World Academy of Science, Engineering and Technology
International Journal of Transport and Vehicle Engineering
Vol:15, No:9, 2021

The Role of Synthetic Data in Aerial Object Detection

Ava Dodd, Jonathan Adams

***Abstract*—**The purpose of this study is to explore the characteristics of developing a machine learning application using synthetic data. The study is structured to develop the application for the purpose of deploying the computer vision model. The findings discuss the realities of attempting to develop a computer vision model for practical purpose, and detail the processes, tools and techniques that were used to meet accuracy requirements. The research reveals that synthetic data represent another variable that can be adjusted to improve the performance of a computer vision model. Further, a suite of tools and tuning recommendations are provided.

***Keywords*—**Computer vision, machine learning, synthetic data, YOLOv4.

I. Introduction

THE purpose of this study is to explore the question of whether machine learning can be successfully undertaken using only a machine learning computer and media that is readily available. This research attempts to understand the challenges associated with high failure rates [11] for companies seeking to implement machine learning. Some research suggests that many solutions found in the current literature are developed in simulated environments, making the results "not close enough to what is expected in real-life applications" [15]. Further, the study sought to understand whether synthetic imagery has the potential to substitute for authentic images and mitigate pervasive human labeling errors noted in research literature [5]. The project goal is to employ available resources to train YOLO4 to detect sharks, as a means to improve human-shark interactions in recreational settings.

The completed pipeline included a detailed production specification for 3D artwork, tools for managing the datasets, and potential strategies to optimize the performance of the model were investigated [19]. The synthetic images used in this study are sourced from publicly available 3D models, which are processed to crate dataset of 2D labeled images, accompanied by box locations and labels for the target objects. The completed Blender/Python API was used to augment image qualities such as respective camera positions, visual, and lighting effects. After the computer vision model was trained, a series of tests were conducted to evaluate the performance of the model, strategies were explored to investigate whether the systems' accuracy could be refined.

A. Data for Computer Vision

The volume of machine learning research has increased over the past five years, which has led to an increased availability of software, theories, and many resources for advancing machine learning systems are readily available [1]. Large public datasets such as such as Imagenet (image-net.org) or COCO (cocodataset.org/), have been used widely in computer vision research, but these resources do not include marine imagery that is specific to the project goals.

The Nature Conservancy and a collection of international partners have assembled a marine dataset for machine learning applications to enhance the management of fisheries. While a majority of the marine footage is captured from an aerial platform, the purpose of the database and the purpose of the project do not match. This mismatch is a problem of topical diversity within the dataset, where many examples represent one subset of activities (shark fishing) but offer few examples of other subsets of activities (sharks in recreational settings).

There are many instances where data sources do not include a sufficient number of images, with a clearly defined subject, in a variety of poses. Other factors that have limited the availability of quality imagery are privacy, the amount of time to collect data, and the cost of producing a labeled dataset [22]. Similar shortages are reported in a range of computer vision research, including work with thermal-infrared visual tracking [2] and pose estimation [3]. In more complex applications, the sheer volume of imagery needed makes machine learning nearly impossible. For example, computer vision researchers have noted that training a self-driving car to an acceptable level of performance might require the data generated by 100 cars driving 365 days per year, 24 hours, every day for 12 years to capture enough data [4].

B. Synthetic Data

Synthetic imagery is generated by either encoding sample images using computation methods or by using animated, 3D models to generate 2D images of the subject to be identified. Examples of computation encoding methods are Generative Adversarial Networks (GANs) and Autoencoders. Such methods encode and process a collection of images and decode the data with features changes, a system that is capable of generating unique photo-realistic images based on the original samples. A GAN consists of two collaborative networks: a generator that composes an image, and a discriminator that evaluates the quality of the synthetic images [6]. These two networks cyclically create-and-review images, which enhances the performance of the generator until it consistently configures

Jonathan Adams is with the School of Information, Florida State University, Tallahassee, FL., 32306 USA (corresponding author, e-mail: jladams@fsu.edu).

photorealistic synthetic images [7]. Auto encoders use a single neural network to encode or compress data into a latent space that describes the image. New features are introduced by altering weights and bias in decoding [21].

Blender is an open-source, three-dimensional modeling software with a Python application programming interface 3D animation is often used in a synthetic image pipeline [24]. A pipeline that employs this strategy starts with a human artist who creates a static model or a short, animated scene [8]. The artistic representations are manipulated using a Python script that controls rendering by using Blender's Python (bpy) interface. The processing script adjusts lighting, adds effects and positions the camera dynamically in order to quickly produce images that include a variety of scenes. The techniques are similar to what has been accomplished by other researchers using Blender [23] but do not appear to be reproduced otherwise with the Unreal game engine tools that can generate virtual worlds based entirely on synthetic imagery [9].

The pipeline developed for generating synthetic imagery has a number of related benefits. For example, the processing script can include functions that automatically capture the bounding box of the 3D model and apply an annotation. Error-free boxing and labeling by itself presents an enormous advancement in machine learning. Synthetic imagery solves other problems that plague computer vision training as well, including the ability to circumvent privacy regulations, and reduce dependency on large commercial datasets. Research has also demonstrated that using synthetic images to support training improves accuracy by as much as 10% [1], [10].

Fig. 1 Example of a synthetic image

Data management is a labor-intensive task associated with supervised machine learning. Because hundreds of thousands of images may be needed for training, data collection and management alone may take weeks or months. Once the data collection has been completed, images are segregated into sub-datasets. Then each unlabeled image must be examined and 'captured' with a bounding box and annotated with a label. This step in the process is time-consuming and is prone to human error. For example, in a recent study conducted at MIT, researchers discovered that about 6% of images contained in visual datasets are mislabeled or boxed improperly [5].

Data preparation activities lead to the accumulation of high project costs and high failure rates that have been noted to be as high as 51% for large enterprises and 74% for small enterprises [11]. Synthetic data production reduces cost, mitigates risk, dramatically reduces data management, and eliminates error in each of the data management tasks because data preparation tasks can be automated. Synthetic data offer a manner by which to circumvent issues associated with preparing data for training, thereby reducing time, effort, and risk.

C. Unmanned Aerial Search and Rescue

The advantages of using unmanned aerial vehicles (UAVs) as a platform for computer vision have been recognized as having great potential in marine search and rescue (SAR). Oftentimes marine searches encompass areas as large as 16,000 square miles [13]. It is understandable that human fatigue plays an important mitigating factor in such situations, making it difficult to find people or crafts lost in the ocean. The use of manned aircraft is costly and oftentimes are not readily available. Remote unmanned aircraft also offer safety advantages in such operations, by reducing the number of people in the field. Most important, UAVs have the potential to deploy in emergencies in a shorter period of time [12].

Experiments have demonstrated effective use of aerial cameras to identify and track objects in real time [13], during daylight and night operations using thermal cameras [14]. In recent times there has been an increase in demonstration projects that use artificial intelligence. It has been noted that many recent experiments are designed in simulated environments, and as such are somewhat limited in accounting for aircraft dynamics, or environmental conditions that are likely to affect performance in real-time applications [15].

D. Purpose of Study

This study evaluates a synthetic dataset production pipeline and reports subsequent performance testing using YOLOv4. The intention is to encompass an end-to-end process of manipulating 3D artwork, managing the training and validation dataset, and evaluating the performance of YOLOv4 when training is enhanced with synthetic data. A number of related research studies have demonstrated an increase in accuracy when training is enhanced with synthetic imagery [1]. The present research, then, focuses on the implications of establishing a synthetic data production pipeline to train a computer vision model.

The synthetic images are imported, manipulated and saved using a Python algorithm developed for the study. Two species of sharks (Great White and Hammer Head) were chosen as subjects for training. The project focuses on producing images to train YOLO to detect sharks from an aerial platform. While deployment of the algorithm is not within the scope of the study, such a system is intended to provide rescue workers the ability to avoid or manage close encounters between sharks and people.

II. Procedures

While the research is focused on synthetic data, authentic

images are still needed to train a computer vision model. The combination of authentic and synthetic imagery provides indistinguishable samples for the computer to learn from and strengthens predictions. Videos that featured sharks, captured from drones and helicopters, were collected from public sources. Individual frames were exported from the video at 6 frame intervals. Using this technique, 550 authentic images of Hammerhead sharks and 650 images of Great White Sharks were organized. From this pool images were set aside for testing and validation. The training and testing images were then individually labeled.

The synthetic dataset was created by rendering animated 3D models. Sharks breathe by moving through the water to keep oxygen circulating through their gills. To capture this movement with a synthetic model, an armature is added to the 3D object. An armature is composed of bones and acts as a skeleton for a model. The bones are paired with the skin of the object, called the mesh, to synchronize their movements and produce a moving model. Each time the model is set in a new position, a keyframe is set. A keyframe is a time marker that saves a place in an animation cycle where the user would like to save the objects' position. The animation that results from stringing the keyframes together makes the shark model appear to swim as the body and torso swing side to side. Lines of code were added to randomly select one of these keyframes for each render to add variation in training data.

A production specification was designed to provide a Blender artist with instructions to adjust the quality of the imagery. The processing script controlled the playback head, camera coordinates, object location, object texture, object position and lighting. The production specification detailed qualities necessary for error-free processing when the 3D model is processed. For example, the 3D object should be centered, and needs to have a base color with an added texture overlay that is bright enough to distinguish the texture from the background. The animation did not have to loop seamlessly as long as it captured the full range of motion. These specifications are optimized for training.

$$
\begin{gathered}
Set\ N_{count}\ =\ 0. \\
Set\ a\ =\ 4\pi r\text{^}2/N\ and\ d\ =\ \sqrt{(a.)} \\
Set\ M\_\vartheta\ =\ round[\pi/d]. \\
Set\ d\_\vartheta\ =\ \pi/M\_\vartheta\ and\ d\varphi\ =\ a/d\_(\vartheta.) \\
For\ each\ m\ in\ 0{:}\ (M\_\vartheta-1)\ do\{ \\
Set\ \vartheta\ =\ \pi(m+0.5)/M\_\vartheta \\
Set\ M\varphi\ =\ round[2\pi sin(\vartheta/d\varphi)] \\
For\ each\ n\ in\ 0{:}\ (M\varphi-1)\ do\{ \\
Set\ \varphi\ =\ 2\pi n/M\_\varphi \\
x\ =rsin\vartheta cos\varphi \\
y\ =rsin\vartheta sin\varphi \\
z\ =rcos\vartheta \\
N\_count+=\ 1
\end{gathered}
$$

Fig. 2 Algorithm for generating equidistributed points on the surface of a sphere with regular placement.

The processing script included several requirements, but none was more important than the camera positions. Camera perspectives are crucial for creating realistic, varied data for training an object detection algorithm. With the 3D model positioned at Blender's origin (0,0,0), the algorithm for generating equidistributed points on the surface of a sphere with regular placement was created to position the camera, to ensure adequate coverage of the object [17].

The camera orientation was adjusted to mimic video frames captured from an aerial platform. Camera positions ranged from 0 to 2*π, with φ set to generate downward viewing images. Further, implementing a regular method to place the camera at different angles around the shark allowed each angle to be utilized equally. This strengthened the synthetic dataset by introducing more images that include a diverse set of poses. The complete rendering process was neatly packed in the leopardi library [20].

Once the 3D model has been rendered into a 2D image, a truth background is selected randomly from a separate directory holding several such images, and merged with the image.

Each training session in YOLO requires the training images with their corresponding annotation files, a class text file, a data file, a training file, a testing file, a configuration file, and a convolutional weight file. The class text file lists the target domains in order, so the predictions are labeled with the correct name. For this experiment, the class file only contains the shark label class. The training and validation text files each contain a list of image pathways to be fed through the neural network (NN). The configuration file sets up the NN parameters, such as the batches, subdivisions, image resizing, learning rate, and learning rate decay. The data file contains the pathway to the class, training, and test files as well as the backup directory where training weights are stored after every 100 iterations. This file is called along with the convolutional weight file to run darknet's training command.

Two YOLOv4 models were trained with the same configuration; however, one model was trained on 10000 images and the other was trained on 3000 images. This was done to evaluate whether the *quality of characteristics* contained within the imagery can improve training. In both cases, the synthetic and authentic data were mixed together, in a ratio of 10:1, for training to optimize the model's results, which is consistent with research findings [1].

Training iterations were set to match the number of images in the training dataset [16]. After training, the final weights are stored in a backup directory. These weights are used to test the model on video frames unknown to the model.

The validation set proved crucial to the mean average precision (mAP) calculation set to occur at every epoch. The last 10% of the training data proved not to be varied enough to use as a benchmark, so a script was written using the Numpy library's random sampling function. 1,000 authentic images were randomly sampled from the training pool and used as a new validation set. This change increased testing accuracy by approximately five percent.

Training and tests were performed on a 16 core Intel i7, with 256 GB of RAM, and an NVIDIA 2080 GPU with 8GB of RAM.

III. Results

The model's best weights, the ones resulting in the highest mAP, were used for testing on shark videos. Two models were trained with the same configuration; however, one model was trained on 10000 images and the other was trained on 3000 images. The learning rate was .001, the learning rate decay was .0005, the batch size was 24, and the image input size was 608x608 pixels.

The mosaic feature and batch normalization were defaulted on the YOLOv4 configuration file as well. The adversarial learning rate feature was added with a value of 0.05. A validation set of 1000 randomly sampled authentic images were used to calculate the mAP value at each epoch. In this study, the model trained on 3000 images performed better; it had a higher mAP value and reduced the number of false positive detections.

Figures 2 and 3 illustrate the training loss (blue line) and validation mAP (red line), where fig. 3 resulted from training the model on 10000 images. The loss steadily decreased; however, the mAP oscillated throughout training. Although the mAP was 97% at the end of training, the model made numerous false positive detections during the testing phase. The loss value was not a reliable way to measure the accuracy of the model, rather how well the bounding boxes fit the object, whether it was the right object or not.

Fig. 4 shows the model's loss and mAP after training the model on 3000 images. The loss value decreased more gradually when compared to Fig. 3, and the training finished with a higher loss value. The mAP values are much more stable throughout training and much higher on average, finishing training with a mAP value of 99%. There were fewer labeling errors during the testing phase with this model.

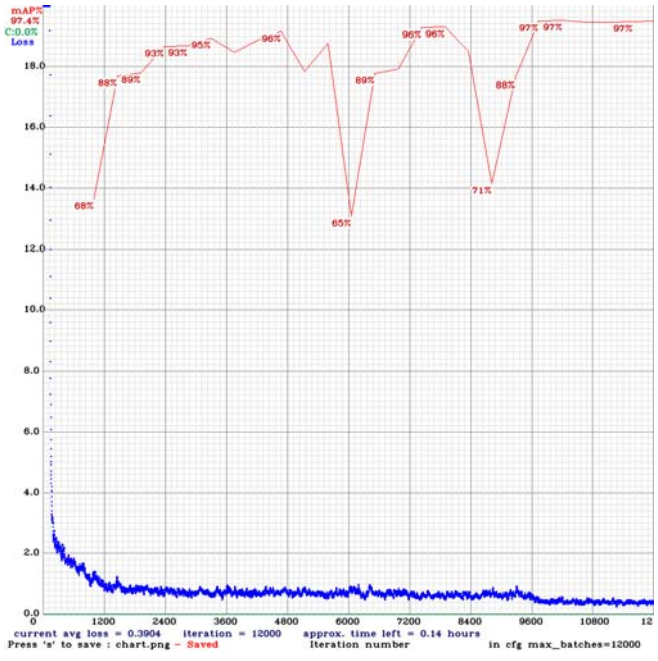

Fig. 3 Training loss and validation mAP after trained on 10000 images

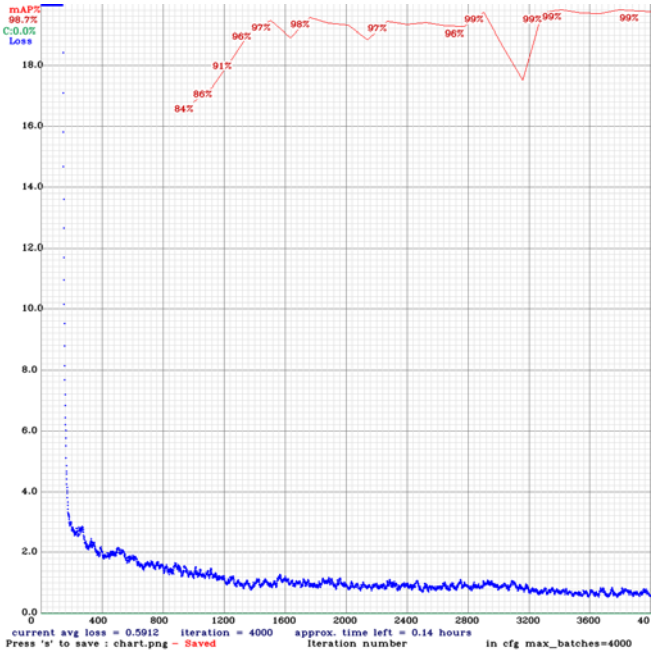

Fig. 4 YOLOv4 training loss and validation mAP after trained on 3000 images

IV. Discussion

Synthetic data play an important role in machine learning. During the course of this study, finding the "right kind" of authentic images was problematic. The synthetic imagery served to enrich the authentic data with a wide variety of poses, scale and truth backgrounds. Having the ability to produce images suitable for machine learning also offers researchers to manipulate the qualities of the data being presented to the model, and by doing so, improve the performance of the model.

At the outset, our strategy was to generate thousands of images to introduce to YOLO. However, we discovered that in reality, when training a computer vision model, more is not better. This changed the amount of time required to generate images and training substantially. Because fewer images are needed for training, training time decreased from 20 hours to 6, a substantial improvement. We have observed that presenting images that features, textures and other characteristics are enlarged appears to enhance the accuracy of the model.

To evaluate whether further performance enhancements might be achieved, YOLO's batch size, the learning rate, and the learning rate decay were evaluated. YOLOv4's Bag of Freebies was also utilized to evaluate the adversarial learning rate, label smoothing, mosaic, and DropBlock features. The results of these tests did not coincide with current research [18]. Current research literature suggests that there is also no definitive way to choose or optimize the model's hyperparameters, as the tools have been noted as being context-sensitive, suggesting the necessity to be aware of how these hyperparameters may affect performance. This study focused on how the model's inputs affect training [19].

In the beginning of this study, it was assumed more images would result in a more accurate model due to previous studies suggesting more data would fix overfitting. However, the

model appeared to overfit to the 10000-image dataset by the end of training. To remedy this, a smaller dataset composed of 3000 images was used to train the model. This resulted in a higher, more stable mAP and a model that made significantly less mistakes during testing. Therefore, the quality of the images is more important than the size of the dataset. By using synthetic data, a dataset composed of "good" data can be assembled in mere hours. The model also required less training time since there were less images to look through. Training time decreased by 30%.

V. Conclusion

This study focused on creating an end-to-end production pipeline that uses synthetic data in order to speed production without sacrificing the performance of the model. The inclusion of synthetic data was key to filling in the dataset collected for the current research, thereby providing a greater variety of poses.

The 3D renders required several production cycles and tests before the media was working well with the algorithm. Once the algorithm was configured and optimized using a single 3D model, a production specification outlined how to produce the textures, lighting, and animation features for the remaining 3D models. This strategy allowed the researchers to evaluate how well the synthetic media performed based on their specific qualities. For example, it appears that synthetic images perform best when scaled up to fill a larger percentage of the image canvas. It appears that the magnification of edges and features has an impact on training, making it possible to reach acceptable performance standards with fewer images, and subsequently less time.

A set of production tools was developed that was necessary to structure and manage the training, testing, and validation datasets. It is often necessary to track different collections of images that comprise a training dataset. For example, naming conventions can be used to identify different images characteristics such as how images were sourced, quality markers, or domains. A tool is authored to rename batches of files and their accompanying annotation files. A function to translate YOLO annotations to VOC format was also developed as some datasets are managed by one labeling system or the other. Finally, another tool critical to managing datasets is a script that will segregate percentages of images into subsets for training and validation.

Finally, the researchers investigated tuning hyperparameters in order to enhance the model's performance. It should be noted that the most consistent finding in the research literature is that consistent, reliable methods to tune hyperparameters tend to be case-specific. Evaluating hyperparameter adjustments requires time-consuming training cycles and stepwise evaluation. While such adjustments are considered key to optimizing an algorithms performance, most are considered too advanced to be practical [24]. These specifications provide data with clearly defined features and edges with specific use cases in mind. For this study, the camera angle was a crucial variable because there were a limited number of shark images taken from an aerial perspective. Using synthetic data also provides more opportunity for variety in a shorter amount of time than collecting real data. Synthetic imagery fills in the gaps where authentic imagery is insufficient in a variety of poses. Capturing images of animals is much more difficult than inanimate objects because there is more variety in body positions, and they are able to interact with their environment.

This study focused on optimizing input data to minimize overfitting and build confidence. The input data determine how "good" a model will be when tested in the field. A model must be able to generalize well, so it does not underperform beyond lab testing. Synthetic data are the solution to the lack of available, clean data for machine learning.

VI. Limitations

This study is limited by many factors. There are a limited amount of aerial shark images and videos. Many of the authentic images that are available are too small, where many examples of visible sharks underneath the surface of the water were 16x16 pixels or smaller in size. The researchers used public video to source training samples, therefore, many of the authentic images extracted from video feeds had similar qualities (i.e., background, textures, color). This has the benefit of extracting the full range of motion of the sharks but has the disadvantage of the training and validation sets containing video frames that look very similar.

Acknowledgment

The authors graciously thank Arcvale for providing expertise on three-dimensional rendering and specifications. We also thank Florida State University for providing laboratory space for conducting this research.

References

[1] J. Adams, E. Muphy, J. Sutor, A. Dodd. "Assessing the qualities of synthetic visual data production." Proceedings of the ICIET 2019: 7th International Conference on Information and Education Technology Okayama, Japan March 27-29, 2021, Association for Computing Machinery New York NY United States, 2019.

[2] L. Zhang, A. Gonzalez-Garcia, V. J. Weijer, M. Danelljan, M., & F. S. Khan, "Synthetic data generation for end-to-end thermal infrared tracking." *IEEE Transactions on Image Processing,* vol. 28 no. 4, pp. 1837-1850. 2019.

[3] A. R. Khadka, M. Oghaz, W. Matta, M. Cosentino, P. Remagnino, V. Argyriou, "Learning how to analyse crowd behaviour using synthetic data." in *CASA '19: Proc. of the 32nd Int. Conf. on Comp. Animation and Social Agents,* Paris, France, July 2019, p 11-14

[4] N. Kalra, S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" in *Transportation Research Part A: Policy and Practice,* vol. 94, RAND Corporation Santa Monica, CA. 2016, pp. 182-193,

[5] C. G. Northcutt, A. Athalye, J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks." Preprint in review. Retrieved June 1, 2021 from https://arxiv.org/pdf/2103.14749.pdf.

[6] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, X. He, "AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA June 2018. pp. 1316-1324.

[7] Google Developers (2019, October 08). Overview of GAN structure | Generative Adversarial Networks. Retrieved Jan. 10, 2021, from https://developers.google.com/machine-learning/gan/gan_structure

[8] K. Lee, D. Moloney, "Evaluation of synthetic data for deep learning stereo depth algorithms on embedded platforms." *4th International Conference*

on Systems and Informatics (ICSAI), Hangzhou, China, Nov. 2017.

[9] W. Qiu, F. Zhong, Y. Zhang, S. Qiao, Z. Xiao, T. S. Kim, Y. Wang, "UnrealCV: Virtual worlds for computer vision." In: Hua G., Jégou H. (eds) *Computer Vision – ECCV 2016 Workshops*. European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 9915.pp. 909–916.

[10] P. S. Rajpura, M. Goyal, H. Bojinov, R. S. Hegde, "Dataset Augmentation with synthetic images improves semantic segmentation." In Rameshan R., Arora C., Dutta Roy S. (eds) *Computer Vision, Pattern Recognition, Image Processing, and Graphics.* NCVPRIPG 2017. Communications in Computer and Information Science, vol 841. Springer, Singapore.

[11] A. Komarraju, "Unfortunately, commercial AI is failing. Here's why." In *Analytics Insight: Artificial Intelligence Latest News,* Feb. 13, 2021. Retrieved June 5 2021, from https://www.analyticsinsight.net/unfortunately-commercial-ai-is-failing-heres-why/

[12] S. Yeong, L. King, S. Dol, "A review on marine search and rescue operations using unmanned aerial vehicles." *International Journal of Marine and Environmental Sciences*, vol 9, no 2, pp 396 – 399, 2015.

[13] P. Nousi, I. Mademlis, I. Karakostas, A. Tefas and I. Pitas, "Embedded UAV real-time visual object detection and tracking," *2019 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 708-713, 2019.

[14] C. Burke, P. R. McWhirter, J. Veitch-Michaelis, O. McAree, H. Pointon, S. Wich, S. Longmore, "Requirements and limitations of thermal drones for effective search and rescue in marine and coastal areas." *Drones*. Vol. 3 no. 4, 2019.

[15] F. A. de Alcantara Andrade, A. Reinier Hovenburg, L. Netto de Lima, "Autonomous unmanned aerial vehicles in search and rescue missions using real-time cooperative model predictive control." *Sensors.* 21 no. 4, Sep 20, 2019.

[16] A. B. Alexey, "Darknet." Unpublished. Retrieved on June 12, 2021 from https://github.com/AlexeyAB/darknet/blob/master/README.md.

[17] M. Deserno, How to generate equidistributed points on the surface of a sphere. Unpublished. Sept. 28 2004, Retrieved on Jan. 18, 2021 from https://www.cmu.edu/biolphys/deserno/pdf/sphere_equi.pdf.

[18] F. Hutter, H. Hoos, K. Leyton-Brown, "An efficient approach for assessing hyperparameter importance." In ICML, pages 754–762, 2014.

[19] Yu, Tong, and Hong Zhu. 2020. "Hyper-parameter optimization: A review of algorithms and applications." http://search.ebscohost.com.proxy.lib.fsu.edu/login.aspx?direct=true&db=edsarx&AN=edsarx.2003.05689&site=eds-live&scope=site.

[20] Sutor, J. (2021, May 25). *johnsutor/leopardi*. GitHub. https://github.com/johnsutor/leopardi.

[21] P. Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures." in *Proceedings of Machine Learning Research.* Proceedings of ICML Workshop on Unsupervised and Transfer Learning, vol. 27 pp. 37-49, 2012.

[22] K. Mason, S. Vejdan, S. Grijalva, "An 'On the Fly' Framework for Efficiently Generating Synthetic Big Data Sets" *IEEE International Conference on Big Data (Big Data),* pp. 379-338, 2019.

[23] S. Reitmann, L. Neumann, B. Jung, "BLAINDER-A Blender AI Add-On for Generation of Semantically Labeled Depth-Sensing Data." *Sensors*. 21, no. 6, Mar 18, 2021.

[24] L. Li, "Why Does No One Use Advanced Hyperparameter Tuning?" *Determined AI.* Oct. 08, 2020, Retrieved from https://www.determined.ai/blog/why-does-no-one-use-advanced-hp-tuning