Comparing the Effects of Annotation Type on Machine Learning Detection Performance

James F. Mullen Jr.
james.mullen@raytheon.com

Franklin R. Tanner
franklin.r.tanner@raytheon.com

Phil A. Sallee
philip.a.sallee@raytheon.com

Raytheon Corporation
22270 Pacific Blvd
Dulles, VA

Abstract

The most prominent machine learning (ML) methods in use today are supervised, meaning they require ground-truth labeling of the data on which they are trained. Annotating data is arduous and expensive. Additionally, data sets for image object detection may be annotated by drawing polygons, drawing bounding boxes, or providing single points on targets. Selection of annotation technique is a tradeoff between time to annotate and accuracy of the annotation. When annotating a dataset for machine object recognition algorithms, researchers may not know the most advantageous method of annotation for their experiments.

This paper evaluates the performance tradeoffs of three alternative methods of annotating imagery for use in ML. A neural network was trained using the different types of annotations and compares the detection accuracy of and differences between the resultant models. In addition to the accuracy, cost is analyzed for each of the models and respective datasets.

1. Introduction

Over the past few years, a tremendous amount of research has focused on improving deep neural network (DNN) architectures and their efficacy for applications in computer vision. A majority of these works focus on the architectures themselves, and less has been done to explore the data which makes these networks run. Since neural networks can only be as good as the data they are trained on, the input data is possibly the single most important part of the neural network. For supervised methods, this includes the annotations that are used as ground truth.

Today's deep networks require an even greater burden of labeled data than previous generations [4, 30]. Annotating datasets for use in supervised machine learning is expensive and time consuming due to the amount of data often required. While a number of works have looked at the effects of noise [26, 11, 19, 12, 30], and adversarial perturbations [18, 16, 17, 14] in the training set, little information is currently available on the tradeoffs between image annotation types and the resulting performance of the network. Approaches for manually annotating imagery, such as with polygons, bounding boxes, or target centroids, have considerably different annotation cost due to their varying complexity, and the utility of each may depend on the use case. Therefore it is essential to understand the tradeoffs between cost and performance for each type so as to select the most appropriate method for a given use case. Cost per label information can be gathered from commercial annotation companies.

To the authors' knowledge, no prior work compares the effect of annotation types on neural network performance. [5] describes the need for exploring different data annotations and a method of sequentially feeding higher fidelity annotations when the prior annotation was not sufficient for a traditional segmentation model. The lack of work exploring annotations type with deep networks is surprising, considering the cost of image annotation, amount of images needed for training, and the widespread ramifications of such research for appropriately tailoring annotations to the demands and goals of a given application. The potential to cheapen data annotation, even for small numbers of use cases, could be felt throughout the computer vision industry.

2. Background

The most prominent machine learning methods require ground-truth labeling of the data on which they are trained. Supervised learning using deep convolutional neural networks represents the current art in machine vision. When trained with sufficient quantities of properly labeled data, these networks have been shown to be highly accurate, close to or potentially exceeding human performance for many classification and detection tasks [6, 13, 10, 24, 8, 9]. For the current investigation, we focus on the challenge of detecting vehicles in satellite images.

Despite its potential, supervised learning carries a heavy data requirement that is often a limiting factor for success. Labeling sufficient amounts of data may be prohibitively expensive, especially when using the highest fidelity methods. Commercially, datasets become a competitive advantage and barriers to entry grow ever higher for new players. For academic and commercial researchers alike, annotating datasets can become an extremely costly activity, setting research back months or taking up significant portions of budgets. In many of these cases researchers may pay for more annotation fidelity than they need. If high fidelity annotations are not required to achieve the necessary performance, cheaper options may represent a significant cost or time savings. We present a workflow that supports the type of cost benefit analysis needed to realize these savings.

While a lot of work in computer vision has labeled images with a single class per chip, this is only appropriate when a single object dominates the field of view. Here, we focus on pixel-wise annotation methods that are more useful for image segmentation and semantic labeling, including the detection and classification of multiple objects in a scene. The three most common pixel-wise annotation methods used for this purpose are polygons, bounding boxes, and centroids.

Polygons, closest to ground truth, often require an annotator to identify many points around a target of interest to fit its shape. This makes polygons the most time consuming, and thus the most expensive of the three annotation types. Bounding boxes are defined as the tightest fitting box around a target. These are more time efficient to produce than polygons, requiring only two clicks from the annotator, however they may encapsulate a large number of pixels outside of the true target area. Centroids are relatively trivial to annotate, requiring only a single point at the geographic center of the target. For this investigation, we extrapolate a circular "point-target" region around each centroid having the same area as the average target. This allows the network to train on a region instead of a single pixel. Similar to bounding boxes, these regions may not fit well to the identified target. While there are generally fewer false positive pixels with point targets than bounding boxes, they also contain false negative pixels where the circular regions clip the targets.

Pixel discrepancies resulting from bounding boxes and point targets may potentially hamper network performance, discouraging researchers from pursuing these data annotation methods. However, the cost savings for annotation makes them more attractive options otherwise. One commercial annotation company estimated that it takes humans six times longer to draw a polygon around a target and two times longer to draw a bounding box than simply putting a marker on a target centroid [27]. Another source shows the time difference between bounding boxes and polygons as 7 seconds per bounding box and 54 seconds per polygon [5]. Figure 1 shows a single vehicle annotated with the three methods described above.

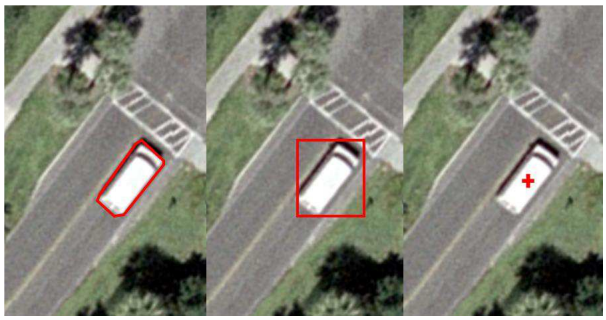

Figure 1. Images of targets identified with polygons (left), bounding boxes (middle), and target centroid (right). Image from [28]. Images courtesy of the U.S. Geological Survey.

3. Experimental design

We trained three separate networks, one for each annotation type (polygon, bounding box, and centroid), and compared network performance for detecting two classes (car / non-car) at each pixel location. The following describes the data, network architecture and evaluation metrics used in our experiment. An overview of our workflow, starting with separating and conditioning the data for use in network training as described in this section, is presented in Figure 2.

3.1. Data selection and mask creation

Our experiments are based on the Overhead Imagery Research Dataset (OIRDS) [28]. OIRDS is open source, freely available, and includes both centroid and polygon annotations, in addition to other vehicle features. For training the networks, we generated binary masks for each annotation type. Bounding boxes were calculated from the polygons, using the maximum and minimum x and y coordinates. For point targets, we generated a circle around each centroid with an area of the average target size. Despite the richness of OIRDS, it is still a very small dataset compared

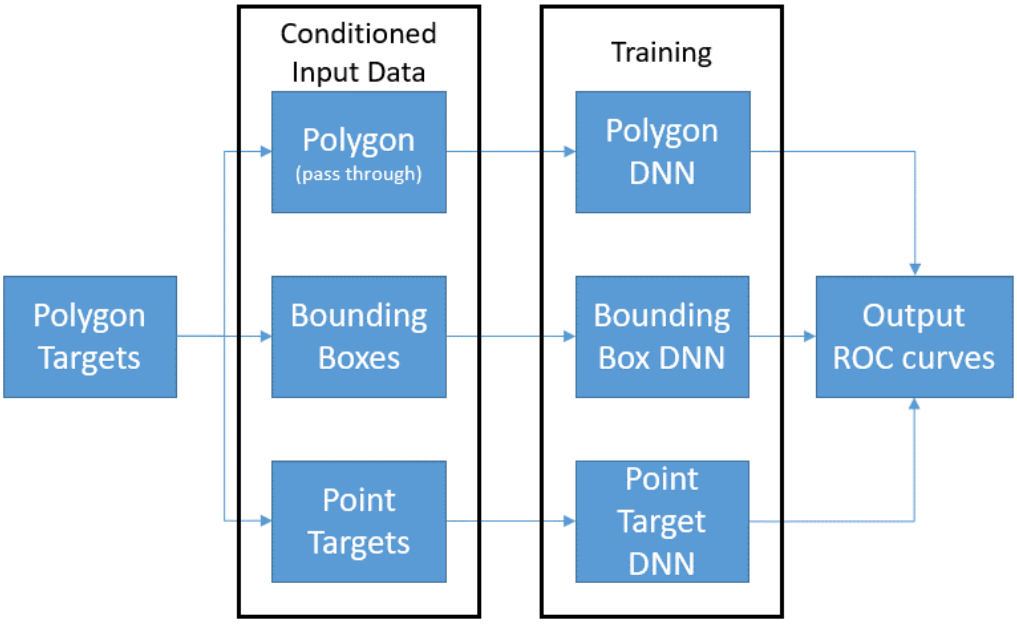

Figure 2. Image processing workflow starting with polygon targets, extracting separate data sets for each annotation type and then feeding each dataset into an individual Deep Neural Network before analysis with a ROC Curve.

to most modern datasets required for deep learning. The OIRDS contains approximately 1000 images with approximately 2000 targets versus ImageNet with its 14+ million annotated images [4].

3.2. Network architecture and implementation

We used the Overfeat [22] network architecture because of its simplistic architecture, ease of implementation, and competitive results. We then modified this network to retain its spatial dimensions, by setting padding [23] to "same," making sure that edges were not lost, as well as removing stride in the convolution and pooling layers. This maintained a 257 by 257 output size equal to the input image dimensions, where each output node represents the detected class for an image pixel. This is compared to our training masks, or ground truth information, as presented in Figure 3. The last two fully connected layers of the network were also removed to retain satisfactory spatial resolution for the pixel map output. Our pooling layers were max pooling and prior to our output layer we implemented a dropout [25] layer with probability of 0.5.

We used the Tensorflow [1] deep learning package for building and training the networks. Training was performed using RMS-Prop optimizer [21] with a softmax cross-entropy loss function [2] and a learning rate of 0.00005.

Each network was trained for two-hundred epochs with each epoch consisting of 192 batches of four images. The training subset of our dataset was eighty-nine percent of the dataset with the remaining eleven percent set aside for later testing and validation. Training took roughly 24 hours for each network on a NVIDIA V100 [15] graphics processing unit. While trained for two-hundred epochs, the network only saw incremental improvement after thirty epochs. Based on this we estimate that an effective network for testing could be trained in as little as 3.5 hours.

Figure 3. Example of, from left to right, an input image from OIRDS, a polygon input mask, and the output of the trained network for reference. See Figure 6 and 7 for input masks and outputs for the bounding boxes and point targets. Input Image (left) courtesy of the U.S. Geological Survey.

3.3. Evaluation criteria

To evaluate performance, we used a Receiver Operating Characteristic (ROC) curve evaluated at all pixel locations and also computed the area under the curve (AuC) [3, 29, 7]. For each network, a ROC curve was produced using its own respective annotation type that it was trained with as

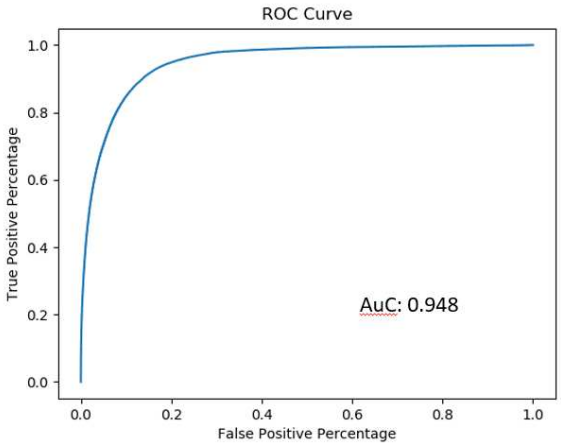

Figure 4. Hypothetical example of an output ROC curve and AuC value.

the truth, and also a curve using polygons as the truth. An example ROC curve is provided for reference in Figure 4. The ROC curve shows the ratio of the false positive rate to the true positive rate for a range of thresholds applied to the network score, yielding a means of comparing performance agnostic to a threshold value. A perfect ROC is a vertical line at 0.0 false positive rate and a horizontal line at 1.0 true positive rate, with an AuC of 1.

4. Results and discussions

This section presents and then explores our experimental results both quantitatively and qualitatively. All of our results were obtained using our separate test subset of the OIRDS dataset.

4.1. Quantitative evaluation

The right plot of Figure 5 shows ROC curves for each network evaluated with the type of annotations they were trained with (itself truth). E.g., bounding box network was trained and evaluated using bounding box annotations, while the point target network was trained and evaluated using point targets. All of the networks had an AuC value of over 0.9. As expected, the polygon network significantly outperformed both bounding box and point target networks with an AuC of 0.95. Although bounding boxes are considered a higher fidelity annotation than point targets,the point target network surprisingly outperformed the bounding box network in this evaluation.

The left plot in Figure 5 compares the ROC curves obtained when each network is evaluated using the polygon labels. From this, it can be seen that both the bounding box network and point target network exhibited significantly better results against the polygon truth than against their own truth labels. Although the polygon network still outperforms both of the other networks, all three now have similar performance, within .007 AuC. This indicates that the networks are not learning to paint the shape of the training labels. Rather, each network learns to paint detections that approximate the actual shape of the target object. This effect can also be seen qualitatively in the next section.

4.2. Qualitative evaluation

Qualitative results were extracted by creating an image out of the input masks and the output detections of the network. This was done while testing with each image producing an output detection tensor to be turned into an image. For visual clarity, the input and output masks were overlaid with the input image and the output detection mask was thresholded at 0.2 where any pixel of confidence over 0.2 was classified as a detection and made 1 (pure white) and

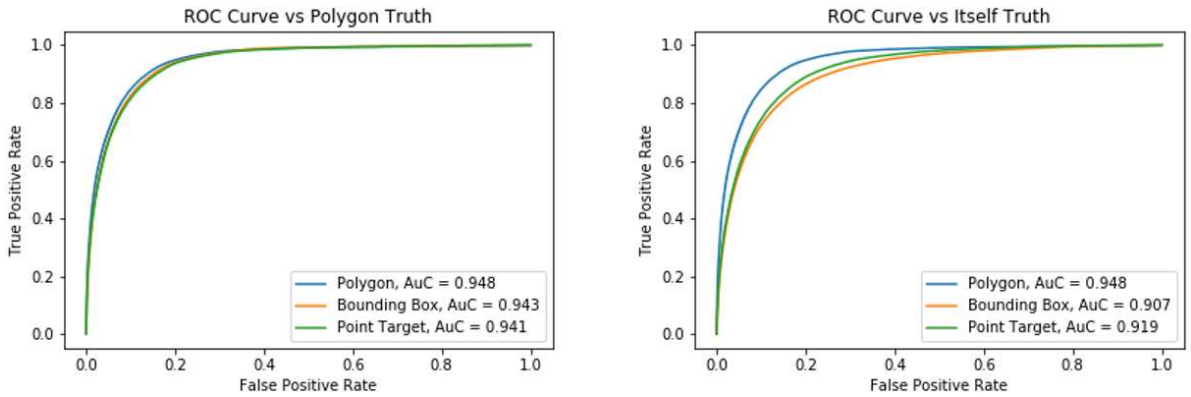

Figure 5. Output ROC curves and AuC values for each of the three networks with the polygon, bounding box, point target networks all presented together. The left chart is when the networks are compared with polygons as ground truth and the right chart is when the network is compared with the type of annotation it was trained on.

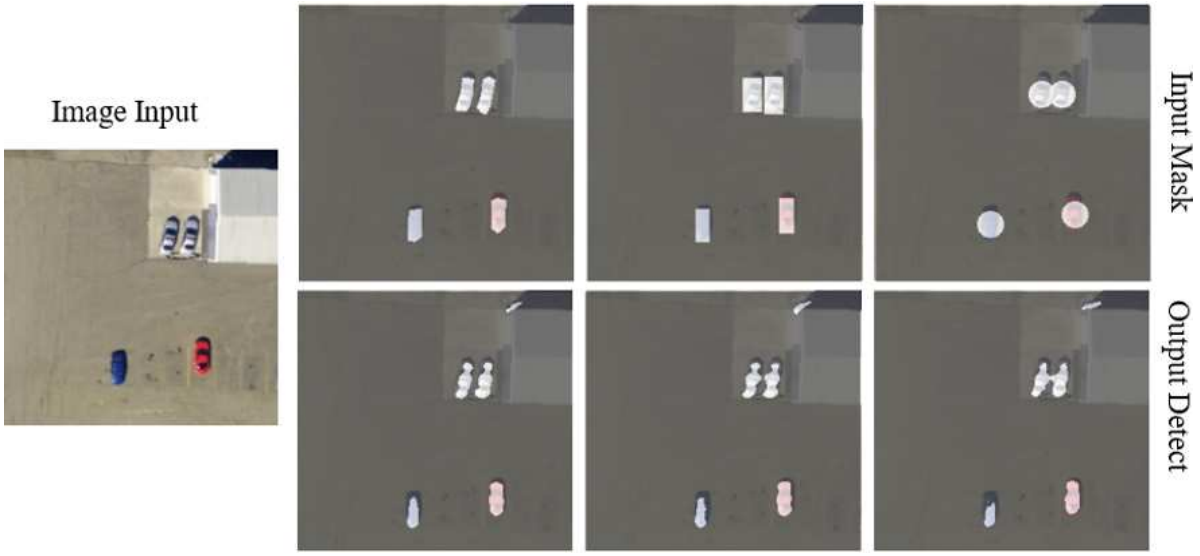

Figure 6. Example of each networks output detection. From left to right, the polygon network, bounding box network, and point target network. Input Image courtesy of the U.S. Geological Survey.

Figure 7. Example of each networks output detection with an occluded target and a significant false positive detection. From left to right, the polygon network, bounding box network, and point target network. Input Image courtesy of the U.S. Geological Survey.

any pixel of confidence lower than 0.2 was set to be zero (pure black).

When analyzed qualitatively, the similarities between the three networks, regardless of the type of data trained on, become even more apparent. Figure 6 and 7 exhibit some examples of the outputs of these networks. When looking at the network output masks alone, it is very difficult to discern any clear differences between the three networks. In Figure 6, you can see all of the targets captured effectively by each network. Interestingly, there is very little presence of "boxiness" with the bounding box network detections and similarly, no indication of circles in the point target network.

We observed a general increase in false positive detections for the bounding box and point target networks relative to the polygon network. While all three networks showed false positive detections at similar locations, the detection area for these false positives was typically larger for the bounding box and point target networks than for the polygon network. An example is presented in Figure 7.

5. Conclusions

Based on our results, we conclude that comparing annotation types should be an important step when assembling a dataset for deep learning applications. A sample methodology was provided to perform these comparisons, which may be expanded or adapted for other use cases. Our quantitative results show that all three of these networks do a remarkably similar job on the detection task with the polygon trained network exhibiting only marginally better performance than the bounding box trained network and the point target trained network. This evidence supports the consideration of bounding boxes or centroids for annotations instead of polygons for cost savings. The unexpected result that the bounding boxes and point targets perform better compared when evaluated with polygons than with their own masks shows that the mislabeled pixels do not cause the network to learn incorrect shapes of objects, but that the system still generally learns to paint the objects correctly. This indicates that annotation type may not be as important as previously thought in some cases. It may be appropriate to consider the difference between these annotation types in terms of label noise, where some pixels in each bounding box or point target annotation are incorrectly labeled. This is consistent with previous work that demonstrates the robustness of deep networks against noisy labels [20].

While we selected the polygons, bounding boxes, and centroids as our annotation types of interest, the choice of annotation type should be tailored to a particular application. Similarly, network architecture and evaluation metrics should be based on the application and type of data in use. Our selection of the Overfeat [22] network architecture and the area under the ROC curve metric may not be the best options for all use cases.

Overall, we have shown a clear method for researchers to evaluate their datasets, prior to paying for the annotation of the entire dataset. This will allow researchers to potentially save considerable amounts of money or alternatively annotate significantly more data with the same amount of money. For example, annotating 1M images with polygons at $0.06/polygon would cost $60K vs. $0.01/centroid at a cost of $10K. Figure 8 provides a graphical view of this cost difference. This knowledge will empower researchers and others to consider the annotation task as less daunting and allow them to continue exploring new and exciting use cases.

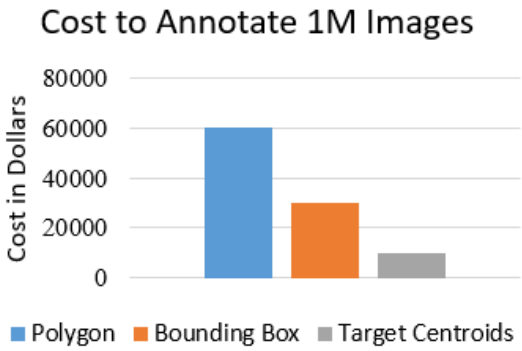

Figure 8. Estimated cost to annotate 1M Images using Polygons Bounding Boxes and Target Centroids.

5.1. Future work

There are a number of ways that this research could be expanded to provide more insight and better cost analysis for annotation types. The question remains whether additional point or bounding box labels could provide the same performance as polygons. To provide a more complete cost benefit analysis we would like to extend this experiment using additional bounding box and point target data to see if training with the additional data can fill the performance gaps present with the polygon trained network. If so, a definitive cost analysis could identify the lowest-cost annotation type for a given detection performance. This might support the potential for researchers to purchase more data with their budgets instead of the polygon annotations.

Additionally, more diverse datasets with targets of other shapes, as well as other annotation types should be explored. The OIRDS is a relatively small dataset, with the potential to create bias in the investigation based on the lack of diversity. A similar dataset of images taking obliquely would be particularly interesting to explore. One specific annotation type that would be interesting to evaluate is an ellipse. Ellipses could provide more fidelity than point targets or bounding boxes while taking a similar amount of time to annotate as a bounding box.

Network selection and tuning could be expanded upon by selecting different networks and tuning them specifically for the annotation type. Implementing networks designed to be fully convolutional could produce improved results as they are designed to produce pixel level output. Regularization and kernel sizes are examples of parameters that could potentially be tuned on the networks to help compensate for the variations between the data types.

On the analysis level, more work can be done to provide further quantitative analysis of the networks. The precision recall curve, and its respective area under the curve, is arguably better suited than ROC curves for evaluating detection of objects vs. pixels. Evaluating a more complete and diverse set of metrics would provide a larger and less biased picture of the strengths and weaknesses of different annotation options, providing a better basis for decisions about which annotation type to select for a given dataset.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and M. Kudlur. November. *Tensorflow: a system for large-scale machine learning*, 16:265–283, 2016.

[2] G. Authors. *Softmax Cross Entropy*. 2019.

[3] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[4] J. Deng, W. Dong, R. Socher, L. j. Li, K. Li, and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database*. In CVPR09, 2009.

[5] S. Dutt Jain and K. Grauman. Predicting sufficient annotation strength for interactive foreground segmentation. pages 1313–1320. In Proceedings of the IEEE International Conference on Computer Vision, 2013.

[6] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. arxiv. preprint, 2016.

[7] S. A. Israel. Performance metrics: how and when, geocarto international. *Vol.*, 21:2, 2006.

[8] S. A. Israel, J. M. Irvine, A. Cheng, M. D. Wiederhold, and B. K. Wiederhold. Ecg to identify individuals. *Pattern recognition*, 38(1):133–142, 2005.

[9] S. A. Israel, W. T. Scruggs, W. J. Worek, and J. M. Irvine. Fusing face and ecg for personal identification. *In Applied Imagery Pattern Recognition Workshop Proceedings*, 32:226–231, 2003.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning*. ICML, 2013.

[12] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and J. Li. Learning from noisy labels with distillation. pages 1928–1936, 2017 IEEE International Conference on Computer Vision (ICCV), 2017.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. October. In *Ssd: Single shot multibox detector. In European conference on computer vision . , Cham*, pages 21–37, 2016.

[14] S. m. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard. *Universal adversarial perturbations*. In Computer Vision and Pattern Recognition. IEEE, 2017.

[15] S. Markidis, D. Chien, S. W., E. Laure, I. B. Peng, and J. S. Vetter. Nvidia tensor core programmability, performance & precision. arxiv. preprint, 2018.

[16] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. In *International Conference on Learning Representations*, 2017.

[17] J. H. Metzen, M. C. Kumar, T. Brox, and V. Fischer. Universal adversarial perturbations against semantic image segmentation. arxiv. preprint, 2017.

[18] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, CA, 2016. San Jose.

[19] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. arxiv. preprint, December 2014.

[20] D. Rolnick et al. Deep learning is robust to massive label noise. preprint, arXiv, 2017.

[21] S. Ruder. An overview of gradient descent optimization algorithms. arxiv. preprint, 2016.

[22] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. arxiv. preprint, 2013.

[23] P. Y. Simard, D. Steinkraus, and J. C. Platt. August. *Best practices for convolutional neural networks applied to visual document analysis*, p. 958, 2003.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arxiv. preprint, 2014.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[26] S. Sukhbaatar and R. Fergus. Learning from noisy labels with deep neural networks. arxiv. preprint, 2014.

[27] F. Tanner. *"Annotation Information"*. 2018.

[28] F. Tanner, B. Colder, C. Pullen, D. Heagy, M. Eppolito, V. Carlan, O. C., and P. Sallee. *Overhead Imagery Research Data Set - An annotated data library and tools to aid in the development of computer vision algorithms*. in IEEE Applied Imagery Pattern Recognition Workshop, Washington D.C, 2009.

[29] K. Woods and K. W. Bowyer. Generating roc curves for artificial neural networks. *IEEE Transactions on medical imaging*, 16(3):329–337, 1997.

[30] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. *Learning from massive noisy labeled data for image classification*. In CVPR, pages 2691-2699, 2015.