Automated Detection of Archaeological Targets in LiDAR surveys

Final Thesis Defence

M. Olivier

Leiden University - ESIEA, November 2020

Introduction

Introduction

In this presentation I will summarise my work done during my final year internship at the department of Digital Archaeology, Leiden University, Netherlands, under the supervision of Wouter Verschoof-van Der Vaart, PhD candidate.

What exactly are we trying to achieve ?

Develop an object detector for use in Archaeology that:

- Works using LiDAR surveys
- Is able to detect 3 distinct classes of object
- Is (fairly) fast, with good precision and specificity
- Is integrated at least partially with a GIS for post inference analysis

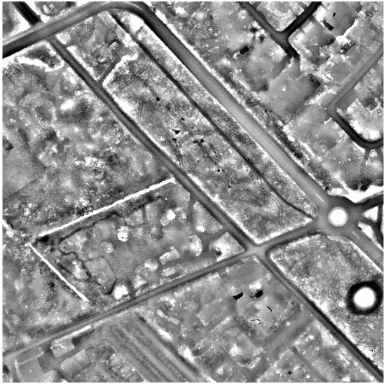
Table of Contents I

Table of Contents II

Elementary Concepts

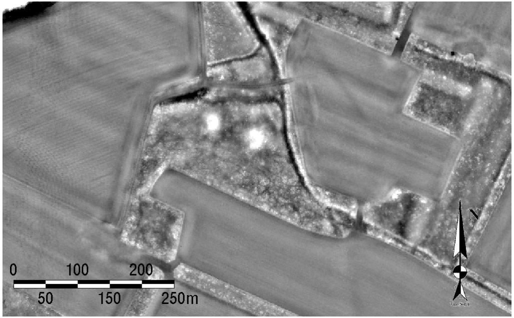
LiDAR

LiDAR is a method for accurately measuring distances. LiDAR is used to make high resolution depth maps, and has seen uses in geography, seismology, autonomous vehicles...
LiDAR works by emitting short laser pulses and measuring the time of flight between its emission and the time at which it his an object.
Here, LiDAR is used to create Digital Elevations Models for uses in Geographical Information System.



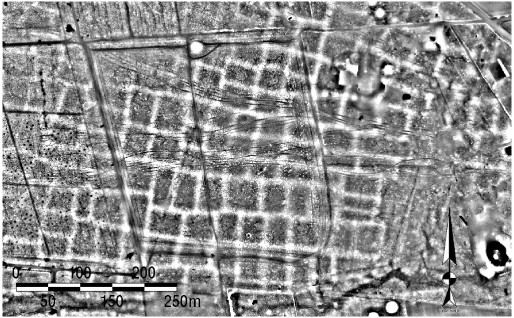
Archaeological Targets: Barrows

Barrows are mounds of raised earth or stone, over a burial placed in the middle - they are funerary monuments. They can be found all around Europe, and date from 2200BC to 1100BC. They are usually around 10 meters in diameters.



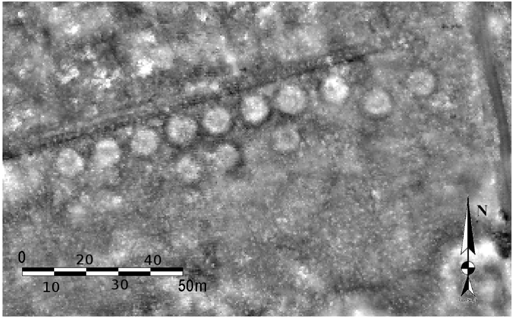
Archaeological Targets: Celtic Fields

Celtic Fields are traces of ancient agricultural field systems. They are often found in North Western Europe. They are divided in rectangular cells of about 30×30m. **In our annotation system, each cell is an Celtic Field object, not the entire field**



Archaeological Targets: Charcoal Kilns

Charcoal Kilns are circular structure used for coal production. They can be found all around Europe, and mostly around forests. Each charcoal kiln is about 5-10 meter in diameter, and are often found in groups.



Precision, recall and F1-Score

- Precision is the percentage of correct predictions (True Positives) over all the predictions (True Positives and False Positives)
- Recall is the percentage of correct precision over all correct elements (True Positive and False Negatives)
- The F_1 Score is the harmonic mean of Precision and Recall

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

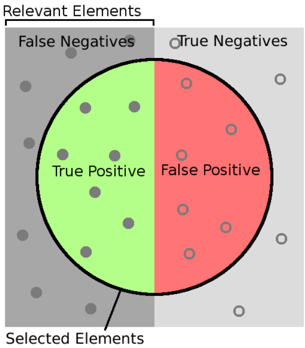
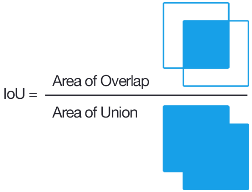


Figure: Precision and Recall

IoU(s)

- Intersection over Union (IoU) is a measure of the quality of a bounding box
- It is computed by dividing the intersection of the predicted bBox and the ground truth bBox by the union of those two bBox
- There is ongoing research toward a better IoU: see GIoU, CIoU, DIoU...



$$IoU = \frac{\text{area}(BBox_{truth} \cap BBox_{pred})}{\text{area}(BBox_{truth} \cup BBox_{pred})} \tag{2}$$

mAP

The mean Average Precision (mAP) is a popular metric for object detector as is computed as follows:

- Draw a list of all the predictions made by the model, and rank them by predicted confidence level
- Compute the area under this curve: $AP = \int_0^1 p(r)dr$, with $p(r)$ being the precision of the model given a recall value
- the mAP50 is simply the AP average over all classes for IoU that are over 0.5

Challenges

- Occlusion is a common issue in object detection
- In Aerial Remote Sensing, Vegetation might also pose a problem, with trees covering the ground
- Large Earthworks, such as Barrows, might not be recognized as of Archaeological importance, and be partially destroyed to make way for new constructions

- Remote Sensing Surveys are **large**, both in resolution and in amount of terrain covered
- This means that downscaling, usually done by detection frameworks destroys precious information !
- In Archaeology, this also means that manual verification, by way of prospection or otherwise is very difficult as it involves hundreds if not thousands of objects, separated by kilometers !

State of the Art

Learning To Look At LiDAR

This paper by Verschoof-van der Vaart and Lambers[1] present both a new dataset and applies a Deep Learning based object detection framework.

- Same 3 classes as discussed previously: barrows, Celtic fields and charcoal kilns
- Dataset comprised of LiDAR surveys from central Netherlands, about 2350 km^2 covered

- The model is a tweaked version of Faster-RCNN[2]
- Two different backbones were experimented with: ResNet50[3] and VGG16[4]
- Encouraging results were obtained, but charcoal kilns were not detected

Combining Deep Learning and Location-Based Ranking for Large-Scale Archaeological Prospection of LiDAR Data from The Netherlands

This paper by Lambers, Verschoof-van der Vaart and Bourgeois[5] improves on previous results by combining an object detector and domain knowledge

- Same surveys as before
- WODAN 2.0 now also contains negatives examples

- The model is a tweaked version of Faster-RCNN[2]
- Domain Knowledge in the form of Location Based Ranking was used
- This version obtained even better performance than WODAN 1.0 and was able to detect charcoal kilns

Automated Detections in Satellite Imagery

A lot of work has been done in the past years to improve detections of objects in satellite imagery. Since they share many of the challenges we face in automated LiDAR detection, their solutions might apply to our case:

- You Only Look Twice, Van Etten[6]
- A Simple and Efficient Network for small Target Detection, Ju et al.[7]
- Object Detection in Remote Sensing Images Based on Improved Bounding Box Regression and Multilevel Features Fusion, Qian *et al.* [8]
- A Single Shot Framework with Multi-Scale Feature Fusion for Geospatial Object Detection[9]

A (much more) comprehensive review is available as its own document.

General Principle

YOLO[10] is one of the fastest object detector on the market, and is widely considered as the SoTA for object detections. Its basic operating principle is as follows:

- Does one pass on the image $\rightarrow$ You Only Look Once
- Divide the input image into $S \times S$ grid, and predicts B bounding boxes and confidence score for each cell
- Predicts an $S \times S \times (B \times 5 + C)$ tensor
- Essentially rephrase the detection and classification problem into a regression one
- Blazing fast and accurate!

Latest Improvements: Architecture

YOLOv4[11] tried a wide range of architecture modules:

- Attention Modules (SAM)[12]
- Receptive Fields Improvements (SPP)[13]
- Feature Fusion: Cross-stage partial Connections, multi-input weighted residual connections
- Path Aggregation (PAN)[14]
- Novel Losses: Mish[15] and Swish[16]
- **Novel IoUs**: DIoU[17], CIoU[17], GIoU[18]...
- And much more !

Latest Improvements: Training

YOLOv4 also heavily experimented with training regimes enhancements, the most notable being:

- Self Adversarial Training
- **Mosaic Data Augmentation**
- Cosine Annealing Scheduler
- Cross Mini-Batch Normalization
- And more...

Implementation

Dataset details

- The original dataset from WODAN was constructed using LiDAR surveys from the Veluwe Region, Netherlands
- This data came in the form of a dozen TIFs, along with annotation in GIS form
- For each object in the dataset, we centered an segment of a fixed size, with an added jitter of $\pm$ 100 pixels
- Bounding boxes are encoded in the COCO format: $(x, y, width, height)$
- All in all, the datasets contained about 7500 images. Some images contained multiples examples of objects, some none.

Versions and annotations

- The dataset went through multiple versions, testing how resolution and negative examples might affect accuracy
  - V1: 1000×1000 pixels image, reuse of objects
  - V2: 1000×1000 pixels image, no reuse of objects $\rightarrow$ loss of performance
  - V3: 500×500 pixels image, reuse of objects
  - V4: 1000×1000 pixels image, reuse of objects, addition of 1000 image in random coordinates and Negative Examples
- V4: version gave the best results

Model Architecture and Training

Our model is based upon YOLOv4, and we experimented with the following modules:
Architectural:

- IoUs: GIoU, DIoU, CIoU
- Non Maximum Supressions: DNMS, NMS
- Different non linear units: Mish, Swish, SeLU, ReLU...
- Receptive Fields : Dilated Convolutions Modules, SPP...
- Various Input Pixel Size
- Optimized *vs* unoptimized anchors

Training:

- Mosaic Data Augmentation
- Different datasets
- Different losses
- Learning rate tweaks

The best Model uses the following:

- Distance Non Maximum Supression
- GIoU
- CV2 Data Augmentation
- Mosaic Data Augmentation
- 516×516 pixels input size

Results

Per Class Accuracy

	Kilns	Barrows	Celtic Field
Faster-RCNN[1]	—	55%	58%
WODAN 2.0[5]	12%	73%	66%
Vanilla YOLOv4	63.04%	77.04%	85.73%
Our Model	**82.37%**	**80.50%**	**95.92%**

General Performance Metrics

	Precision	Recall	F1-Score	Av. IOU	map@50
Faster-RCNN[1]	—	—	0.63	—	—
WODAN 2.0[5]	—	—	0.37	—	—
Vanilla YOLOv4	0.58	0.84	0.69	45.75%	75.27%
Our Model	**0.64**	**0.93**	**0.76**	**57.48%**	**86.27%**

Confusion Matrix

A confusion matrix shows which classes are being confused for which other classes. A perfect model would mean a diagonal matrix.

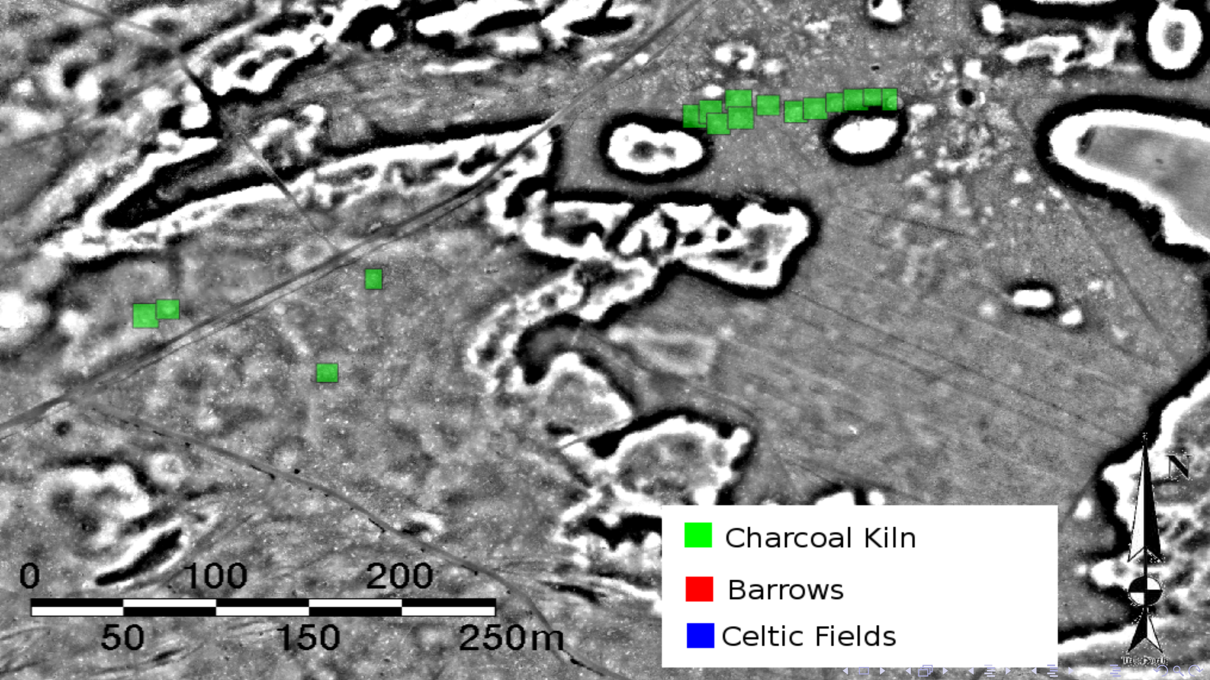
		Predictions		
		Kiln	Celtic Field	Barrow
Truth	Kiln	0.93	0.01	0.05
	Celtic Field	0.01	0.95	0.06
	Barrow	0.02	0.19	0.73

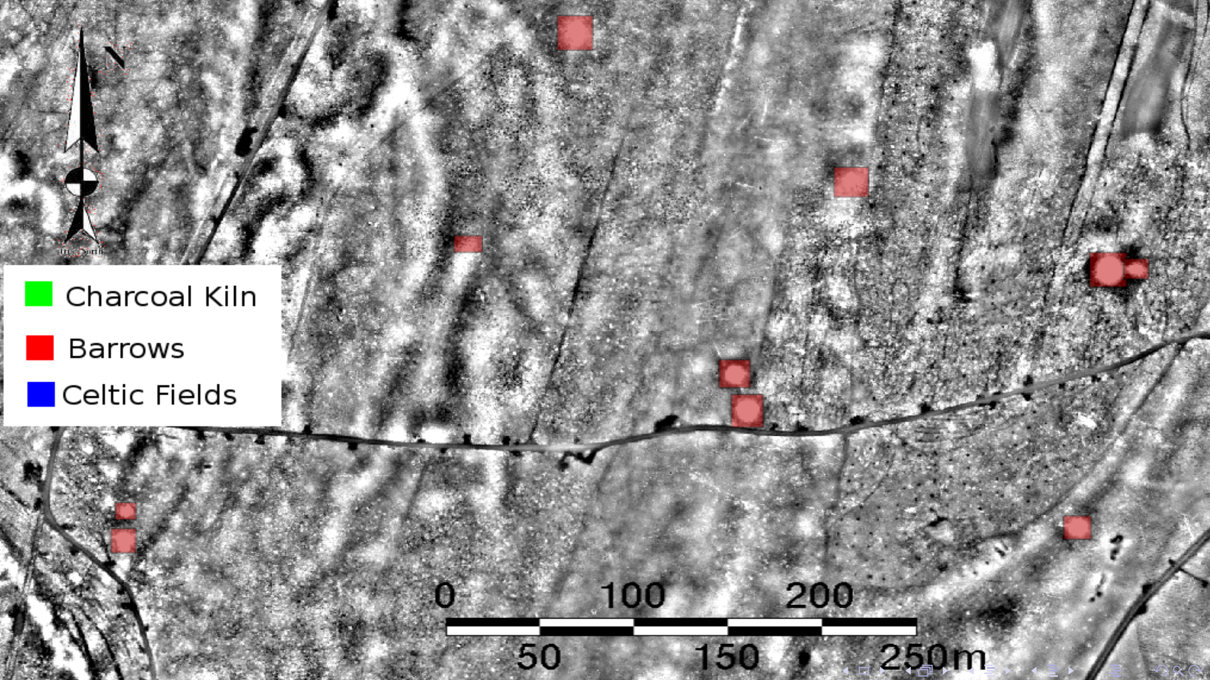
Analysis

- An optimised YOLO significantly outperforms a Vanilla YOLOv4 on this task
  - Tweaking and experimenting is worth it
- Slight class confusion between Barrow and Celtic Fields
  - Might be an indicator of low amount of examples
- Our model also outperform the State of the Art
- My personal PoV: using the latest object detection frameworks and tweaking it is the key to best performance, rather than domain knowledge
  - But using both would be even better !

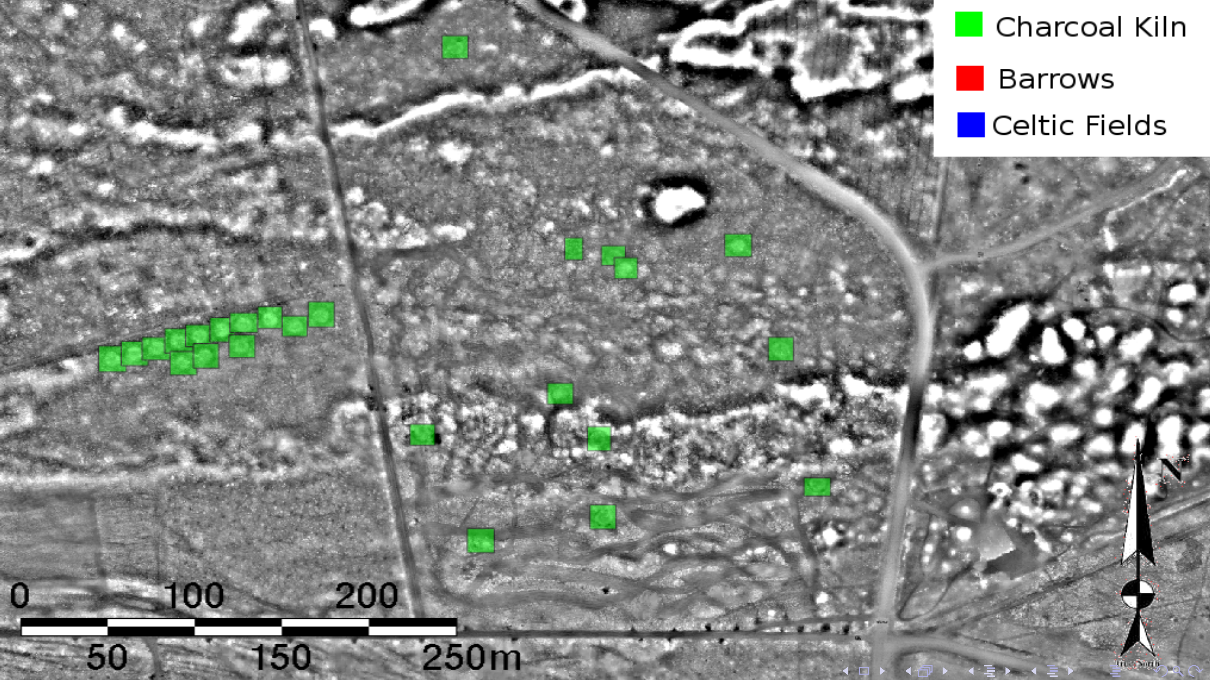
Integration

- We are also able to integrate the detection results into a Geographical Information Software for easier analysis
- We simply collect all detection results on all the images and their coordinates and confidence
- After a transformation from image coordinates to a specific coordinates reference system, the detection are stored in a CSV database
- Import to QGIS[19] is then done in a few clicks
- Post detection analysis is way easier using dedicated software !


Charcoal Kiln
Barrows
Celtic Fields
0
100
200
50
150
250m
N



N
Charcoal Kiln
Barrows
Celtic Fields
0
100
200
50
150
250m


Charcoal Kiln
Barrows
Celtic Fields
0
100
200
50
150
250m
N

Conclusions

Discussion

- Our system is accurate on all classes, and fast ! About 5 minutes for all 2350 km^2
- A modified and finetuned model is able to outperform models with added domain information
- Occlusion and confusion is still an issue $\rightarrow$

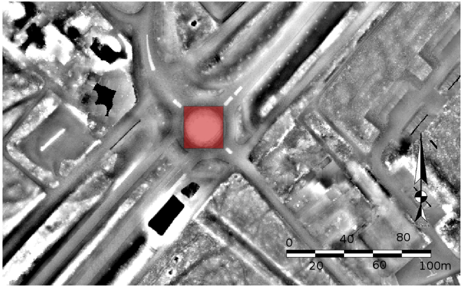


Figure: Confusion between a barrow and a roundabount

Further Improvements and what's next

- Semantic Segmentation with a SoTA model like Fast-FCN or Gated-SCNN
- Multi Data Inference: Use of multiple type of input data, like a combination of satellite imagery and LiDAR survey
- Integration of domain knowledge, as in WODAN2.0[5]
- Redaction of an article to present those results !

Question Time !

Thank you for your attention
Any questions ?

Wouter Verschoof-van der Vaart and Karsten Lambers. "Learning to Look at LiDAR: The Use of R-CNN in the Automated Detection of Archaeological Objects in LiDAR Data from the Netherlands". In: 2 (Mar. 2019), pp. 31–40. DOI: 10.5334/jcaa.32.

S. Ren et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *arXiv e-prints* (June 2015). arXiv: 1506.01497 [cs.CV].

Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *CoRR* abs/1409.1556 (2014). URL: http://arxiv.org/abs/1409.1556.

Wouter Verschoof-van der Vaart et al. "Combining Deep Learning and Location-Based Ranking for Large-Scale Archaeological Prospection of LiDAR Data from The Netherlands". In: *ISPRS International Journal of Geo-Information* 9 (May 2020), p. 293. DOI: 10.3390/ijgi9050293.

Adam Van Etten. "You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery". In: *CoRR* abs/1805.09512 (2018). arXiv: 1805.09512. URL: http://arxiv.org/abs/1805.09512.

M. Ju et al. "A Simple and Efficient Network for Small Target Detection". In: *IEEE Access* 7 (2019), pp. 85771–85781.

- Xiaoliang Qian et al. "Object Detection in Remote Sensing Images Based on Improved Bounding Box Regression and Multi-Level Features Fusion". In: *Remote Sensing* 12 (Jan. 2020), p. 143. DOI: 10.3390/rs12010143.
- Shuo Zhuang et al. "A Single Shot Framework with Multi-Scale Feature Fusion for Geospatial Object Detection". In: *Remote Sensing* 11 (Mar. 2019). DOI: 10.3390/rs11050594.
- Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640.
- Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: *arXiv e-prints*, arXiv:2004.10934 (Apr. 2020), arXiv:2004.10934. arXiv: 2004.10934 [cs.CV].

Sanghyun Woo et al. "CBAM: Convolutional Block Attention Module". In: *CoRR* abs/1807.06521 (2018). arXiv: 1807.06521. URL: http://arxiv.org/abs/1807.06521.

Kaiming He et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *CoRR* abs/1406.4729 (2014). arXiv: 1406.4729. URL: http://arxiv.org/abs/1406.4729.

Shu Liu et al. "Path Aggregation Network for Instance Segmentation". In: *CoRR* abs/1803.01534 (2018). arXiv: 1803.01534. URL: http://arxiv.org/abs/1803.01534.

Diganta Misra. "Mish: A Self Regularized Non-Monotonic Neural Activation Function". In: *arXiv e-prints*, arXiv:1908.08681 (Aug. 2019), arXiv:1908.08681. arXiv: 1908.08681 [cs.LG].

- Prajit Ramachandran, Barret Zoph, and Quoc V. Le. "Searching for Activation Functions". In: *CoRR* abs/1710.05941 (2017). arXiv: 1710.05941. URL: http://arxiv.org/abs/1710.05941.
- Zhaohui Zheng et al. "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression". In: *arXiv e-prints*, arXiv:1911.08287 (Nov. 2019), arXiv:1911.08287. arXiv: 1911.08287 [cs.CV].
- Seyed Hamid Rezatofighi et al. "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression". In: *CoRR* abs/1902.09630 (2019). arXiv: 1902.09630. URL: http://arxiv.org/abs/1902.09630.
- QGIS Development Team. *QGIS Geographic Information System*. Open Source Geospatial Foundation. 2009. URL: http://qgis.osgeo.org.