢ Machine Learning

Training Classifier Algorithms

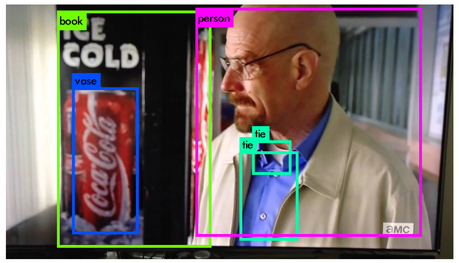


- **IBM Watson** (Simplest):
  a) Go to https://visual-recognition-demo.mybluemix.net/train
  b) Upload a dataset of images that you want to train on, specify positive and negative examples
  c) Simply upload an image, the output will then be a JSON file which says which indicate classifiers have the greatest matches
- **TensorFlow**: TensorFlow is an open-source software library for numerical computation using data flow graphs. We simulated logistic regressions, linear regressions, etc. (see in references)
- **Darknet**: Open source neural network framework written in C and CUDA.
- **Darkflow**: We were recommended to use Darkflow, a translation of Darknet to TensorFlow, by people who had worked with TensorFlow . This method gave several issues so it wasn't explored further.
- **YOLO**: This algorithm, built on Darknet, was chosen because we were looking to build a classifier that was really fast at a minimal cost to accuracy, since, ideally, accuracy was going to be done by ASIFT at the cost of speed. Previously object detection repurposes classifiers to perform detection. However, now we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evolution.

You Only Look Once

a) We built ours from the original YOLO net given, changing the number of classifiers, C, and filters, F.

$$F = (C + 5) * 5$$



b) Create a *.data* file with the number of classes and the paths to the *.txt* files with lists training and validation data
c) Change the *.names* file to the classifiers being trained
d) Create *.txt* files with the paths of the images being trained and validation data
e) Put training data in the path specified, requires *image.jpg* and *image.txt*, annotated in the format **<object-class> <x><y> <width> <height>**, where everything is in ratios
f) Give a sample weight file so it can train on

- Although the model was set up successfully, it did not get any meaningful results because dataset was not significant enough or at least too small.

➢ Methods: ASIFT

- Scale Invariant Feature-Transform is a good algorithm to detect if a feature like a logo is present in an image.
- Affine SIFT or ASIFT is used instead because SIFT is limited to features with the same orientation as the original image.
- ASIFT performance drastically drops when the image quality drops, and improves when the image region of interest is isolated.

Enhancing Algorithms Used

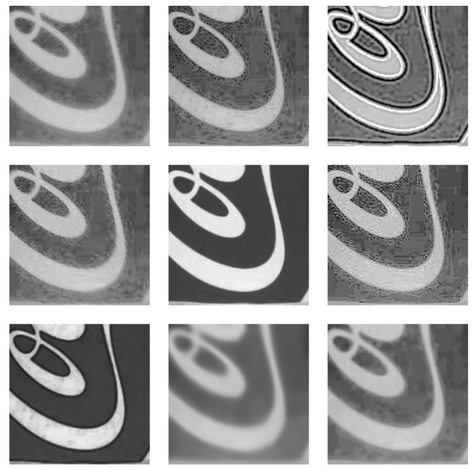
a) Custom affine anisotropic filter
b) Gradient multiplied filter
c) Gaussian deconvolution
d) Directional gradient weighted average Filter
e) Non-equalized median filter
f) Color contour averaging
g) Non-equalized Inverse gradient dependent median filter
h) Reference trained sparse encoding
i) Equalized median filter

- The reference image tested on is a good quality image of a Coca Cola can.
- The test image is a scene from a TV show featuring a similar can, and a third image zooms in the can featured in this scene.

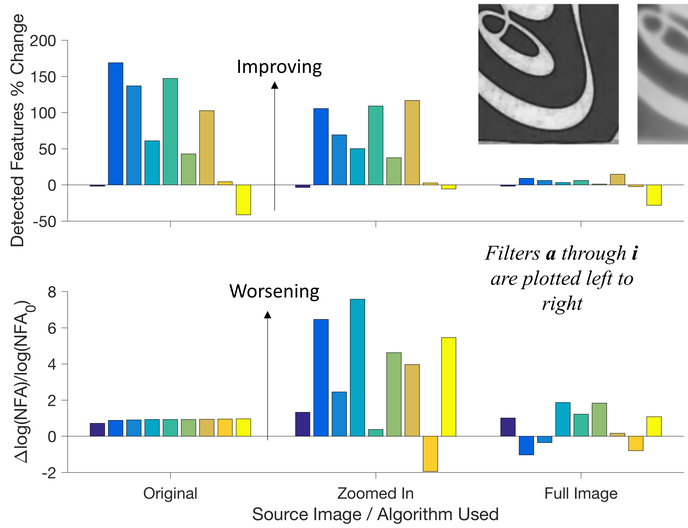
➢ Results: ASIFT

The largest problem with enhancement is that while features were sharpened, so was the noise in the image.

*Filters **a** through **i** are shown left to right, top to bottom*



- Detection of the Coke can in the test image was not successful.
- ASIFT was not robust enough to handle finding a logo in different images when the resolution is low.
- The only improvement in detection of low quality images came at a large computational cost



*Filters **a** through **i** are plotted left to right*

Although very computationally costly, the trained sparse encoder performed the best at improving detection.

However ASIFT did not find any significant matches in the test image, meaning the improvements were still unsuccessful