REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
DECEMBER 2022	TECHNICAL PAPER	JULY 2022	OCTOBER 2022

4. TITLE AND SUBTITLE

3DLIVE TECHINQUE ANALYSIS: A study of segmentation, classification and object detection of 3D point cloud datasets

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
	N/A	
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)

Damain Moquin, Claire Thorpe, Caleb Williams, Casey Schwartz, Dakota Turk, Ariana Emad

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory - RIED, RIEA
26 Electronic Pkwy
Rome, NY 13441

8.PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10.SPONSOR/MONITOR'S ACRONYM(S)	11.SPONSOR/MONITOR'SREPORT NUMBER(S)
Air Force Research Laboratory/RIED, RIEA 525 Brooks Road Rome NY 13441-4505	AFRL/RI	AFRL-RI-RS-TP-2022-022

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. PA # AFRL-2022-5542
Date Cleared: 16 Nov 2022

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The aim of this research is to discuss the current state-of-the-art practices and methods for machine learning algorithms that perform on point cloud data. The research conducted will be applied to the in-house efforts of the Three Dimensional Lidar Visualization and Exploitation (3DLIVE) team, whose primary goal is to create a new system for visualization and interaction with point cloud data for Target Coordinate Mensuration (TCM). The proposed machine learning methods relate to three main topics in machine learning for 3D point clouds and computer vision, each of which had its own segment of papers researched. These topics are segmentation, classification and object detection, and the selected papers are of recent studies that achieved state-of-the-art performances. The findings of this research are a select few methods that show the most promising results and effectiveness to the 3DLIVE team. Effectiveness is largely dependent on the scalability and applicability of the algorithm to the 3DLIVE use case as well as its accuracy and precision.

15. SUBJECT TERMS

LiDAR, point cloud, 3D, machine learning, deep learning, classification, segmentation, object detection

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
U	U	U	SAR	49

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
DAMAIN MOQUIN	**N/A**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

The aim of this research is to discuss the current state-of-the-art practices and methods for machine learning algorithms that perform on point cloud data. The research conducted will be applied to the in-house efforts of the Three Dimensional LiDAR Visualization and Exploitation (3DLIVE) team, whose primary goal is to create a new system for visualization and interaction with point cloud data for Target Coordinate Mensuration (TCM). The proposed machine learning methods relate to three main topics in machine learning for 3D point clouds and computer vision, each of which had its own segment of papers researched. These topics are segmentation, classification and object detection, and the selected papers are of recent studies that achieved state-of-the-art performances. The findings of this research are a select few methods that show the most promising results and effectiveness to the 3DLIVE team. Effectiveness is largely dependent on the scalability and applicability of the algorithm to the 3DLIVE use case as well as its accuracy and precision.

2.0 INTRODUCTION

In traditional computer vision problems, 2D data has been the dominant form of information used for reasoning. With the recent development of affordable and widely available 3D sensors (such as the Apple depth camera, Kinect, and time of flight cameras) 3D data has become abundant and offers many advantages for solving computer vision problems. Namely, it contains more topological information (depth dimension, shape and scale information) that can be crucial for scene understanding and provides a more natural representation of the world. Due to this technical facet, applying 3D data to the areas of autonomous driving, robotics, remote sensing, and medical treatment has been the focus of recent research and will continue to expand to other domains [1].

3D data can come in many formats including meshes, depth images, volumetric grids, and point clouds. The most common format for scene understanding applications is point cloud-structured-data because this data form preserves the original geometric information in 3D space without any discretization loss. Before proceeding with analysis, a point cloud needs to be defined: a point cloud is a set of data points (x,y,z) that typically represent the external surface of a 3D object(s) and is produced either synthetically or by 3D scanners. One challenge that 3D data faces is storage requirements—a 3D scene takes several orders of magnitude more storage than the same scene in 2D. Point clouds solve this issue because it does not require the storage of a polygonal-mesh, therefore improving performance and lowering overhead—key considerations for time sensitive applications [2].

The Three-Dimensional LiDAR Visualization and Exploitation (3DLIVE) project aims to create a new system for Target Coordinate Mensuration (TCM) and 3D analysis. The current approach for TCM uses stereoscopic imagery, utilizing NVIDIA 3D Vision Glasses as well as specialized GPUs and monitors to view overlaid 2D images, giving the perception of three-dimensionality. However, this approach is difficult to train for and it can cause eye-strain; in addition, the NVIDIA software and hardware it utilizes has reached its End-of-Life (EoL) and is no longer being supported or produced. Thus, a new solution for 3D data exploitation needs to be developed.

The 3DLIVE approach for TCM aims to utilize 3D Point Clouds primarily collected by LiDAR sensors. This data is then visualized using the game engine, Unity. Furthermore, because said data can be efficiently loaded into Unity via the Octree format, massive-scale datasets can be used. Metadata information for the points can be viewed and analyzed within the viewer, and the user can navigate throughout large geographical regions and select points to analyze. There are multiple ways to interact with the data, from dropping a point of interest at a location to measuring distances, lengths, and areas.

There are multiple modalities to interact with 3D Point Cloud data. They include a standard mouse and keyboard, virtual reality, and augmented reality (using the Hololens 2). The augmented reality interaction is the primary development focus for the 3DLIVE team because it immerses the user within the data while still being similar to the stereoscopic glasses approach. We also are currently seeking to use Machine Learning (ML) to allow us to obtain information about objects within these point cloud datasets automatically, such as what object they are and what boundaries they have within the space, and perform automatic target recognition (ATR).

In the past 10-20 years, most deep learning computer vision research has focused on 2D images, but with the rise of more available 3D data, recent research has looked at applying traditional deep learning techniques to 3D data for computer vision. This new research has allowed for major advances in scene understanding scenarios, but there are still hurdles that come from transitioning models from 2D to 3D. For point clouds specifically, the data is unstructured and unordered, meaning that deep learning networks that work with point clouds as input cannot directly apply standard deep learning methods such as convolutional neural networks (CNNs) [1]. Instead, custom solutions have to be developed to be permutation invariant, typically achieved with a symmetrical function. Another challenge is capturing local and global structural information from a point cloud. Evaluating a point cloud by individual points loses the local and global structural information between points so networks have to account for this in their design by looking at neighboring data. Due to the difficulties of working with point clouds directly, many methods transform the point cloud data into an intermediate format, like projecting a point cloud into a 2D image, so that traditional deep learning methods can be applied [1]. Lastly, point cloud data collected from 3D sensors is not perfect—there is often noise contamination and outliers due to the limitations of sensors, inherent noise of the acquisition device, and the reflective nature of the surfaces being captured can disrupt data collection [1]. As can be seen above, applying deep learning methods on point cloud data is not straightforward and requires reworks of existing techniques for use in networks, but the gain in descriptive power of 3D point cloud over 2D data outweighs the negatives.

Computer vision tasks are normally split into 3 distinct categories: classification, object detection, and segmentation. For point clouds, these categories are usually defined as: 3D shape classification, 3D object detection and tracking, and 3D point cloud segmentation [1].

3D shape classification methods attempt to classify (label) objects in the point cloud by learning the embeddings of each point first and then extracting a global shape embedding from the entire point cloud using an aggregation method. This global embedding is input into several fully connected layers to achieve a classification [1].

3D object detection and tracking methods can be binned under 3 categories: 1) object detection, 2) object tracking, and 3) scene flow estimation. For object detection methods, they produce

oriented 3D bounding boxes around each detected object for an input point cloud. Next, the objective of 3D object tracking is to predict the state of an object given its previous state. Related to object tracking is 3D scene flow estimation, where given two point clouds of the same scene at two different moments in time, the movement of each point to get from the first point cloud to the second is described [1].

Like object detection and tracking, 3D point cloud segmentation can be split into 3 categories depending on the granularity desired. Those categories are, from most general to least: semantic segmentation (scene level), instance segmentation (object level), and part segmentation (part level). Given a point cloud, the goal of 3D point cloud semantic segmentation is to separate the point cloud into several subsets according to the semantic meanings of points (e.g. color all chairs in a scene the same color). At one level lower is 3D point cloud instance segmentation, which is more challenging than semantic segmentation because it requires more accurate and fine-grained reasoning of points. Instance segmentation needs to distinguish points with not only different semantic meaning but also separate instances with the same semantic meaning (e.g. color each chair a different color instead of all the same color). Finally, at the most granular level, part segmentation attempts to separate parts of objects with the same semantic meaning (e.g. color parts of the chair differently), making the task especially difficult because shape parts with the same semantic label have large geometric variation and ambiguity [1].

One of the goals of the 3DLIVE effort is to create a system (utilizing Machine Learning) that intakes a point cloud of a geographical area, groups points with similar properties into objects, and labels each constituent object and structure to make the data easier to work with and analyze. Before we proceeded towards these goals, we determined it would be valuable to research the current state of the art for segmentation and classification on point cloud datasets. Guo et al. completed a survey of deep learning methods for point clouds in 2019 [1]. We aimed to confirm that the information presented in the study is still accurate and relevant (ML for point cloud datasets is a rapidly developing field), perform our own research and create a similar survey, and decide which of the researched methods for classification, segmentation, and object detection would be the best for our use case. The AFRL RIEA/RIED In-House Research Team (IHURT) was assembled to do this research alongside the 3DLIVE team and answer the following research question:

What is the current state of the art for 3D point cloud segmentation and classification, and which approaches will work best for the 3DLIVE effort? Can we begin to lay the framework and develop a course of action for segmentation, classification, and object detection for the large-scale 3D urban point clouds that we intend to work with?

The results of this study will allow the 3DLIVE team to move forward with ML point cloud analysis. We hope to eventually replicate the highest performing and most relevant methods for

segmentation, classification, and object detection and utilize them on geographical 3D point cloud data from NGA's Geospatial Repository and Data Management (GRID) server. In addition, the 3DLIVE team has already developed a method for generating large-scale synthetic urban point cloud datasets, and we can leverage this synthetic data as additional training data for the models that we create and use. This research will lay the groundwork for the 3DLIVE team to use ML to create additional tools to help warfighters analyze and mensurate 3D data. This will ultimately lead to the aforementioned goal of creating a new TCM system for targeteers (such as those in the 363rd ISR Wing and other targeting branches) to use, replacing the currently deprecated technology with an alternative that leverages increasingly available native 3D data.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

For the selection of each individual paper, there was a set of criteria each paper needed in order to be considered for the full review. Each paper needed to be written within the past 5 years (i.e. from 2018 to present), be peer-reviewed, and collectively no authors were repeated across papers more than twice. Since the 3DLIVE team has a goal of replicating the methods outlined in the papers, a higher priority was put on papers that also included links to code used. This strict set of criteria would allow for the team to gather a diverse set of papers.

Once the 3DLIVE team finished their assessments on the set of papers, the pruning process began shortly after. The pruning process was done not necessarily to eliminate papers, but rather to highlight and prioritize papers that met more optimal criteria. The optimal criteria were: the paper had to detail a method that would be extendable to large-scale point cloud (a point cloud of a one-to-three block radius); the paper was focused on outdoor datasets from either LiDAR or urban environment; and the paper had open source code. By those requirements, papers could be more easily eliminated. Because the 3DLIVE team is looking to implement these algorithms, having code available was of high importance, especially in this category due to the lack of urban datasets being tested on for detection processes. Therefore, for all three subcategories, the following assumptions were made during the selection process:

I. The algorithms could be implemented with large-scale urban datasets
II. The methods detailed in the papers can easily be implemented or tested for accuracy
III. The algorithms are running on medium-tier computers unless otherwise stated

Note: after the **Eliminated Articles** section for each subtopic, the curated articles will be presented in order of relevance to the 3DLIVE team goals, ranked from least relevant to most relevant.

3.1 Segmentation

3.1.1 Summary of Common Datasets

The conducted survey of current 3D point cloud segmentation applications highlighted five common segmentation datasets: ScanNet, KITTI, SemanticKITTI, S3DIS, and Semantic3D. The ScanNet dataset contains RGB-D videos of different indoor settings. The 2.5 million views and 1500 scans contained within the videos are annotated with instance-level semantic segmentations as well as surface reconstructions, and 3D camera poses. The provided annotation within this dataset made it easily accessible for experimental 3D point cloud segmentation.

KITTI is a vision benchmark dataset initially cultivated for autonomous driving research. The dataset contains high-resolution color and grayscale annotated video. SemanticKITTI builds upon the original KITTI dataset via multiple additions of different semantic annotations; such

additions include the sequences of the Odometry Benchmark; the moving and non-moving target annotations of distinct classes; and sequential scan-labeling.

S3DIS, also known as the Stanford 3D Indoor Scene dataset, contains annotated scene point cloud data for 271 different rooms. S3DIS is a popular benchmark dataset for both Semantic Segmentation, 3D Instance Segmentation and 3D Object Detection. The Semantic3D dataset contains robustly labeled 3D point clouds of various natural scenes which are annotated with object representations and labeled points.

Table 1. Summary of common segmentation datasets

Dataset (training size, testing size)	**Data Classes**	**Metadata**
ScanNet ~1500 scans → 2.5 million views Train: 1045, Test: 312, Validate: 156	Unannotated, wall, floor, chair, table, desk, bed, bookshelf, sofa, sink, bathtub, toilet, curtain, counter, door, window, shower curtain, refrigerator, picture, cabinet, other furniture (22)	Segment groups
KITTI Train: 6,347, Test: 711, Validate: 423	building, sky, road, vegetation, sidewalk, car, pedestrian, cyclist, sign/pole, fence (10)	Tracking labels, 3D bounding boxes, 3D points, 2D camera imagery, y-axis rotation, object dimensions, object location
SemanticKITTI Train: 11 sequences, Test: 11 sequences	Road, sidewalk, parking, other-ground, building, other-structure, car, truck, bicycle, motorcycle, other-vehicle, vegetation, trunk, terrain, person, bicyclist, motorcyclist, fence, pole, traffic sign, other-object(28)	Tracking labels, 3D bounding boxes, 3D points, 2D camera imagery, y-axis rotation, object dimensions, object location
S3DIS 6 indoor environment, 270 rooms	ceiling**,** floor**,** wall**,** beam**,** column**,** window**,** door**,** table**,** chair**,** sofa**,** bookcase**,** board**,** and clutter (13)	dimensions, number of points
Semantic3D Train: 15 scenes Test: 15 scenes	Man-made terrain, natural terrain, high vegetation, low vegetation, buildings, hard scape, scanning artifacts, and cars (8)	labels

3.1.2 Summary of Common Metrics

The singularly-shared metric across all of the papers for segmentation was the mean Intersection over Union (mIoU). This metric is calculated by averaging multiple iterations of IoU values, a calculation detailed in **Figure 1**.

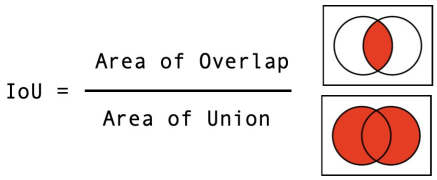

Figure 1. Intersection of Union (IoU) Calculation

The IoU is a calculation of how well the model is able segment a class from the background of the image. Essentially, the original data is labeled with bounding boxes around the desired item for segmentation; then, during segmentation, the algorithm applies its own bounding boxes to the data. The IoU is calculated by dividing the area of overlap between these two bounding boxes by the area of the union of bounding boxes. Therefore, this metric shows us how well the algorithm is able to segment the class item in alignment with the ground-truth segmentation.

3.1.3 Eliminated Papers

LatticeNet: Fast Point Cloud SegmentationUsing Permutohedral Lattices

A total of ten different segmentation articles were read for this study; however, only eight were found to be relevant for intended 3DLIVE scenarios. The first of the methods detailed in the two eliminated articles, LatticeNet, unfortunately did not meet our needs for point-cloud segmentation because the efficacy of the model was predicated on the time and memory use of operations on permutohedral lattices. While those metrics are valuable for the implementation of models on point-clouds, we are more interested in methods to classify segments of our point-clouds, rather than optimizing them [3].

Geometry Sharing Network for 3D Point Cloud Classification and Segmentation

The other network, the geometry sharing network, was found to be less than relevant to the 3D LIVE use case on the basis of the data utilized in the study [4]. The Geometry Sharing Network focussed on an improved representation of objects in a small scale indoor space. When comparing this study to other studies in the pool which were already successful on more desirable data, it was determined that it was more efficient to opt out of an in depth review of the Geometry Sharing Network, as any attempt to replicate the work done in this study on 3DLIVE relevant data would require additional work in comparison to the other studies reviewed. Hence, the geometry sharing network was eliminated. Now, the relevant articles will be discussed.

3.1.4 Individual Tree Crown Segmentation

In their study, Chen, Jiang et. Al. perform segmentation on point clouds of trees in hopes of using it for acquiring the parameters of forest trees. The first part of this research study is developing the point clouds from LiDAR data. To develop their point cloud dataset they combine top down perspectives with horizon views. After combining these perspectives to create point clouds of all of the individual trees, the point clouds are sent through the PointCloud network for segmentation with the addition of voxelization [5]. This method allowed researchers to perform the necessary segmentation with enough accuracy to gather specific parameters for individual trees in their LiDAR data.

Although the intended purpose of this study seems unrelated to the 3DLIVE effort, this study is still useful since it demonstrates how to collect and prepare LiDAR data for successful results with PointNet. If the data preparation process in this study can be replicated with the LiDAR data the 3DLIVE team is interested in, the team could then hypothetically achieve optimal results using PointNet, which may be easier to use than some of the other methods described in this section.

3.1.5 RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation

In their research Xu, Zhang et. Al. explain that there are three different mediums of point cloud representation; point-based sets, voxel-based cells or range-based images (i.e., panoramic view). Each of these mediums has their own benefit to the point cloud segmentation process which is what inspired the researchers to pursue the potential of combining all three of these representations before feeding them to a Point Cloud segmentation network [6]. All three mediums are generated via multi-view interactive learning and then the informative data from each medium is exchanged to perform segmentation, as shown in **Figure 2**.

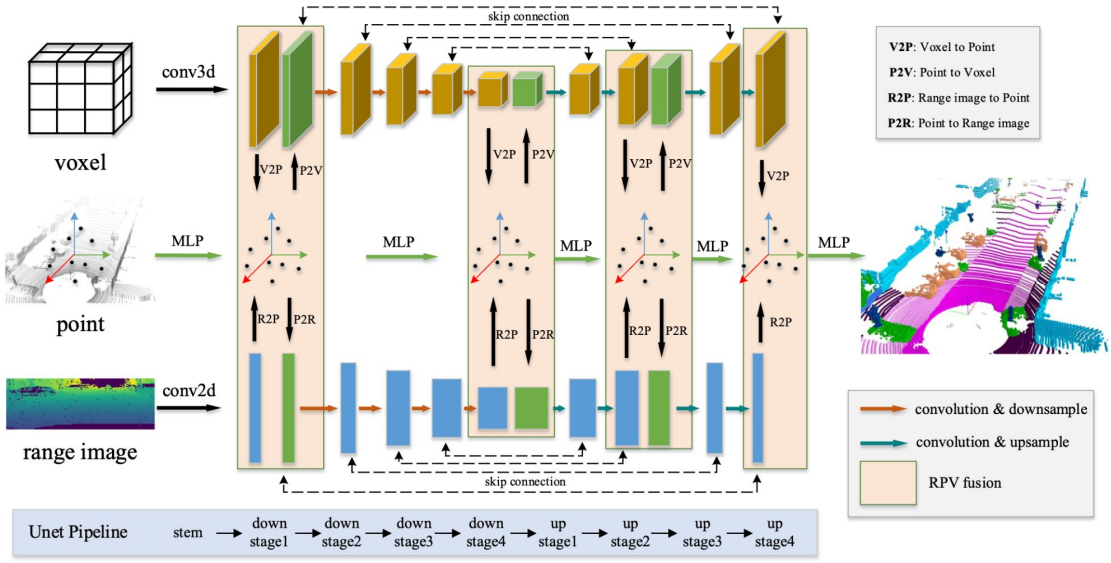

Figure 2. Overview of RPVNet. It is a three-branch network with multiple interactions among them, where voxel- and range-branch share the similar Unet architecture, and point-branch only utilize per-point MLPs. Originally shown in [6].

RPVNet was evaluated on the SemanticKITTI and nuScenes datasets via mean Intersection of Union (mIoU) metric. Both SemanticKITTI and nuScenes contain point clouds of outdoor environments, making them very relevant to 3DLIVE potential use cases. For both SemanticKITTI and nuScenes, RPVNet achieved a mIoU metric higher than its competitors and acquired either the highest score, or a score within 5 percentage points of the highest, for each of the segmentation classes [6].

RPVNet's success on outdoor scenic data directly connects to a potential 3DLIVE use case. As the 3DLIVE team has been interested in performing 3D point cloud segmentation on 3D point clouds that can span multiple miles of terrain. RPVNet has proven to be successful on this the same variety of data that the 3DLIVE team is interested in. However, since the RPVNetwork utilizes voxelization it may be less efficient to apply and deploy this method in comparison to the other methods reviewed in this research effort.

3.1.6 SCF-NET and SCF Module Implementation

The first component of the SCF implementation is to modify and obtain additional information from 3D Point Clouds. The three block module modifies the spatial representation of the Point Cloud so it does not change when rotated on the z-axis, obtaining more discriminative features as well as a global context for each 3D point in the point cloud. The three block SCF module is embedded into SCF-Network [7]. The SCF Network performs segmentation and identities

classes within 3D point clouds by performing multiple levels of encoding and decoding before using nearest neighbors to construct the 3D point cloud feature map and finally predicted semantic labels for the classes. The integration of the SCF module is what differentiates it from pre-existing segmentation models such as PointNet.

When SCF Net is evaluated on S3DIS and Semantic3D the network achieves a mean Intersection of Union (mIoU) percentage of 71.6, approximately 24 percentage points higher than PointNet which only achieves a mIoU percentage of 47.6 for S3DIS and a mIoU of 77.6 on Semantic3D [7]. In terms of semantic label accuracy, SCF Network accuracy is on par with the other benchmark networks, being consistently within at most 10 percentage points from the highest value among the other networks.

The success of SCF Net on the Semantic3D dataset demonstrates its potential usefulness because it achieves better results than PointNet and utilizes outdoor, urban data that is similar to data the 3DLIVE team would use in application. The code for SCF Net is also open source and available for download off of GitHub, which makes it even easier to attempt to recreate these results. However, these results were not as impressive as other articles that were reviewed, so it still ranks lower overall.

3.1.7 Point Cloud Translation

Xiao and Huang et. Al. deploy a new method they identify as point cloud translation in order to perform point cloud segmentation capabilities. The first part of this method is to develop a synthetic LiDAR point cloud dataset that is similar to their available real LiDAR dataset. Then, once this dataset is created, a study is performed to identify any key differences or gaps between the synthetic data and real data. Once these differences are identified, the researchers deploy a PCT translator so they can utilize their synthetic data to supplement their real data point clouds. The translation of the synthetic data to real data is especially important because it is significantly more efficient to label the synthetic data and these labels can be transferred to the real data it can yield much larger and labeled real point cloud datasets. For this study, SemanticKITTI and SemanticPOSS were utilized as the real datasets [8].

Once the synthetic data is transferred onto the real data, the now improved real data is fed through a segmentation network known as MinkowskiNet [8]. The mean Intersection of Union (mIoU) achieved by the transfer learning dataset surpassed pre-existing data preparation methods such as SynLiDAR + SematicPOSS/SemanticKITTI (S+T), Maximum Classifier Discrepancy (MME), Maximizing for Domain Invariance (MMD), and Attract, Perturb, and Explore (APE). All of these datasets were fed through the same segmentation network for their results.
The success of the point cloud translation method on LiDAR data is especially relevant to the 3DLIVE use case because the 3DLIVE team has access to relevant synthetic LiDAR data.

Therefore, the opportunity to replicate this method and utilize this synthetic data to yield successful results on real LiDAR data is especially interesting. This experimental method also suggests that any state-of-the-art network could be used because pre-existing models minimize both time and cost.

3.1.8 Fuzzy Spherical Kernel Convolution

Fuzzy spherical kernel convolution is a promising method for large-scale analysis of complex point clouds, with Lei, Akhtar, and Mian's[9] convolution algorithms being able to segment over a million points per second with great accuracy. Fuzzy spherical kernel will achieve these results by applying a spherical convolution to the point cloud by separating depth-wise and pointwise operations proceeding Xception application. Most impressively, the neural net used, SegGCN, was evaluated on S3DIS and ScanNet, both robust and complex semantic segmentation datasets.

The method requires some preprocessing, however: as is common for utilizing segmentation models that can produce valuable results, some form of sub-cloud iteration is necessary. In this model's case, a closed sphere of neighboring points is generated for each point, with range search used for segmentation over k-nearest-neighbors because range search is not only resilient in dealing with brute-force requisite situations, but also provides consistent metric information that is "robust to density changes" in the point clouds. It must be noted that fuzzy kernels have a decided advantage over traditional hard kernels, which suffer from "ambiguity and sharp changes at bin boundaries".

Now, output features will be calculated via the learnable parameters defined by the fuzzy kernel (the nullspace of the open-set of points sharing some relationship associated with a membership function). And, based on our new partitioning schema, the convolution presented becomes a combination of parameters that ameliorate the sharp changes between bin boundaries.

Also, the spherical kernel is far more efficient and efficacious; while the spherical kernel used might require three additional multiplications and two additional additions to the hard kernel, said computations pale in comparison to the computations necessitated by a dense vector setup. Plus, these computational qualities of the spherical kernel are achieved by limiting the density of the coefficient vectors, further ascribing quickness to applying said algorithm.

In addition, the fuzzy kernel is resilient against adverse boundary conditions if weighted assignments are removed, as well as it being robust against missing data, the perhaps most prevalent issue facing point-cloud analytics. It must be stated that the spherical convolution has been evaluated on a battery of metrics, including OA, mAcc, IoU, and mIoU, so there should be no doubt that the method's results are aberrational. Below, in **Table 1**, these results are presented, and while said improvements are potentially shallow, they are improvements nonetheless:

Table 2. Performance of various kernels on S3DIS dataset. Notable that the fuzzy spherical kernel improves upon the performance of both SPH and KPConv evaluation networks.

kernel	hard SPH	fuzzy SPH	hard KPConv	fuzzy KPConv
OA	88.0	**88.2**	86.9	87.1
mAcc	69.6	**70.4**	68.7	68.6
mIoU	62.9	**63.6**	60.5	61.0

Fuzzy kernel segmentation is nonetheless incredibly viable for 3DLIVE. With a readily available codebase and a metrizably-simple algorithm that exploits the intricacies of a cloud to flesh out the significant regions, kernel manipulation should be implemented as a resource for identifying important targets.

3.1.9 Contrastive Boundary Analysis

Contrastive boundary analysis holds great promise for improving segmentation results because of an intuitive sub-partitioning schema, and through its contrasting of point features to scene boundaries, feature representation is improved. Tang, Zhan, Chen, Yu, Tao's[10] method is particularly clever because it prioritizes the various classes' boundaries of the point cloud instead of the 'innards' of each relevant region.

This contratastive boundary analysis is predicated on the use of a generalized InfoNCE loss contrastive optimization goal for boundary points that would learn features from neighboring points of the same class rather than neighboring points of disparate classes. As a result, the feature discrimination across scene boundaries is refined; and, as is common among segmentation methods, partitioning the dataset using sub-clouds can yield better representations of scene boundaries, at which point contrastive boundary learning will be applied further. To combat ground-truth issues with continued sub-cloud sampling, an iterative approach wherein sub-sampled clouds require boundary points of each higher-ordered cloud before proceeding with label determination, will be used. From here, the proportion of k-classed points in each group of points generating a specific sub-cloud will be leveraged to determine and evaluate the boundary points of the original cloud.

The method's results were corroborated by comparing them to the ConvNet baseline on the S3DIS and Semantic3D datasets. Against 3SDIs, the contrastive boundary approach demonstrates perfectly that its algorithm does not trade off "between scenes of major and minor classes" (T,Z,C,Y,T) by identifying specific classes via sacrificing the accuracy of moot classes (walls, ceilings, etc). Against Semantic3D, the contrastive boundary approach notably improves on the high/low vegetation classes, both of which confuse most other models because of said class similarity and shared features. **Figure 1** highlights these results:

Table 3. ConvNet results before and after the addition of contrastive boundary algorithmic partitioning. The red values indicate ConvNet's overall accuracy and mean intersection over union differential growth.

	CBL		mIoU(%)	OA(%)
	@input	@sub-scenes		
ConvNet			69.71 -	88.97 -
	✓		70.05 +0.34	89.01 +0.04
	✓	✓	70.98 +1.27	89.31 +0.34
ConvNet (multiscale head)			69.83 +0.12	88.88 -0.09
	✓	✓	71.33 +1.62	89.40 +0.43

It should be noted that the paper's overemphasis on scene boundaries proved to dilute the model's available time in analyzing inner areas of the point cloud. As a result, exploration of the relationship between the boundary of point clouds and the inner areas of point clouds must be elaborated upon.

3DLIVE utilizing contrastive boundary analysis is not predicated on said exploration's success, however: in fact, even if a small measure of inner cloud area identity is sacrificed, the emphasis on said inner regions in most segmentation methods bolsters contrastive boundary analysis's usefulness if it were to be supplemented with other similar methods, such as the soon-to-be-described multi-path segmentation algorithm.

3.1.10 Multi-Path Segmentation

Wei, Lin, Yap, Hung, Xie[11] propose a pre-segmentation task aimed at using multi-path region mining to create point level labels and is achieved via a weakly supervised learning method using scene and sub-cloud labels. However, before their classification algorithm is applied, each class appearing in the scene must be hand-identified and said task can pose its own issues because generating the requisite training data can be cumbersome. For the time being, let's assume this task is trivial because of some large, available workforce taking care of this labor. Now, we address the form of hand-identification: for sub-cloud labels, the paper proposes a query radius of two meters by which classes are correspondingly hand identified with images of 5.5 meters x 5.1 meters x 2.4 meters. From here, the complexity of each scene likely relegates us to feeding the scene into the model via some partitioning and stacked sub-clouds; afterwhich, the classification occurs via the use of the KPConv network and multiple ResNet blocks on the ScanNet dataset. Then, a Momentum SGB optimizer will be incorporated.

Most importantly are the results: for sub-cloud classification, it must be said that traditional classifiers using only scene-level labels can have a difficult time learning the more discriminative and identifying features of the scene because sampled regions will contain walls/floors that heavily bias the results. While the model does not completely deal with the issue, this sub-cloud sampling allows for regions to contain the smaller and discriminative features while 'minimizing' the necessity of wall/floor forms.

The multi-path module is their proprietary technology, though, so its results are far more important; the empirical data gleaned from the mIOU metric suggests that the MPRM improves, by a large margin, the segmentation performance using "both scene-level and subcloud-level labels". As a result, 3DLIVE should integrate these multi-path modules; the approach is straightforward, the preprocessing does not pose any issue for the vast hierarchical structure of the Air Force, and a codebase is available for use and development.

3.1.11 SSPC-Net Segmentation

SSPC-Net is a powerful tool for classification of point clouds by way of biasing the learning method toward more discriminative and contextual features of segmented regions. Cheng, Hui, Xie, and Yang[12] suggest that they can achieve these results by taking a semi-supervised segmentation network and introducing a superpoint graph-embedding module, then using a dynamic label propagation network bolstered by "superpoint dropouts and coupled attention mechanism learning".

However, preprocessing must be done before SSPC-Net can be implemented: superpoints must be found and their features gleaned. To accomplish this, the paper suggests the use of unsupervised partitioning to generate the superpoints which will then generate, via neural net, superpoint graphs for feature embedding. From here, superpoint label propagation must be done: this propagation can be implemented by adopting a "dynamic propagation strategy that generates pseudo-labels. Now, a coupled attention mechanism will learn discriminative contextual features by weighing all extended superpoints; the exact method by which the extraction of new contextual features is by randomly choosing a superpoint via modular multiplication. Finally, training of the SSPC-Net is done using an ADAM optimizer and the following datasets: S3DIS, ScanNet, and vKITTI.

Simply put, their model achieves greater performance than the standard semi-supervised point cloud segmentation methods and it does so with fewer labels. This claim is corroborated by several standard metrics in **Table 3**: mean IoU (mIou), mean class accuracy (mAcc), and overall accuracy (OA):

Table 4. SSPC-Net evaluation on the ScanNet and vKITTI datasets

Method		Rate	ScanNet		vKITTI		
			mIoU	OA	mIoU	mAcc	OA
Full	PointNet	100%	-	73.9	34.4	47.0	79.7
	PointNet++	100%	-	**84.5**	-	-	-
	SSP + SPG	100%	-	-	**52.0**	**67.3**	84.3
	G+RCU	100%	-	-	35.6	57.6	79.7
	RSNet	100%	**39.3**	79.2	-	-	-
	3P-RNN	100%	-	-	41.6	54.1	**87.8**
	3DCNN	100%	-	73.0	-	-	-
Semi-	Baseline	0.01%	24.1	38.2	35.7	53.4	79.2
	SSPC-Net	0.01%	27.1	66.6	41.0	55.7	81.2
	SSPC-Net	0.05%	**39.3**	**77.1**	**50.6**	**64.8**	**85.4**

However, a drawback must be noted: if few labeled points are available or not correctly identified, there could be a large gap between the data distributions of semi-supervision versus full supervision. That fault is outweighed largely when looking at the number of extended superpoints: the percentage of extended superpoints is larger when fewer supervised points are available, thus demonstrating the value of pseudo-labels when impressively few point-annotations are present.

As far as 3DLIVE is concerned, a notable asset of SSPC-Net must be addressed: sub-cloud segmentation can harness the need to emphasize microcosms over macrocosms. Utilizing inner-cloud-regions to flesh out larger regions has its place in identifying large-scale effects (as in the fuzzy kernel segmentation method), but these iterative sub-cloud partitions can conversely identify the minutiae of a scene. This property is incredibly useful when assessing battle damage to a region; it would be remiss to not mention that fuzzy kernel segmentation and SSPC-Net could be used in tandem to produce a more holistic assessment, but SSPC-Net could independently evaluate exceptionally on its own.

3.1.12 Segmentation Findings

Segmentation algorithms should be further investigated by the 3DLIVE team because they have an exceptional ability in discriminating between the vaguely-defined classes (i.e. walls, floors) and the valuable and scrutiny-worth classes that truly identify regions of a point cloud. Said segmentation can accomplish this identification via the incredibly useful tactic of sub-cloud iteration, an implementation that is not only cost-effective but also computationally simple. Fuzzy kernel convolution and contrastive boundary analysis, in particular, have had the most success in exploiting said tactic, exemplified by the results shown in **Table 1** and **Table 2**.

3.2 Classification

3.2.1 Summary of Common Datasets

The most common datasets for 3D point cloud classification include the Vaihingen dataset of the International Society for Photogrammetry and Remote Sensing (ISPRS) dataset, the GML(B) 3D dataset, and 2019 IEEE GRSS Data Fusion Contest 3D point cloud classification challenge (DFC 3D dataset).

Vaihingen is a subset of the dataset provided by International Society for Photogrammetry and Remote Sensing and was captured over Vaihingen, Germany. It consists of three test areas which contain various object classes. Those classes are: powerline, low vegetation, impervious surface, car, fence/hedge, roof, facade (building sides), shrub, and tree. XYZ coordinates, intensity values, number of returns and point labels are provided in the data files.

The GML(B) dataset is an ALS point cloud dataset with four semantic classes: ground, building, tree, and low vegetation. The dataset contains around 1.5M data points for training and 1.2M for testing. 3D coordinates are provided for each point in the dataset.

The DFC 3D dataset is also an ALS point cloud dataset, which is scanned from Jacksonville, Florida and Omaha, Nebraska. Six classes are predefined in the dataset, including ground, high vegetation, building, water, elevated road/bridge, and unlabeled points. 3D coordinates, intensity, and return number are provided for each point in the dataset.

Table 5. Summary of common classification datasets

Dataset (training size, testing size)	Data Classes	Metadata
Vaihingen (753,876 training points, 411,722 testing points)	Powerline, low vegetation, impervious surface, car, fence/hedge, roof, facade, shrub, tree (9)	3D Coordinates, intensity values, number of returns, labels
GML(B) (approx 1,000,000 training points, approx. 1,000,000 testing points)	Ground, building, tree, low vegetation (4)	3D Coordinates, labels
DFC 3D (approx. 111,000,000 training points, approx. 10,000,000 testing points)	Ground, high vegetation, building, water, elevated road/bridge, unlabeled (6)	3D Coordinates, intensity values, number of returns, labels

3.2.2 Summary of Common Metrics

The most common metrics used to assess classification methods come from the ISPRS contest; they are precision, recall, F1 score, and overall accuracy (OA). Overall accuracy is generally used to evaluate the overall classification accuracy for all categories, and is defined as the percentage of correctly classified points out of the total points. The F1 score is the harmonic average of precision and recall of each category and is more suitable for unevenly distributed datasets (i.e. low number of examples for one category). F1 score is defined as:

$$2 \cdot \frac{precision \cdot recall}{precision + recall}.$$

3.2.3 Eliminated Papers

An Efficient and General Framework for Aerial Point Cloud Classification in Urban Scenarios

Özdemir et al. propose a 2D CNN architecture that receives 2D patches of data and processes them like an image-based object detection network outputting class probabilities [13]. The classification framework relies on handcrafted features as well as self-learnt ones, and would therefore be ill-suited for the 3DLIVE use case that wants an end-to-end trainable model.

Multispectral LiDAR Point Cloud Classification Using SE-PointNet++

A point-wise multispectral LiDAR point cloud classification architecture termed as SE-PointNet++ is proposed via integrating a Squeeze-and-Excitation (SE) block with an improved PointNet++ semantic segmentation network [14]. The results are state-of-the-art, but require multispectral point cloud input that is difficult to acquire.

Aerial Point Cloud Classification With Deep Learning and Machine Learning Algorithms

This paper compares multiple machine learning algorithms for the task of point cloud classification [15]. Those algorithms are: One vs One classifier, BiLSTM, 1D CNN, and 2D CNN; each algorithm uses your standard architecture. Due to the simplicity of the models, they would be easy to implement for 3DLIVE but the results shown in the paper were obtained by removing and combining classes from the Vaihingen dataset, making them difficult to compare to other methods.

PointHop: An Explainable Machine Learning Method for Point Cloud Classification

The PointHop method consists of two stages: 1) local-to-global attribute building through iterative one-hop information exchange, and 2) classification and ensembles [16]. In the attribute building stage, they address the problem of unordered point cloud data using a space partitioning procedure and by developing a robust descriptor that characterizes the relationship between a point and its one-hop neighbor in a PointHop unit. The model is explainable and shows good results, but the dataset used is not similar to the ones of interest to the 3DLIVE use case in terms of size and location.

3.2.4 ALS Point Cloud Classification by Integrating an Improved Fully Convolutional Network into Transfer Learning with Multi-Scale and Multi-View Deep Features

This paper proposes an ALS point cloud classification method that integrates an improved fully convolutional network into transfer learning [17]. Specifically, first multi-scale voxel and multi-view feature maps are extracted to characterize the 3D point cloud comprehensively, and then the feature maps are fed into the pretrained DenseNet201 CNN model which results in deep features with good generalization. Then, a fully convolutional neural network is designed to

classify the ALS point cloud using these features. Finally, they utilize a graph-cuts algorithm to refine the classification results.

Their method results in an Overall Accuracy (OA) of 89.8% and an Average F1 Score of 83.6% on the International Society for Photogrammetry and Remote Sensing (ISPRS) dataset. However, the drawback to their approach lies in the fact that their procedure for feature map generation is very complex, and there are many preprocessing steps involved.

The dataset used is applicable to the 3DLIVE and AF use case as it is outdoor and relatively large scale. The objects classified (such as roofs, vegetation, and cars) would be relevant to our interests for targeting and battle damage assessment (BDA) applications. The ability to automatically classify these sorts of objects would be valuable. However, while the results are extremely good, they are probably not replicable for the 3DLIVE use case. There are multiple complex preprocessing steps (including multi-view and voxelization methods) they may be difficult to recreate, especially considering our limited resources. In addition, the authors do not provide their source code so their method would be difficult to reverse engineer. Thus this is one of our lower priority findings in terms of applicability.

3.2.5 CBF-Net: An Adaptive Context Balancing and Feature Filtering Network for Point Cloud Classification

This point cloud classification paper was written by Qian Zang, Wenhui Diao, Kaiqiang Chen, Ling Liu, Menglong Yan, and Xian Sun, and published on August 20, 2021 in the IEEE Journal of Selected Topics In Applied Earth Observations and Remote Sensing. The authors seek to create an adaptive context balancing and feature filtering network for point cloud classification, which would address the issues of distinguishing contributions of different points towards classification (especially edge points and outliers) and identifying classes sharing similar characteristics (especially in complex scenes [18]. Their network is primarily built on the PointNet++ architecture, consisting of an encoder and decoder network. In each local area, a Balanced Context Encoding (BCE) module is added to balance the contribution of each point based on its input features. Features derived from the third layer are put into a Filtered Feature Aggregation (FFA) module, which maps high-dimensional features into a low-rank subspace and therefore removes some redundant information.

This method results in an Overall Accuracy of 83.0% and an Average F1 score of 69.2% using the ISPRS and RueMonge2014 datasets. These are good results compared to previous state of the art models such as PointNet++ (OA: 81.6%, Avg F1 67.0%), and the model achieves particularly good scores for minority classes, such as Powerline. The model also has good time and memory consumption stats (they show for example that D-FCN uses almost 20 times the parameters for a similar classification performance with a slower inference time compared to their method)

compared to other methods. However, there are some points of misclassification located at the junction of certain categories like Roof and Façade. In addition, categories with similar characteristics such as Fence/Hedge and Shrubs tend to be easily misclassified.

The datasets used are applicable to the 3DLIVE and AF use case for similar reasons to the previously discussed model, and the results should be applicable to the same use cases of Targeting and BDA. The results are good, somewhat in the middle of the methods we have researched (outperforming PointNet++ and D-FCN in terms of OA and Avg F1 but with lower scores than other models such as GACNN and GADH-Net). and observed for the state of the art in 3D point cloud classification. The approach is fairly straightforward and is based on the widely known PointNet++ architecture, so it should be relatively possible for the 3DLIVE team to recreate it. However, the source code is not available, and this compared with the (relatively) middling results puts the priority of this model for 3DLIVE usability lower on the list.

3.2.6 PointNet++ Network Architecture with Individual Point Level and Global Features on Centroid for ALS Point Cloud Classification

This paper on point cloud classification authored by Yang Chen, Guanlan Liu, Yaming Xu, Pai Pan, and Yin Xing was published on January 29, 2021 in the MDPI Remote Sensing Journal. It takes the PointNet++ architecture and adds point-level and global information on centroid points in the sample layer which are added to local features at multiple scales to extract useful features [19]. A modified loss function based on the focal loss function is proposed to solve the uneven category distribution problem in 3D point cloud datasets. Additionally, an elevation and distance-based interpolation method is proposed for objects in ALS point clouds that exhibit elevation distribution discrepancies. The network follows a U-Net architecture, with 4 downsampling and upsampling levels and skip connections for each level. Point-level and global information is concatenated to the local features to capture useful information at each downsampling stage. Skip-linked features from the abstraction level, and fully connected ReLU layers are adopted to capture each point feature vector, and a fully connected layer is used on the last upsampling layer for classification.

Their method results in an Overall Accuracy of 83.2% and an Average F1 score of 71.2%. The ISPRS Valhingen dataset is used for training and testing and the GML(B) 3D dataset is used for generalization testing. These are relatively good results (once again beating previous SOTA like PointNet++ in OA/F1 metrics but being outperformed by GACNN and GADH-Net), and in terms of training time their model outperforms many existing approaches (they mention that GADH-Net takes 7 hours to train while their module only takes 2). Their proposed method also significantly reduces processing time and memory consumption. Their architecture is also relatively simple compared to other point cloud classification models. The only major drawback

is that their model does not quite reach state-of-the-art performance as compared to models like GADH-Net when tested.

The dataset used is applicable to the 3DLIVE and AF use case as it is outdoor and relatively large scale, and the objects classified (such as roofs, vegetation, and cars) would be relevant to our interests for things like targeting and BDA. Their approach is straightforward and carefully described and laid out, so it should be possible for the 3DLIVE team to recreate the steps that the authors have taken. Their code is not available, but the PointNet++ code that they base their architecture on is. Overall, this is a reasonable option for the 3DLIVE use case.

3.2.7 Airborne LiDAR point cloud classification with global-local graph attention convolution neural network

This point cloud classification paper was written by Congcong Wen, Xiang Li, Xiaojing Yao, Ling Peng, and Tianhe Chi, and published on April 20, 2020 to ArXiv. Their goal was to create a graph attention convolution neural network (GACNN) that could be directly applied to the classification of unstructured 3D point clouds collected by airborne LiDAR [20]. First they introduce a graph attention convolution model that incorporates global contextual information and local structural features. They then further design an end-to-end encoder-decoder network (GACNN) inspired by SegNet and PointNet++ which captures multiscale features of the point clouds, therefore enabling more accurate airborne point cloud classification. Raw 3D point clouds with coordinates and other optional features are directly input into the encoder network.

Their model achieves an Overall Accuracy of 83.2% and an Average F1 score of 71.5% on the ISPRS dataset. Their model achieved a new state of the art for classification performance in terms of average F1 score at the time, eventually being surpassed by other models such as GADH-Net. Their model also has good generalization capabilities as showcased by its performance on the 2019 Data Fusion Contest Dataset, where it has the highest performance on the bridge deck, building, and ground classes and the highest Avg F1 score compared to any other model at the time. Façade classification is one of the model's weak points, likely due to the mingling of the category with the roof, tree, and shrub categories. The model also has relatively poor classification performance for the fence/hedge (F1: 37.8%) and shrub (F1: 46.7%) categories, most likely due to the similar topological features and distribution of them [20]. Shrubs are also sometimes misclassified as trees, which probably results from the mixing and lack of obvious boundaries of the categories. However, the performance of the model is still quite impressive despite these weaknesses.

The dataset used is applicable to the 3DLIVE and AF use case as it is outdoor and large scale, and the objects classified would be relevant to our interests (targeting and BDA). The approach is relatively straightforward and should be recreate-able by the 3DLIVE team, especially

considering the model directly takes in the point cloud as input without any additional preprocessing steps. The results are also quite good (outperforming PointNet++ and DGCNN in terms of OA and Avg F1), especially considering this model is a few years old at this point (progress is fast in this field). Overall this would be a solid choice for the 3DLIVE team to explore for point cloud classification.

3.2.8 Directionally Constrained Fully Convolutional Neural Network for Airborne Lidar Point Cloud Classification

The authors of [21] proposed a directionally constrained fully convolutional neural network (D-FCN) that can take as input raw point cloud data (x,y,z coordinates and intensity) and classify arbitrary input point sizes. This is achieved, partly, via a novel convolution module (D-Conv) that performs convolution in an orientation-aware manner by using a directionally constrained nearest neighborhood search. The overall architecture is a multiscale network with downsampling and upsampling blocks employed in unison with D-Conv to enable multiscale point feature learning. More specifically, the network follows an encoder-decoder framework in U-Net-like fashion. It is worth noting that no additional geometry features are used as input to enhance local structure information, which many other models use.

The major innovation with this model is the D-Conv module, which allows for convolutions that capture the spatial structure of input points, unlike PointNet-like architectures that apply unordered operations to aggregate the features of all neighbor points and thus ignore order information. The D-Conv module, like the conventional convolution operator, covers a small receptive field and is responsible for local feature extraction. In addition, the D-Conv module follows the same steps for convolution: 1. A local receptive field for each point is constructed and 2. The convolution operation is applied to the feature values within the receptive field using the kernel weights. The first step is accomplished by projecting all the input 3D points onto an xy plane and then dividing the 2D point space into 8 (default) evenly divided cone subspaces. The projection from 3D to 2D is performed because point clouds from an airborne LiDAR have a higher variance in the horizontal direction (xy coordinates) than in the vertical direction (z coordinates), meaning that information is denser in the xy plane than 3D space. For example, doing a nearest neighbor search by taking a sphere of points on a building rooftop in the 3D space introduces many vacant points (above the roof) into the neighborhood that contribute little to the convolution operation. Instead, converting the point cloud onto the xy plane and taking the neighborhood of a point on the roof eliminates the empty space above the roof and maximizes the number of informative neighboring points. It's also more efficient than the 3D search. **Figure 3** shows this concept. After the projection is split into 8 subcones, K-nearest neighbor points in each area (subcone) within a radius of R are selected to form the receptive field of the central point. The z coordinate is maintained for use in future point feature encoding.

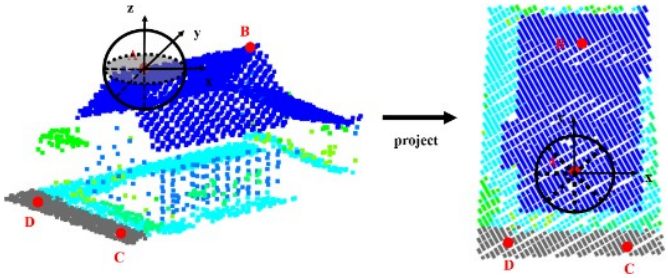

Figure 3. 3D neighborhood search (left) and projected 2D neighborhood search (right). Originally shown in [21]

The next step in the D-Conv module involves aggregating the features in each subcone into a one-dimensional vector using a 1 x K convolution with a stride of K. Each of these is a directionally-aware feature vector. Then, with these feature vectors and a central point, an orientation-aware convolution is applied to extract the point descriptors for each central point (and associated directionally-aware feature vectors). The output of this is added to the input vector using an element-wise summation operation, depicted in **Figure 4**.

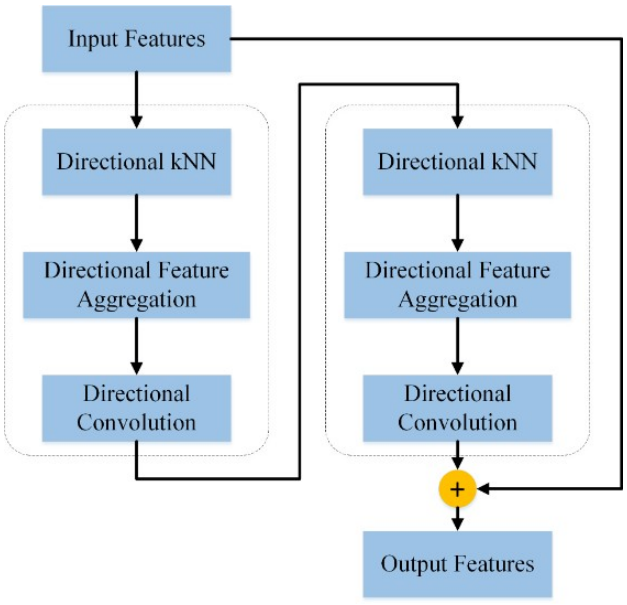

Figure 4. Depiction of the proposed D-Conv module. The dashed box indicates a convolution block and the '+' is an element-wise summation operation. Originally shown in [21]

The overall architecture, as mentioned before, follows the same pattern as U-Net, with 3 downsampling layers followed by 3 upsampling layers, with regular convolutions and D-Convs in between layers to expand and contract the number of channels as well as to embed orientation-aware information into the feature matrix. The features of the last upsampling layer are input into a fully connected layer to label each point. Skip connections are added between the downsampling and upsampling stages to boost the convergence speed and increase performance. Finally, the features from the downsampling stage are concatenated with the feature matrix of the sample point set size in the upsampling stage.

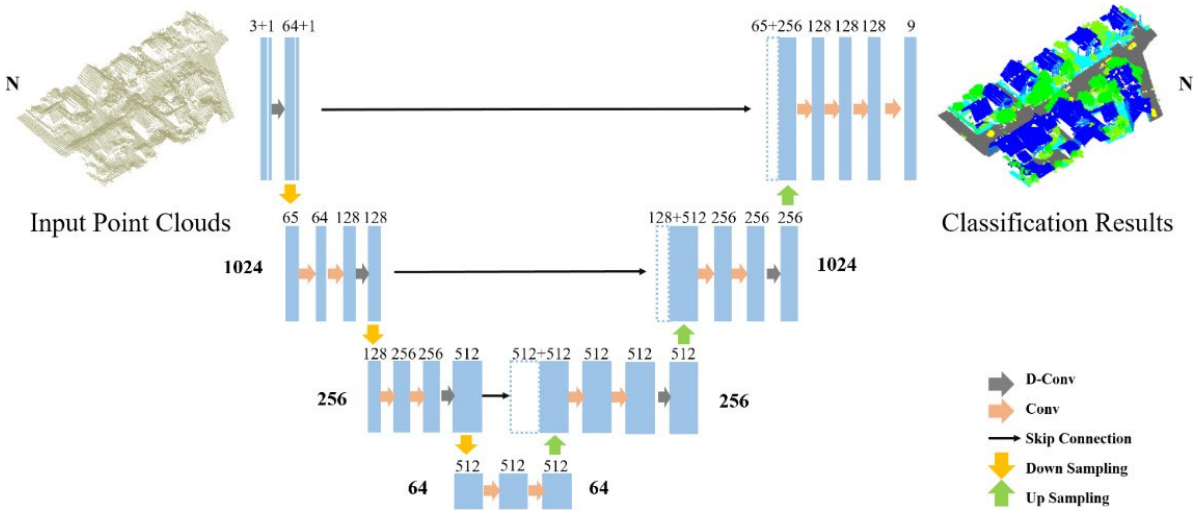

Figure 5. Illustration of the D-FCN network architecture. The network includes both a downsampling path and an upsampling path. Originally shown in [21]

Results for D-FCN are promising when compared to state-of-art models (2019) on the ISPRS dataset. In terms of average F1 score, D-FCN achieved a score of 70.7%, 0.14% higher than the previous best model [21]. The overall accuracy is less impressive, lagging behind the best model by 0.3%, although the authors contribute this to optimizing for F1 score in their hyperparameters, not OA. In practice, we want to optimize for F1 score instead of OA because OA will cause minority categories to be ignored. Compared to other models, D-FCN is exceptionally good at classifying not well represented categories, like power lines in ISPRS because of designing the model for F1 score.

In terms of the data used to train the model, it is applicable to the 3DLIVE and AF use case as it is outdoor and large scale, and the objects classified (such as roofs, vegetation, and cars) would be relevant to our interests for things like targeting and BDA. The results are near state-of-the-art, but are slightly edged out by newer models like GADH-Net. Their approach is relatively straightforward compared to other state-of-the-art models (e.g. GADH-Net) and

carefully described, so it should be easy for the 3DLIVE team to understand and adjust the model, as necessary. Their code is available, and instructions are provided on how to train and run the model. Overall, this is a good option for the 3DLIVE use case.

3.2.9 A Geometry-Attentional Network for ALS Point Cloud Classification

In the paper [22], the authors developed an end-to-end deep learning model that takes advantage of three characteristics that are present in point clouds from ALS to improve classification. Those characteristics are: (1) myriad geometric instances (2) large scale variation between classes (e.g. car vs building) (3) elevation discrepancy between objects.

To address (1), the model uses a novel geometry aware convolution that can learn discriminative geometry of neighborhoods around individual points in the point cloud. The convolution first calculates a low-level geometric feature prior for the neighborhood of each central point as a means to induce feature learning of high-level local pattern representation. This geometric feature prior is calculated using a combination of eigenvalues of covariance matrix and height information. The prior feature vector is then passed to several multi-layer perceptron layers to get the high-dimensional feature vector that now contains underlying geometric properties of the neighborhood, which is finally used as weights for the neighborhood feature vector to encode geometric information. **Figure 6** shows this process.

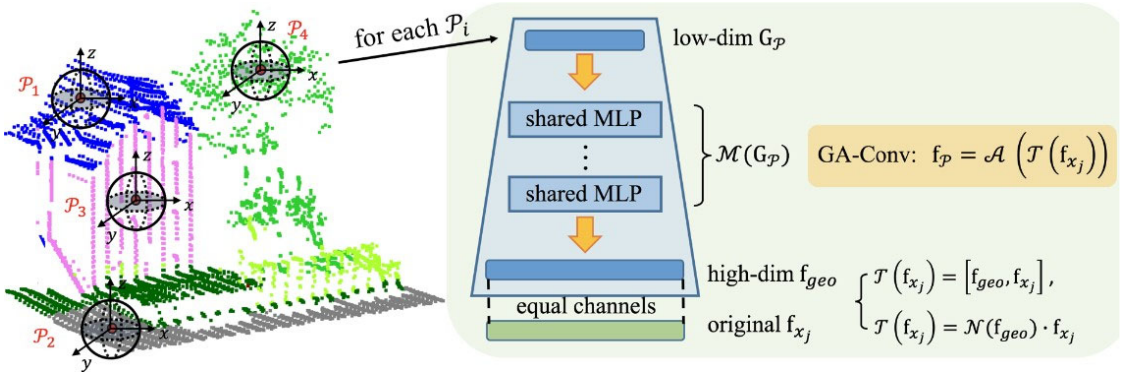

Figure 6. Illustration of the geometry-aware convolution (GA-Conv). Originally shown in [22]

For (2), the authors developed a dense hierarchical architecture that incorporates multi-level receptive field information in the classification to improve results on multiple categories at different scales. It is well known that smaller objects prefer small receptive fields, while larger objects respond better to large receptive fields for classification. Therefore, it is recommended to use all levels of receptive information to accurately classify objects at varying scales. The proposed network uses a U-Net like architecture, with multiple down sampling operations, the GA-Conv we defined above, followed by multiple up-sampling operations, in this case feature propagation, to restore the size of the original pointwise features. In addition, each

down-sampled level is fully up-sampled to the original feature vector size for use in a fully connected layer that creates a class probability map that is used for final class prediction, which is how the network considers multiple receptive fields. Densely connected skip connections are added to reduce information loss during the down-sampling process and weighted cross entropy is used to combat class imbalance issues. **Figure 7** shows the dense network.

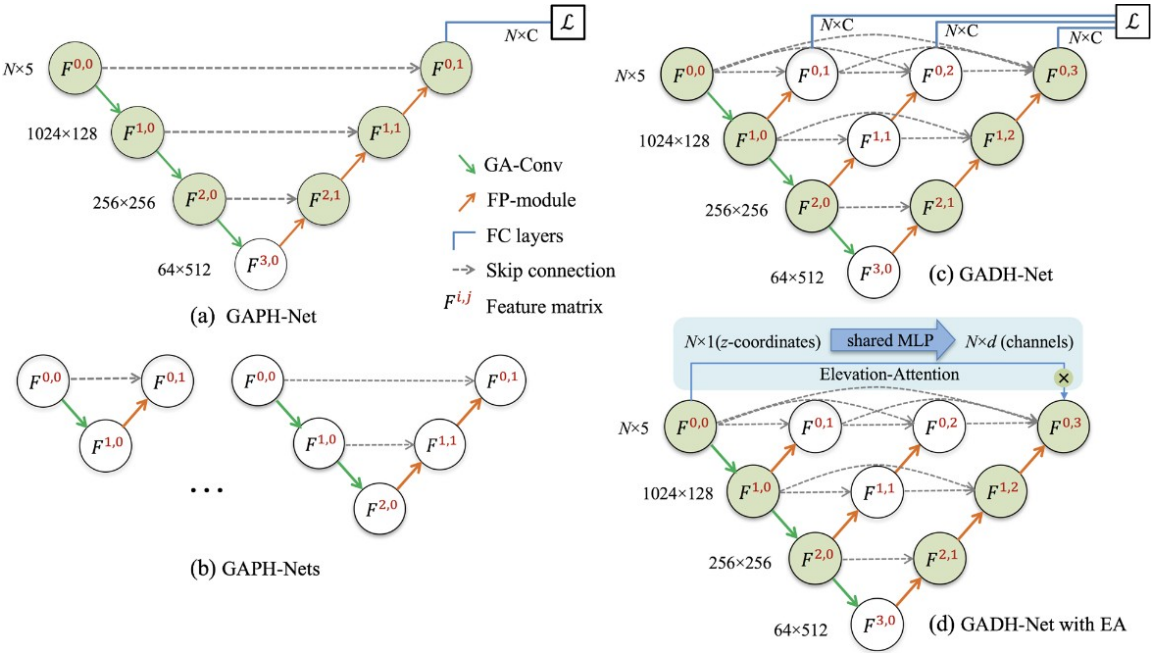

Figure 7. Illustration of GADH-Net. (a) Plain hierarchical network, GAPH-Net, with encoder, decoder, and sparse skip connections. (b) GAPH-Nets with less down-sampling layers and lower-level receptive fields. (c) Dense GADH-Net hierarchical network (d) GADH-Net with the elevation-attention module. Originally shown in [22]

Finally, for (3), an elevation-attention module is developed to emphasize elevation information when generating the final class probability map to improve classification results. The importance of elevation information in classifying ALS point clouds was studied and proven in [2] and so many of the previous classification models use HaG features as input to take advantage of elevation information. The problem is that HaG requires an initial classification of ground points, which is a tedious and time-intensive process. The authors attempt to solve this issue by building the module into the network and allow it to be trained end-to-end. To accomplish this, the z coordinates of the point cloud are passed as a one-dimensional feature vector to a three-layer MLP to map them to the same number of channels as the last feature vector in the dense network (in this case, 128 channels). The result is normalized so it can be used as elevation-attention weights and is applied to the final feature matrix of the dense network via channel-wise multiplication before classification to enable elevation-attention in the final prediction. **Figure 8** shows the elevation-attention module attached to the dense network.

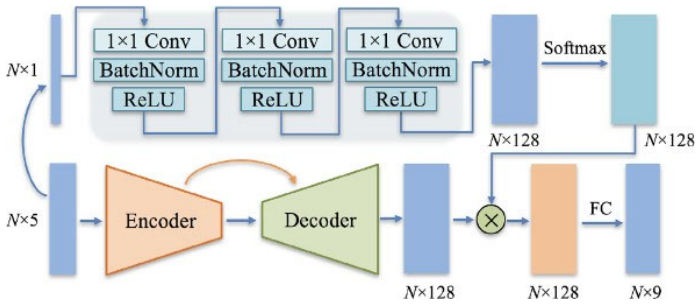

Figure 8. Illustration of the GADH-Net with EA. Originally shown in [22]

Results of this network are impressive, achieving state-of-art results in terms of F1 Score on Vaihingen 3D dataset with an average F1 score of 73.2% and outperforming the baseline PointSIFT model by 1.8% and 4.4% in OA and average F1 score, respectively. OA results are slightly below state-of-art results due to the weighted cross entropy loss used to mitigate class imbalance issues of the Vaihingen dataset. When not using a weighted loss, the model performed as well as any model for OA (85%), but F1 score drops to 71.7 [22]. It's recommended to use the weight loss model in most applications because classes that have low representation in the data will still be classified accurately, while using non-weighted loss skews towards the dominant classes. As mentioned before, most previous deep learning models (including the ones used in this results comparison) use HaG features as input to encode elevation information, but this model learns those features end-to-end, with no human input.

Generalization ability is also studied and compared to the baseline PointSIFT model on the DFC 3D dataset by predicting class labels without retraining. The model outperforms PointSIFT by more than 10% in both average F1 score and OA, and heavily outperforms it on the high vegetation class.

The dataset used is applicable to the 3DLIVE and AF use case as it is outdoor and relatively large scale, and the objects classified (such as roofs, vegetation, and cars) would be relevant to our interests for things like targeting and BDA. The results are state-of-the-art, and are especially impressive for objects with elevation differentiation (e.g. power lines). Their approach is complicated but carefully described, so it should be possible for the 3DLIVE team to understand and adjust the model, as necessary. Their code is available, and instructions are provided on how to train and run the model. Overall, this is the best classification option for the 3DLIVE use case.

3.3 Object Detection

3.3.1 Summary of Common Datasets

The two datasets that are commonly used for object detection for 3D point clouds are the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) and Waymo datasets. Both datasets were collected from mobile vehicles and contain collections of vehicles, pedestrians, trees, roadways, and various road-side objects. These datasets are commonly used to design an AI for autonomous vehicles. The 3DLIVE team aims to feed these algorithms urban datasets with segmented objects; said algorithms would ideally produce viable results despite the change in input.

Table 6: Summary of common object detection datasets

Dataset (training size, testing size)	Data Classes	Metadata
KITTI (7,481 training, 7,518 testing)	Cars, trucks, trams, pedestrians, cyclists	Tracking labels, 3D bounding boxes, 3D points, 2D camera imagery
Waymo Open Dataset (1000 scenes at 20s durations, 798 training, 202 validation)	Vehicle, pedestrian, cyclists	Tracking IDs, 3D bounding boxes for each object, 3D points, 2D camera imagery
nuScenes (1000 scenes)	Vehicles, cyclists, vegetation, flat surfaces, humans	Tracking IDs, 3D bounding boxes, 3D points, 2D camera imagery

3.3.2 Summary of Common Metrics

The main metrics used between all of the detection algorithms are: average precision (AP), mean average precision (mAP), average precision by heading (APH), mean average precision by heading (mAPH) , the KITTI benchmark and, in some cases, speed measured in Hz [23].
The measurement mean average precision is calculated by four different sub metrics:

I. The confusion matrix, which is made by the tables of true positives, false positives, true negatives, and false negatives (TP, FP, TN, FN)
II. Intersection over union (IoU), which represents the overlapped area of the true bounding box and predicted bounding box

III. Precision, which represents how well true positives were determined from all positive predictions, and is calculated as follows: **TP/(TP+FP)**
IV. Recall, which represents how well true positives were determined from all made predictions, and is calculated as follows: **TP/(TP+FN)**

In addition, multiple object tracking (MOTA), multiple object tracking precision (MOTP), average multiple objects tracking accuracy (AMOTA), id switched (IDs), and nuScenes detection score (NDS) metrics are frequently used.

3.3.3 Eliminated Articles

Attentional PointNet for 3D-Object Detection in Point Clouds

While the object detection team for this paper were able to find ten papers to discuss, only five papers seemed truly relevant. The 3DLIVE team looked for papers that had great results through the metrics chosen for the paper, whether that be AP, mAP, or mAPH. Provided here is a brief summary of each eliminated paper and an explanation for its elimination. Attentional PointNet proposed a novel deep architecture that directly operates on sparse 3D points, is end-to-end trainable, and able to learn the shape of the objects, not just the appearance of the 3D object [24]; however, this paper was eliminated due to the results performing below most of the other methods, with an AP range of 47.23 - 58.62. Comparing Attentional PointNet [24] to some of the other methods, we saw that this paper did not perform as well.

Structure Aware Single-stage 3D Object Detection from Point Cloud

He et al. proposed a way to improve the localization precision of single-stage detectors by explicitly leveraging the structure information of 3D point clouds [24]. Even though this paper had great results, achieving an AP range of 79.3 - 90.15, the paper only focused on detecting cars in the datasets, meaning it might not perform as well in other categories. Due to this, it is likely to perform poorly on the data that the 3DLIVE team wants to assess.

Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds

In Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Cloud, Complexer-YOLO was proposed as a novel 3D object detection system, which expands on its predecessor, Complex-YOLO. The system uses 3D point clouds and RGB frames from a stream in order to detect, track, and classify multiple objects [26]. While the paper and algorithm was novel, the results it received were often much lower than the other methods, only outperforming the methods in one category. The Complexer-YOLO [26] method also used RGB frames for input, which does not fit with the 3DLIVE criteria of only using LiDAR.

An LSTM Approach to Temporal 3D Object Detection in LiDAR Point Clouds
Rui Hang, et al. proposed the first LSTM-based sequential point cloud processing framework for 3D Object Detection, as well as, a 3D sparse conv LSTM, where a small 3D sparse U-Net replaces the fully connected layer in the vanilla LSTM [26]. This article was eliminated due to its poor results when compared to the non-eliminated papers, achieving only a mAP of 63.6.

PointPillars: Fast Encoders for Object Detection from Point Clouds
H. Lang et al. proposed a novel point cloud encoder and network that operates on the point cloud to enable end-to-end training of a 3D object detection network [28]. While several of the results surpassed some papers, these ones did not exceed the 3DLIVE team's expectations of novelty and usability.

3.3.4 Center-based 3D Object Detection and Tracking

The proposed method in the Center-based 3D Object Detection [29], which tracks the center of detected objects rather than each individual point, is both novel and efficient. This method is incredibly desirable due to its uniqueness in solving the object detection issue, especially when dealing with rotated objects. Not only this, but it also did out perform many of the competitors in their results section. Using a combination of metrics such as mAP, mAPH, MOTA, AMOTA, FP, FN and many more, it was able to outperform the previous state of the art methods: StarNet, PointPillars and VoxelNet when compared against the traditional detection methods.

Instead of calculating axis aligned bounding boxes (AABB) of the objects, the center-based detection method instead calculates the proposed centers of objects. By using their approach the actual rotation/direction of the objects are embedded into the center calculations and improve the accuracy of predictions. In fact, by simply switching between the original AABB method and the center based calculation method, improvements of 3-4mAP can be seen.

When comparing the results of their method against other state-of-the-art methods (StarNet, PointPillars, PPBA, RCD), Center based was able to outperform the next closest to it by a margin of 8.2 mAP and 8.1 mAPH on the Waymo dataset on vehicles and 8.6 mAP and 10.4 mAPH on pedestrians. Similar results can be found throughout all metrics that they have tested and other methods that they tested against. Such metrics include: mAP against KITTI, Waymo and the nuScenes datasets; MOTA and MOTP on the Waymo dataset; AMOTA, NDS, PKL, FP, FN, IDS on the nuScene dataset. On nearly all of these tests, this proposed center based method outperformed the competition.

This method proves to have a wide use case for the 3DLIVE team due to its novelty and techniques to capture rotated data. Since the 3DLIVE team will be working with a large scale of

outside data, cars, buildings, roads–all objects of interest that are not guaranteed to be rotated along one set of coordinate axes. This method also helps to ensure a reliable detection for objects in general, and in future work possible object tracking for the 3DLIVE team.

3.3.5 DeepFusion Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection

The DeepFusion Lidar-Camera [30] method attempts to show that state-of-the-art detection and tracking can be achieved with multi-mode fusion between Lidar data and camera data. While few have been able to prove or show this, the DeepFusion method seems to not only be a unique solution to this but also an effective one. Typically the fusion between multiple modes is done either before features for the algorithms are built, or after the features are built. If done before then the features of one sensor/collect are typically just imbued into the features of the other. For instance, camera color information being overlaid into the point cloud data gives each point a corresponding color value from the camera. Inherently this can be an issue, attempting to correctly overlap the two sensor information can be a challenge. In order to solve this problem, the DeepFusion technique adopts two different methods: InverseAug and LearnableAlign.

The InverseAug method is able to learn an inverse augmentation matrix that is applied to all of the lidar points to get them to be correctly overlaid onto the corresponding pixel in the camera's data collect. This is done by leveraging computer graphics calculations to get the points to overlay as accurately as possible into the 2D viewport. The LearnableAlign method uses a series of filters and dynamic voxelization in order to achieve better results. After using these methods the final result is passed into another model training method such as PointPillars [28] of the Center point method.

This method was able to outperform CenterPoint and many other methods. This method was specifically selected for its novelty in combining the different modes of data collection. While still achieving very good results, this method was able to combine the two common types of point cloud data that the 3DLIVE team will encounter for their efforts, raw point cloud data and bird's eye view data. By using a synthetic dataset to train this method, it should prove effective for top-down views of urban areas or even more densely populated areas like a city.

3.3.6 VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection

Around 2017 – 2018, 3D Object Detection through only the use of LiDAR was not very common, and often only giving satisfactory results. Yin Zhou and Oncel Tuzel addressed this technical gap with their object detection method, VoxelNet [31]. VoxelNet is a novel end-to-end trainable deep architecture for point-cloud-based 3D object detection consisting of three functional blocks: Feature learning network, Convolutional middle layers, and Region proposal network. Starting with the feature learning network, the point cloud data is voxelated, grouped, and randomly sampled. This saves computation time and decreases the imbalance of points between the voxels. From this point, the voxel data is sent through Voxel Feature Encoding

(VFE), which takes the data through a chain of VFE layers to output a feature that combines both pointwise features and locally aggregated features. The output is processed using the non-empty voxels to obtain the list of voxel features to help reduce memory usage and computation cost. From there, the data moves into the Convolutional middle layers. This part of the algorithm applies the output of the Feature learning network through multiple convolution layers; each layer has three components: a 3D convolution, a batch normalization layer, and a ReLU layer sequentially to aggregate voxel-wise features, adding more context to the shape description. These three components are repeated through the multiple layers of the Convolutional middle layers depending on type of detection. Finally, the feature map from the convolution step is sent to a Region Proposal Network, which maps the feature map to the desired learning targets: a probability score map and regression map. VoxelNet was trained and tested through the KITTI dataset for car, pedestrian, and cyclist detection [31].

From the architecture created by Zhou and Tuzel, VoxelNet was able to outperform many of the state-of-the-art object detection based on LiDAR by a large margin, on average obtaining an AP 5-10% higher. For performance testing, they used the KITTI validation set for bird's eye view and 3D detection and used the AP to compare VoxelNet with the results of other methods. For the Car category, VoxelNet was the highest AP of both types of detections, beating out the LiDAR only methods and all other mixed methods. For the Pedestrian and Cyclist categories, VoxelNet only had the HC-baseline method to compare against, which is also a LiDAR method. In these categories, VoxelNet was again able to beat out the HC-baseline method by a range of 0.91% - 10.78% through all test difficulties when testing against the KITTI dataset using average precision as the data metric[30].

Based on the results of VoxelNet compared to other methods on the KITTI dataset, it is shown that VoxelNet is a great method for 3DLIVE to look at for object detection. This method not only has great results, in that it outperforms much of the competition on the KITTI dataset, but also has open-source code, which helps the team understand how to use VoxelNet in future possible implementations.

3.3.7 Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud

Another novel method for point cloud object detection was done by Weijing Shi and Ragunathan Rajkumar called Point-GNN [32]. Point-GNN demonstrates the use of graph neural networks (GNN) with the ability to predict the category and shape of the object from the vertices of the graph. The architecture consists of three main components: graph construction of the point cloud, a GNN for object detection, and bounding box merging and scoring. For the construction of the graph, it starts by downsampling the point cloud data through voxelization. This helps reduce the density of the point cloud and computation time. The graph is then constructed by connecting the points of the point cloud within a certain fixed radius. This graph is then sent to a GNN with auto-registration. The GNN is designed to refine a vertex's state to include information about the object where the vertex is located; however, the relative coordinates induce translation invariance

against global shifts in the point cloud, so Shi and Rajkumar propose an auto-registration mechanism to help. The auto-registration mechanism calculates the offset using the center vertex of the previous iteration, which helps increase the accuracy of the GNN when placing detecting objects in the point cloud. Finally, the model places bounding boxes around the detected objects. Since there is a high probability of multiple vertices on the same object, it is important to have an algorithm that bounds these bounding boxes and applies a confidence score. To do this, Shi and Rajkumar consider the median position and size of the bounding boxes, then compute the confidence score by using the sum of classification scores weighted by two factors: Intersection-of-Union and occlusion. Through this architecture, Shi and Rajkumar were able to demonstrate a highly accurate way of detecting objects in a point cloud through GNN, achieving a range of AP from 40.41 - 93.11% from the tests [32].

In Point-GNN, they used the KITTI dataset to train and test their model, using AP as the comparison for both the bird's eye view and 3D object detection test sets. From the tests, Shi and Rajkumar were able to see that their method's AP score outperformed many previous papers often by a margin of 2 - 12%, beating out VoxelNet and Point Pillars in the Car and Cyclist object detection categories; however, VoxelNet did achieve the best scores in the Pedestrian category when compared to Point-GNN [32].

This method was selected not only for its novelty, but fantastic results on the KITTI Benchmark, showing that it can beat out other state-of-the-art architectures for detecting objects in a point cloud. This paper will prove useful to the 3DLIVE team, as when we look into implementing an object detection algorithm, this paper can be looked at as a novel way to complete this task with extremely desirable results. Another benefit to Point-GNN is that the code was released and can be found on the open source website GitHub. Overall, Point-GNN is a novel method that will benefit the 3DLIVE team.

3.3.8 M3DeTR: Multi-representation, Multi-scale, Mutual-relation 3D Object Detection with Transformers

The M3DeTR [33] method proposes a very novel technique that plans to solve the computational and reliability issue of combining hyperdimensional feature sets for learning algorithms. Given its main focus on developing a path between many sets of data input, it seems to still work well computationally while still remaining simple to understand. Some types of data touched upon in the paper are LiDAR voxel, point and bird's eye view data. This method is able to build these connections between the features by utilizing a set of "transformers".

The first thing that happens to the data when getting passed through the learning algorithm is to go through a feature reduction layer. The output then gets passed through the first set of Multi-Scale and Multi-Representation transformers. This transformation creates multiple sets of scales to apply on the features as well as design a full set of features to use in the later steps.

Then these features are concatenated together and passed through the final phase which is the Mutual-Relation transformers. This last transformer is designed to build the inter-point relationships between the feature sets.

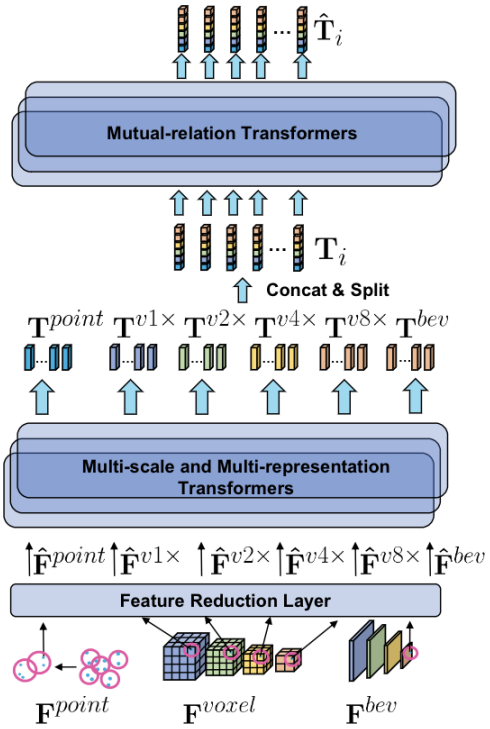

Figure 9. Illustration of the M3DeTR pipeline

M3DeTR is able to use this novel method and still achieve very promising results. On the Waymo dataset, M3DeTR was able to outperform LaserNet (see table 6), PointPillars and a few others (mAP and mAPH). On the KITTI dataset M3DeTR was also able to outperform many other proposed methods for classifying both cars and cyclists. Overall the M3DeTR method is both novel and very effective. The ability it has to combine different types of features is unprecedented, even when handling so many different feature sets (described in the paper is six) while still achieving state-of-the-art performance. This method would be very useful for the 3DLIVE efforts and with further research it may prove to be effective for other machine learning tasks such as segmentation or classification other than object detection [33].

Table 7: Data showcasing M3DeTR outperforming the previous state-of-the art methods. This was all tested on the Waymo dataset [33].

	3D mAP LEVEL_1				3D mAPH LEVEL_1				3D mAP LEVEL_2				3D mAPH LEVEL_2			
Method	Overall	0-30m	30-50m	50m-Inf	Overall	0-30m	30-50m	50m-Inf	Overall	0-30m	30-50m	50m-Inf	Overall	0-30m	30-50m	50m-Inf
LaserNet [28]	52.11	70.90	52.90	29.60	50.05	68.70	51.40	28.60	-	-	-	-	-	-	-	-
PointPillars [19]	56.62	81.00	51.80	27.90	-	-	-	-	-	-	-	-	-	-	-	-
RCD [1]	69.59	87.20	67.80	46.10	69.16	86.80	67.40	45.50	-	-	-	-	-	-	-	-
RangeDet [10]	72.85	87.96	69.03	48.88	-	-	-	-	-	-	-	-	-	-	-	-
PV-RCNN [39]	70.30	91.90	69.20	42.20	69.69	91.34	68.53	41.31	65.36	91.58	65.13	36.46	64.79	91.00	64.49	35.70
M3DeTR	**75.71**	**92.69**	**73.65**	**52.96**	**75.08**	**92.22**	**72.94**	**51.80**	**66.58**	**91.92**	**65.73**	**40.44**	**66.02**	**91.45**	**65.10**	**39.52**
Improvement	**+2.86**	**+0.79**	**+4.45**	**+3.98**	**+5.39**	**+0.88**	**+4.41**	**+6.3**	**+1.22**	**+0.34**	**+0.6**	**+3.94**	**+1.23**	**+0.45**	**+0.61**	**+3.82**

4.0 RESULTS AND DISCUSSION

To pick the best 2 models from each category, we had to consider several factors: performance of the model vs previous state-of-the-art, code availability, dataset relevance to the 3DLIVE use case, and difficulty to implement. For segmentation, the top models identified were 1) SSPC-Net and 2) MPRM (Multi-Path Region Mining) due to their state-of-art performance, code availability, and applicability to the 3DLIVE use case in terms of datasets used to train the models. SSPC-Net is ranked above MPRM because of the difficulty to implement MPRM. In order to run the MPRM model, each class appearing in the scene must be hand-identified as a preprocessing step, increasing the time requirements in a time-sensitive targeting application. The top models for classification were 1) GADH-Net and 2) D-FCN. Both models have their code available, use relevant, urban datasets, don't require any pre-training labeling, and use similar approaches, so the distinguishing factor was performance; In terms of F1 score, GADH-Net outperformed D-FCN by a notable margin, about 2.5%, so GADH-Net was ranked over D-FCN. From the articles we read, the top object detection models were 1) M3DeTR and 2) Point-GNN. Both models used novel approaches to detect objects in the KITTI dataset, with Point-GNN using a graph neural network and M3DeTR using multi-scale and multi-representation transformers. They both used relevant datasets, have code available, and are fully trainable end-to-end without any human intervention. Like the classification models, the distinguishing factor was the performance, with M3DeTR edging out Point-GNN on all categories of the KITTI benchmark (Car, Cyclist, Pedestrian) in terms of Average Precision.

5.0 CONCLUSIONS

Overall, the In-House Research Team's survey on state-of-the-art 3D point cloud segmentation, classification, and object detection methods has largely been a success. We scoured academic literature from within the past three years to find papers describing the top performing models in each of these categories. We only included works that were relevant to the 3DLIVE use case and dataset (large scale urban LiDAR point clouds) and cited by other sources. We recorded all of our findings and compared and discussed them within our team. We then eliminated the least relevant from contention, and further studied and ranked the remaining works in terms of their results and relevance to 3DLIVE. One to two models were chosen as the best approaches for the segmentation, classification, and object detection categories, which we recommend that the 3DLIVE team should move forward in trying to implement. These include SSPC-Net [12] and MPRM (Multi-Path Region Mining) [11] for segmentation, GADH-Net [22] and D-FCN [21] for classification, and M3DeTR [32] and Point-GNN [31] for object detection. There are also other good options ranked behind these best choices that could be of use, especially if there is a major issue in implementation of the top ranked models. In the future, the 3DLIVE team must determine which ML frameworks to use for the implementation of said models, and a training and test dataset of point cloud data must be properly organized. This research is a large step for the 3DLIVE team towards automated target recognition in LiDAR collected datasets, and we hope there is use in our findings towards the ML and 3D GEOINT communities at large.

6.0 RECOMMENDATIONS

The top model that we found for 3D point cloud segmentation was described in “SSPC-Net: Semi-supervised Semantic 3D Point Cloud Segmentation Network” [12]. The reason this model was decided on was because the model achieves greater performance than the standard semi-supervised point cloud segmentation methods and it does so with fewer labels, and in addition the data is relevant to the 3DLIVE use case and the code is available to recreate the model. The best approach for 3D point cloud classification for 3DLIVE was determined to be “A Geometry-Attentional Network for ALS Point Cloud Classification” (GADH-Net) [22]. We fell towards this model being the best choice because (a) results of the network are impressive, achieving state-of-art results in terms of F1 Score, (b) the dataset used is applicable to the 3DLIVE and AF use case as it is outdoor and relatively large scale, and (c) the approach is carefully described and the code is available with instructions which should make their approach relatively straightforward to recreate. For 3D point cloud object detection, we found the best approach to be “M3DeTR: Multi-representation, Multi-scale, Mutual-relation 3D Object Detection with Transformers” [33]. This is because it achieves noticeable improvements on the KITTI and Waymo datasets, has a novel yet recreate-able approach using relevant data to 3DLIVE, and the code is available. For the 3DLIVE use case, a machine-learning automated target recognition approach will likely start with segmentation. We will take one of the most relevant segmentation models and use it to separate out the points that constitute individual objects of interest within the dataset. These segmented objects will then need to be classified (identified), and this calls for the implementation and use of the best choice classification model to do this job. There is also the possibility for the alternate approach of using the top object detection model to draw bounding boxes around objects of interest, with a classification model used subsequently to determine the identity of said bounded objects. The next step is for the 3DLIVE team to take the top segmentation, classification, and object detection approaches, determine how to recreate their methods, and use them on 3DLIVE's point cloud data in order to perform automated target recognition.

7.0 REFERENCES

[1] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep Learning for 3D point clouds: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, Jun. 2020.

[2] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, and L. Xiao, "A review of algorithms for filtering the 3D point cloud," *Signal Processing: Image Communication*, 22-May-2017. [Online]. Available: https://fardapaper.ir/mohavaha/uploads/2019/06/Fardapaper-A-review-of-algorithms-for-filtering-the-3D-point-cloud.pdf. [Accessed: 19-Jul-2022].

[3] R. A. Rosu, P. Schütt, J. Quenzel, and S. Behnke, "LatticeNet: Fast Spatio-Temporal Point Cloud Segmentation Using Permutohedral Lattices," arXiv.org, 09-Aug-2021. [Online]. Available: https://arxiv.org/abs/2108.03917?context=cs.LG. [Accessed: 26-Aug-2022].

[4] M. Xu, Z. Zhou, and Y. Qiao, "Geometry Sharing Network for 3D Point Cloud Classification and Segmentation," arXiv.org, 23-Dec-2019. [Online]. Available: https://arxiv.org/abs/1912.10644. [Accessed: 26-Aug-2022].

[5] X. Chen, K. Jiang, Y. Zhu, X. Wang, and T. Yun, "Individual Tree Crown Segmentation Directly from UAV-Borne LiDAR Data Using the PointNet of Deep Learning," MDPI, 24-Jan-2021. [Online]. Available: https://www.mdpi.com/1999-4907/12/2/131. [Accessed: 26-Aug-2022].

[6] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "RPVNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation," arXiv.org, 24-Mar-2021. [Online]. Available: https://arxiv.org/abs/2103.12978. [Accessed: 26-Aug-2022].

[7] S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye, and F.-Y. Wang, "SCF-Net: Learning Spatial Contextual Features for Large-Scale Point Cloud Segmentation," IEEE Xplore, 02-Nov-2021. [Online]. Available: https://ieeexplore.ieee.org/document/9577763. [Accessed: 26-Aug-2022].

[8] A. Xiao, J. Huang, D. Guan, F. Zhan, and S. Lu, "Transfer Learning from Synthetic to Real LiDAR Point Cloud for Semantic Segmentation," arXiv.org, 12-Jul-2021. [Online]. Available: https://arxiv.org/abs/2107.05399. [Accessed: 26-Aug-2022].

[9] H. Lei, N. Akhtar, and A. Mian, "SegGCN: Efficient 3D Point Cloud Segmentation With Fuzzy Spherical Kernel," IEEE Xplore, 05-Aug-2020. [Online]. Available: https://ieeexplore.ieee.org/document/9157177. [Accessed: 26-Aug-2022].

[10] L. Tang, Y. Zhan, Z. Chen, B. Yu, and D. Tao, "Contrastive Boundary Learning for Point Cloud segmentation," arXiv.org, 11-Mar-2022. [Online]. Available: https://arxiv.org/abs/2203.05272. [Accessed: 26-Aug-2022].

[11] J. Wei, G. Lin, K.-H. Yap, T.-Y. Hung, and L. Xie, "Multi-Path Region Mining For Weakly Supervised 3D Semantic Segmentation on Point Clouds," arXiv.org, 29-Mar-2020. [Online]. Available: https://arxiv.org/abs/2003.13035. [Accessed: 26-Aug-2022].

[12] M. Cheng, L. Hui, J. Xie, and J. Yang, "SSPC-Net: Semi-supervised Semantic 3D Point Cloud Segmentation Network," arXiv.org, 24-May-2021. [Online]. Available: https://arxiv.org/abs/2104.07861. [Accessed: 26-Aug-2022].

[13] E. Özdemir, F. Remondino, and A. Golkar, “An Efficient and General Framework for Aerial Point Cloud Classification in Urban Scenarios,” MDPI, 19-May-2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/10/1985. [Accessed: 26-Aug-2022].
[14] Z. Jing, H. Guan, P. Zhao, D. Li, Y. Yu, Y. Zang, H. Wang, and J. Li, “Multispectral Lidar Point Cloud Classification Using SE-PointNet++,” MDPI, 27-Jun-2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/13/2516. [Accessed: 26-Aug-2022].
[15] E. Özdemir, F. Remondino, and A. Golkar, “Aerial Point Cloud classification with Deep Learning and machine learning algorithms,” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 18-Oct-2019. [Online]. Available: https://www.int-arch-photogramm-remote-sens-spatial-inf-sci .net/XLII-4-W18/843/2019/. [Accessed: 26-Aug-2022].
[16] M. Zhang, H. You, P. Kadam, S. Liu, and C.-C. J. Kuo, “PointHop: An Explainable Machine Learning Method for Point Cloud Classification,” arXiv.org, 16-Dec-2019. [Online]. Available: https://arxiv.org/abs/1907.12766. [Accessed: 26-Aug-2022].
[17] X. Lei, H. Wang, C. Wang, Z. Zhao, J. Miao, and P. Tian, “ALS Point Cloud Classification by Integrating an Improved Fully Convolutional Network into Transfer Learning with Multi-Scale and Multi-View Deep Features,” PubMed, 06-Dec-2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33291256/. [Accessed: 26-Aug-2022].
[18] Q. Zang, W. Diao, K. Chen, L. Liu, M. Yan, and X. Sun, “CBF-Net: An Adaptive Context Balancing and Feature Filtering Network for Point Cloud Classification,” IEEE Xplore, 20-Aug-2021. [Online]. Available: https://ieeexplore.ieee.org/document/9520261. [Accessed: 26-Aug-2022].
[19] Y. Chen, G. Liu, Y. Xu, P. Pan, and Y. Xing, “PointNet++ network architecture with individual point level and global features on centroid for ALS Point Cloud Classification,” MDPI, 29-Jan-2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/3/472. [Accessed: 26-Aug-2022].
[20] C. Wen, X. Li, X. Yao, L. Peng, and T. Chi, “Airborne LiDAR Point Cloud Classification with Graph Attention Convolution Neural Network,” arXiv.org, 20-Apr-2020. [Online]. Available: https://arxiv.org/abs/2004.09057. [Accessed: 26-Aug-2022].
[21] C. Wen, L. Yang, L. Peng, X. Li, and T. Chi, “Directionally Constrained Fully Convolutional Neural Network for Airborne Lidar Point Cloud Classification,” arXiv.org, 19-Aug-2019. [Online]. Available: https://arxiv.org/abs/1908.06673. [Accessed: 26-Aug-2022].
[22] W. Li, F.-D. Wang, and G.-S. Xia, “A Geometry-Attentional Network for ALS Point Cloud Classification,” ISPRS Journal of Photogrammetry and Remote Sensing, 09-Apr-2020. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0924271620300861. [Accessed: 26-Aug-2022].

[23] D. Shah, “Mean average precision (MAP) explained: Everything you need to know,” *V7*, 07-Oct-2022. [Online]. Available: https://www.v7labs.com/blog/mean-average-precision. [Accessed: 12-Oct-2022].
[24] A. Paigwar, O. Erkent, C. Wolf, and C. Laugier, “Attentional PointNet for 3D-object detection in point clouds,” IEEE Xplore, 09-Apr-2020. [Online]. Available: https://ieeexplore.ieee.org/document/9025598/. [Accessed: 26-Aug-2022].
[25] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, “Structure aware single-stage 3D object detection from Point Cloud,” ieeexplore, 05-Aug-2020. [Online]. Available: https://ieeexplore.ieee.org/document/9157660. [Accessed: 26-Aug-2022].
[26] M. Simon, K. Amende, A. Kraus, J. Honer, T. Sämann, H. Kaulbersch, S. Milz, and H. M. Gross, “Complexer-Yolo: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds,” arXiv.org, 16-Apr-2019. [Online]. Available: https://arxiv.org/abs/1904.07537. [Accessed: 26-Aug-2022].
[27] R. Huang, W. Zhang, A. Kundu, C. Pantofaru, D. A. Ross, T. Funkhouser, and A. Fathi, “An LSTM Approach to Temporal 3D Object Detection in Lidar Point Clouds,” arXiv.org, 24-Jul-2020. [Online]. Available: https://arxiv.org/abs/2007.12392. [Accessed: 26-Aug-2022].
[28] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “Pointpillars: Fast Encoders for Object Detection from Point Clouds,” arXiv.org, 07-May-2019. [Online]. Available: https://arxiv.org/abs/1812.05784. [Accessed: 26-Aug-2022].
[29] T. Yin, X. Zhou, and P. Krähenbühl, “Center-based 3D Object Detection and Tracking,” arXiv.org, 06-Jan-2021. [Online]. Available: https://arxiv.org/abs/2006.11275. [Accessed: 26-Aug-2022].
[30] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, B. Wu, Y. Lu, D. Zhou, Q. V. Le, A. Yuille, and M. Tan, “DeepFusion: LIDAR-Camera Deep Fusion for Multi-Modal 3D Object Detection,” arXiv.org, 15-Mar-2022. [Online]. Available: https://arxiv.org/abs/2203.08195. [Accessed: 26-Aug-2022].
[31] Y. Zhou and O. Tuzel, “VoxelNet: End-to-end learning for point cloud based 3D object detection,” arXiv.org, 17-Nov-2017. [Online]. Available: https://arxiv.org/abs/1711.06396. [Accessed: 26-Aug-2022].
[32] W. Shi and R. Rajkumar, “Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud,” arXiv.org, 02-Mar-2020. [Online]. Available: https://arxiv.org/abs/2003.01251. [Accessed: 26-Aug-2022].
[33] T. Guan, J. Wang, S. Lan, R. Chandra, Z. Wu, L. Davis, and D. Manocha, “M3detr: Multi-representation, Multi-scale, Mutual-relation 3D Object Detection with Transformers,” arXiv.org, 22-Oct-2021. [Online]. Available: https://arxiv.org/abs/2104.11896. [Accessed: 26-Aug-2022].

8.0 GLOSSARY

Term	Definition
ALS - Airborne Laser Scanning	See **LiDAR**.
ATR - Automatic Target Recognition	Ability for an algorithm or device to automatically recognize targets or other objects based on data obtained from sensors.
BDA - Battle Damage Assessment	Estimate of damage composed of the physical damage assessment (PDA) and functional damage assessment (FDA), as well as target system assessment, resulting from the application of lethal or nonlethal military force.
Computer Vision	Field of artificial intelligence that trains computers to gain a higher understanding of the world via digital images.
Convolution	Application of a filter to a function that produces a new function
CNN - Convolutional Neural Network	In deep learning, a convolutional neural network is a class of artificial neural network most commonly applied to analyze visual imagery.
Deep Learning	Machine learning methods based on artificial neural networks which can be supervised, semi-supervised or unsupervised.
Decoder	Reconstructs the original input from the encoded representation.
Discretization	Process of transforming a continuous function into a discrete function.
Downsampling	Process of reducing the sampling rate of a signal.

Encoder	Condenses input data into a fixed-length representation while retaining pertinent feature information.
Feature	A measurable property or characteristic of an entity.
Generalization	Model's ability to adapt properly to new, previously unseen data.
GPU - Graphics Processing Unit	Processor designed to handle graphics operations. This includes both 2D and 3D calculations, though GPUs primarily excel at rendering 3D graphics. Also used for machine learning calculations.
ISPRS - The International Society for Photogrammetry and Remote Sensing	Typically referring to the ISPRS Valhingen dataset used for classification benchmarking.
Kernel	A matrix that adjusts the weight given to each input value.
KITTI - Karlsruhe Institute of Technology and Toyota Technological Institute	Typically referring to the dataset used for object detection benchmarking.
LiDAR - Light Detection and Ranging	Method for determining ranges by targeting an object or a surface with a laser and measuring the time for the reflected light to return to the receiver.
Loss	Penalty for a bad prediction.
Loss function	Method of evaluating how well a machine learning algorithm models the featured data set.
MLP - Multi-Layer Perceptron	Fully connected class of feedforward artificial neural network.
PPM - Precise Point Mensuration	Measurement of geographic coordinates of a precise point in geodetic space for aimpoint targeting.
TCM - Target Coordinate Mensuration	See **Precise Point Mensuration**.

Upsampling	Process of increasing the sampling rate of a signal.
Voxel	Unit of information that defines a point on a regular grid in 3D space.
Voxelization	The conversion of an image or model into voxels.
3DLIVE - Three-Dimensional LiDAR Visualization and Exploitation	In-House AFRL/RIED effort which aims to utilize LiDAR data for visualization and TCM.

Distribution List

-DTIC has requirements for distribution statements that we will have to work to satisfy