IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 7, July 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

9940 572 462 | 6381 907 438 | ijircce@gmail.com | www.ijircce.com

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

Deep Reinforcement Learning Framework for Semantic Parsing of Large-scale 3D Point Clouds using 3DCNN-DQN-RNN

Shrujana N [1], Dr H K Madhu [2]

MCA Student, Department of Computer Application, Bangalore Institute of Technology, Bangalore, India[1]

Associate Professor, Department of Computer Application, Bangalore Institute of Technology, Bangalore, India[2]

ABSTRACT: Semantic parsing of large-scale 3D point clouds is a key area of research in computer vision and remote sensing. Traditional methods rely on manually created features and combine them in a simplistic way, often missing the connection and complementary nature of the data. Current deep learning techniques perform well for images but struggle with 3D point clouds due to their unstructured nature and varying point densities.

This paper introduces a new method called 3DCNN-DQN-RNN, which combines a 3D Convolutional Neural Network (3D CNN), a Deep Q-Network (DQN), and a Residual Recurrent Neural Network (RNN). This method uses an "eye window" controlled by the 3D CNN and DQN to efficiently locate and segment objects within the point cloud. The 3D CNN and Residual RNN then extract strong, distinctive features from these segments, improving parsing accuracy. The proposed method automates the process of mapping raw data to classification results and integrates object localization, segmentation, and classification into one framework. Experiments show that this approach outperforms current state-of-the-art methods for point cloud classification.

I. INTRODUCTION

In recent times, deep literacy has been veritably successful in feting images, speech, and textbook. still, classifying large- scale 3D point shadows is still grueling because these point shadows are unorganized and have uneven point consistence, making it hard to directly dissect them.

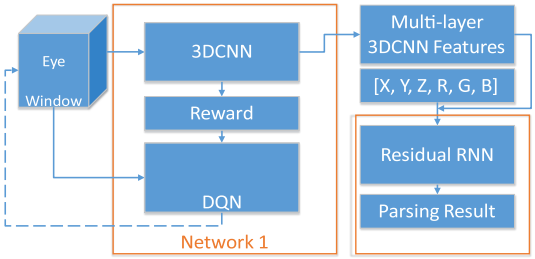

Figure 1: The framework of the proposed approach. (X, Y, Z) and (R, G, B) represent 3D coordinate and RGB colors of each point P_i in the original point cloud.

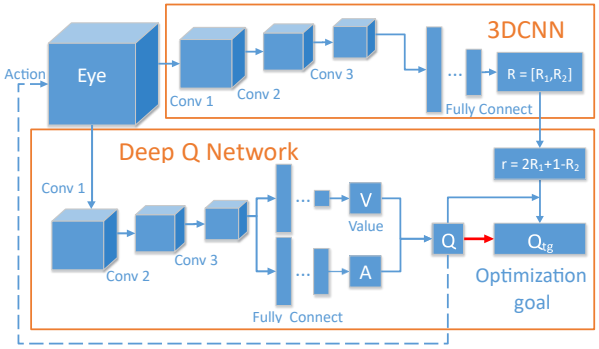

Figure 2: Structure of Network 1

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

This paper proposes a new deep underpinning literacy system called 3DCNN- DQN- RNN to automatically dissect large- scale 3D point shadows. The 3D CNN learns features similar as shape, spatial connections, color, and environment from multiple scales. An" eye window" controlled by the Deep Q- Network(DQN) moves through the data to find and member objects. The 3D CNN evaluates these parts, and the DQN adjusts the eye window's size and position to ameliorate delicacy. Once an object is set up, its features are fed into a Residual intermittent Neural Network(RNN) for final bracket.

The main benefactions are
1. A new deep underpinning learning model that simplifies parameter tuning and integrates object localization, discovery, reclamation, and bracket into one frame.
2. The Residual RNN combinesmulti-scale features, equals, and colors into a robust representation, achieving high bracket performance.
3. The DQN- controlled eye window directly and automatically localizes and parts objects, making the process more effective.

II. RELATED WORK

Numerous recent styles for parsing point shadows calculate on classifiers trained on hand- drafted features like shapes, colors, and contextual information. exemplifications include using Random timbers with 21 features to classify point shadows into five orders, or SVM with 13 features. These traditional approaches frequently miss the connections between features necessary for landing high- position semantic structures. For case, Chehata etal. used Random timbers for bracket, while Kragh etal. used SVM, and Lafarge and Mallet linked objects like structures, and foliage. Deep literacy ways have lately been applied to 3D data, enabling automatic point literacy. For illustration, volumetric CNNs have been used for object bracket and reclamation, and 3D CNNs have been employed for tasks like landing zone discovery in LiDAR data. ways similar as PointNet have further advanced 3D shape bracket and scene parsing. Despite their success in specific operations, these deep literacy styles still struggle with directly parsing large- scale point shadows without any significant homemade trouble. likewise, some recent approaches essay to integrate original and global features for better delicacy. For case, Song etal. presented a 3D ConvNet channel for 3D object discovery, combining 2D and 3D features. Armeni etal. used a sliding window approach to prisoner both original and global features for inner point shadows. These styles, still, bear expansive hand- drafted design work, similar as defining sliding windows or hunt boxes, which limits their effectiveness and scalability Object segmentation styles generally concentrate on 2D images. ways like completely connected networks(FCNs) and DeepLab have been effective forpixel-wise bracket in 2D. still, applying these styles to large- scale 3D point shadows is challenging due to issues like missing data. For illustration, SSCNet has been developed for segmenting RGBD images but faces difficulties when dealing with expansive 3D data. To address these challenges, our proposed system leverages a 3D CNN for point birth, a Deep Q- Network(DQN) to control an eye window for object localization, and a Residual intermittent Neural Network(RNN) for final bracket. This approach integrates object localization, segmentation, and bracket into one frame, automating the process and reducing homemade trouble. By combiningmulti-scale features, equals, and colors, the system achieves robust and accurate parsing of large- scale 3D point shadows. This comprehensive integration helps overcome the limitations of former styles, furnishing a more effective and scalable result for 3D point pall parsing.

III. PROPOSED APPROACH

Humans tackle vision challenges by integrating various techniques, as reinforcement learning and sensory. Specialists believe that advancements in computer vision will largely stem from systems that combine ConvNets (Convolutional Networks) with RNNs (Recurrent Neural Networks) and reinforcement learning to determine points of focus. In this paper, we adopt this concept to examine extensive 3D point clouds. Similar to how humans initially glance at a scene to locate a target and then concentrate on that target to distinguish it from the background, we introduce a novel method that utilizes deep reinforcement learning to recognize individual objects within a 3D point cloud environment, as illustrated in Figure 1.

3.1 Network 1: Recognition and Localization via 3D CNN DQN

Network 1 consists of two principal components. The first portion employs a 3D CNN to identify various objects within a 3D scene and generate unique features for each. The second element is a DQN that leverages insights from the 3D CNN to detect and pinpoint objects. We initiate the process by sliding a fixed-size “eye window” across the scene, using the 3D CNN to interpret the contents. The DQN determines where to position the window and its dimensions based on the CNN’s output. This cycle continues until the window adequately encompasses the objects. Eventually, the

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

features derived from the 3D CNN and the points within the window serve as input for the next phase of the network. Figure 2 demonstrates the collaborative functioning of the 3D CNN and DQN.

3.2 Learning Multi-Scale Features with the 3D CNN

In expansive 3D environments, objects may differ greatly in size, shape, and placement, complicating accurate classification with a single method. To address this issue, we implement a sophisticated strategy instead of merely applying a 3D CNN directly to the entire point cloud data. Our technique involves a dynamic "eye window" that scans various sections of the scene. Initially, we conduct multi-scale convolution, analyzing the environment at different levels of detail to grasp the intricate structures and spatial relationships among objects of diverse sizes.

This convolution produces a reward vector that indicates the probability of different object classes being present within the eye window. The reward vector includes confidence scores (R1 and R2) and parameters (θ) from the 3D CNN, which assist in assessing our certainty regarding the detected object classes. Subsequently, a Deep Q-Network (DQN) utilizes this reward vector along with the data from the eye window to modify its size and position. The DQN iteratively adjusts the window to concentrate on areas where the target object is likely to be until it successfully identifies the object with great accuracy. After the target object is precisely located, we gather features from the 3D CNN that represent the points encompassed by the eye window. These features play a crucial role in the subsequent stages of the process.

[R1, R2] are calculated using the formula:

$f(E(p, q)) = \theta \top E(p, q) = 2R1 + 1 - R2 = r$ (1)

To manage the unstructured nature of point cloud data, we first convert it into a 3D voxel grid. This transformation organizes the point cloud into a systematic grid of small cubes, known as voxels. Each voxel is assigned a specific position in the grid (X, Y, Z) and color information (R, G, B), reflecting the average color of the points contained within it. By structuring the point cloud data into these consistent, manageable units, we enhance the CNN's ability to analyze and extract meaningful features from the data, improving object classification and detection accuracy.

3D CNN Structure:

The 3D CNN comprises three main convolutional layers followed by several fully connected layers. We begin by feeding voxel grids from the "eye window" into the CNN. Each layer of the CNN applies filters to construct a 3D feature matrix from the voxel grids. The filter sizes and strides differ across the layers: Conv1 utilizes 8 filters of size 5 with a stride of 3, Conv2 employs 16 filters of size 4 with a stride of 2, and Conv3 applies 32 filters of size 3 with a stride of 1. Following these convolutions, we utilize average pooling to amalgamate the results and pass them through two fully connected layers. Ultimately, a softmax activation function yields a 1x2 reward vector ([R1, R2]), where R1 indicates the probability of the target class being present, and R2 signifies the probability of it being absent. This reward vector aids the DQN in determining whether the eye window accurately encapsulates the target object.

3.1.1 Object Class Detection by the DQN

The 3D CNN does not tag each point within the eye window individually but rather estimates the likelihood that the points within the window belong to a specific class. It generates a high probability only when the eye window is accurately positioned and sized around the object of interest. This method assists in identifying both the precise location and boundaries of the target object. We can frame the task of locating the correct object in the scene as an optimization problem, where the goal is to adjust the eye window's dimensions and position to maximize the likelihood of the target class being present.

$E(pb,qb)* = \text{argmax}(f(E(p,q)))$ (2)

Our aim is to ascertain the optimal position (pb) and size (qb) for the eye window to reliably identify objects within the scene. The eye window shifts and adjusts based on the feedback it receives. When the window deviates from the appropriate position or size, it modifies itself and attempts again to locate the object. This process involves a systematic decision-making approach, where the 3D CNN assesses the current status of the eye window and provides a probability of locating the object. This probability is subsequently utilized by the DQN (Deep Q-Network) to determine the next action for the eye window, thereby enhancing its ability to accurately locate and identify the object. The DQN incorporates techniques such as Prioritized Replay and Dueling Network to boost search efficiency. Figure 3 illustrates how this search process operates, with a video demonstration available in the supplementary materials.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

Figure 3: Two angles of eye window searching tables. Locked as marked green.

To depict the current condition, the 3D CNN in our system gathers features from the eye window. The DQN (Deep Q-Network) then makes use of these properties to determine what steps are optimal for object location. Value and Advantage are the two divisions of this work according to the DQN.
Value (V) indicates how good the eye window is doing overall right now.
• Advantage (A) compares each potential course of action to other courses of action.

To carry out this, the 3D After passing through the DQN, CNN's output is divided into two sections: one section determines the state's value, while the other half determines the benefit of each course of action. These two components add up to the action's final Q-value.

The value of the state is calculated in one component, while the advantage of each action is calculated in the other. The value of the state plus the action's advantage add up to the ultimate Q-value for a given action. By using this technique, the DQN is able to make more informed decisions about how to best position and enlarge the eye window in order to more successfully locate the target object.

$$\mathbf{Q(s,a;\Theta,\alpha,\beta) = V(s,\Theta,\beta)+A(s,a';\Theta,\alpha)-\sum a' A(s,a;\Theta,\alpha}_{\mathbf{||A||}} \quad \mathbf{(3)}$$

Policy: How to Take Action
The eye window in this system has six sides, and it can expand, contract, or remain unchanged on each side. This means there are always 18 possible activities that the eye window can do. Using the formula *a = argmax Q(s, a; θt)**, we determine which action to do by selecting the one with the highest Q-value for the present state.

Unlike classical approaches, which save all potential state-action combinations, we just store the eye window's size and position. The system becomes more effective as a result. Finding the optimal set of steps that will result in the largest possible total payout is the aim. The DQN models N subsequent actions to determine the ideal order of

$$\mathbf{Qtg} = \boldsymbol{tanh}\left(\sum_{t=0}^{N-1} \boldsymbol{\lambda^t rt} + \lambda^N Q'\right) \quad \mathbf{(4)}$$

where r = 2R1 + 1 − R2 and [R1,R2] is the reward vector. Q′ is the final Q after N times of simulation; λ is the decay coefficient satisfying λ ∈ [0,1]. Approaching the limitation situation—which ought to be expressed as P∞ t=0 λtrt—is the aim of Q′. When the current Q experiences a gradient descent to Qtg, the network parameters are updated:

$$\theta T+1 = \theta T + \lambda(Qtg - Q(s,a;\theta T))\nabla\theta T\ Q(s,a;\theta) \quad (5)$$

The DQN parameters are updated following a thorough N-step simulation; that is, the DQN does an initial probe into the various strategies within N steps and stores the probe result as new network parameters for use in making subsequent decisions. To obtain k times probe simulations and parameter update, we run the simulations k times. When the last parameter is changed, the largest Q is used to select an action.

Random Walk and Winner Replay
We employ two tactics, Random Walk and Winner Replay, to assist the eye window in progressing when it becomes stuck or is unable to move because of low Q-values.

In order to prevent becoming caught in a loop, Random Walk enables the eye window to respond randomly when it becomes stuck.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

By remembering past prosperous states, Winner Replay assists the eye window in making more intelligent selections. We monitor the optimal results from several simulations following a sequence of operations. In this scenario, the eye window will allocate half of its time to selecting actions based on the optimal past result (the "winner"), and the other half based on standard N-step simulations. This approach improves the **DQN's efficiency**, reducing the searching time by **73% on average**.

Algorithm 1 3DCNN-DQN Algorithm

Input:

max iterate step mis, max reside step mrs, max simulate step mss, decay rate λ, mark threshold mth, state-reward dictionary rd, replay memory rm, Q value network Q_{net}, reward network R_{net}

$Is \leftarrow 0$, $Rs \leftarrow 0$, $Ss \leftarrow 0$,$cS \leftarrow initial\ state$

while $Is < mis$ **do**

  action $(a, Q) \leftarrow Q_{net}(cS, a; W)$

  $cS \leftarrow environment(cS, a)$, $Rs \leftarrow 0$

  while $Rs < mrs$ **do**

    $Ss \leftarrow 0$, simulate state set $S \leftarrow []$

    reward set $R \leftarrow []$, $S[Ss] \leftarrow cs$

    while $Ss < mss$ **do**

      action $(a, Q) \leftarrow Q_{net}(S[Ss], a; W)$ or rm

      $S[Ss + 1] \leftarrow environment(S[Ss], a)$

      $Ss \leftarrow Ss + 1$

      if $S[Ss]$ not in $rd's\ key\ set$ **then**

        $r' \leftarrow R_{net}(S[Ss])$

        $rd[S[Ss]] \leftarrow r'$

      end if

      $R[Ss - 1] \leftarrow rd[S[Ss]]$

    end while

    $Q_{tg} \leftarrow \tanh(\sum_{t=0}^{N-1} \lambda^t r_t + \lambda^N Q)$

    Update Q's W using Q_{tg}

    $Rs \leftarrow Rs + 1$

  end while

  $Is \leftarrow Is + 1$

end while

Lock the Target

Lock the Target When the 3D CNN determines that the eye window has successfully set up a class object(when the price is0.9 or advanced), we " lock the target " by labeling the points in the eye window as part of the object. We also prize features from these points using the layers of the 3D CNN.

For each subcaste, we produce point vectors(f1, f2, f3) and combine them into one long point vector that represents the points ' features in the eye window. This point vector includes both the CNN- generated features and the original point information, similar as position(x, y, z) and color(r, g, b). We also use this combined point vector, which is called the 3D CNN point, as the input for Network 2 to continue the processing of the point pall.

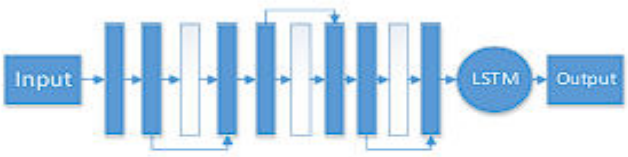

Figure 4: The inner structure of Network 2. Blue rectangle means fully-connected layers, white means drop layer.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

Network 2 Residual RNN for scrupulous Parsing Network 2 uses a Residual RNN to dissect the features of the points set up by the eye window. The RNN treats the points as a sequence, analogous to how a retired Markov model processes data. It learns the connections and differences between features at colorful scales to understand the point pall's complex spatial structure. To manage this complexity, the RNN is deep with numerous layers and uses LSTM cells to handle long- term dependences and avoid issues like grade evaporating. Residual Blocks help maintain performance indeed with a veritably deep network. This setup helps the RNN directly interpret and parse the point pall data by learning both original details and global patterns.

Residual RNN Structure Residual RNN takes reconstructed vectors from the points in the eye window and processes them to ameliorate object parsing. Each vector, Vk, contains the point's position(xk, yk, zk)(x_k,y_k,z_k)(xk, yk, zk), color(rk, gk, bk)(r_k,g_k,b_k)(rk, gk, bk),
and features from the
3D CNN(f1, f2, f3)(f1, f2, f3)(f1, f2, f3).
These vectors are input into the Residual RNN, which has 7 completely- connected layers, 3 powerhouse layers, 2 residual blocks, and an LSTM cell. Figure 4 shows this structure. The RNN processes the vectors in their original spatial order, using Residual Blocks to keep the network effective at deeper situations and LSTM cells to flash back long- term features.

LSTM CELL The Residual RNN in Network 2 deals with a large quantum of data from the eye window. This means the sequence of points fed into the RNN is long. To effectively manage this data, Network 2 uses an LSTM cell to learn long- term connections among the points, landing both the connections and differences in their features.

Residual Block In deep neural networks, adding further layers frequently makes the network perform worse, a problem known as declination. For point pall parsing, a deep network is demanded, but it risks this declination. To break this, we use Residual Blocks, a conception from the Deep Residual Network. These blocks produce lanes or skip connections between layers to save and ameliorate performance as the network gets deeper, precluding the crimes that come from having too numerous layers.

Deep RNN- AMulti-Layer Classifier In Network 2, we use amulti-layer neural network to handle the complex features from Network 1. The features we get are from colorful object types and scales, so we need a sophisticated system to combine them effectively. Simple classifiers like SVM or Random Forest are n't enough because they 're too introductory for this task. rather, ourmulti-layer neural network uses deep literacy to combine features grounded on position, spatial connections, and color, leading to better and more accurate bracket. trials We trained our 3D CNN and RNN models using two NVIDIA K40 GPUs with Python and TensorFlow to make full use of the GPUs' computing power.
For our trials, we used two datasets:

Stanford 3D Semantic Parsing Dataset
o **Content** This dataset contains 3D reviews from 271 apartments, covering a aggregate of 6020 square measures, with objects like ceilings, bottoms, walls, and cabinetwork.
o **Training** and Testing We used 70 of the apartments for training and the remaining 30 for testing.
o **Method F**or simpler structures like ceilings and walls, we used Network 2 directly for bracket. For other objects, we applied our full system to classify them. We also compared our results to other styles to check how well our approach works. 2. SUNCG Dataset
o **Content** This dataset includes 45,622 house models and 84 classes of objects.
o **Method** We converted this data into 3D point shadows with attributes like equals and color. The results are shown in the supplementary accoutrements .
Comparison of Our system with the FCN We compared our 3D point pall parsing system to a **3D Completely Convolutional Network**(FCN), which is analogous to our approach because both styles label different corridor of the scene. still, for large 3D scenes, the spatial connections are veritably complex, and our system handles this better.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	mean
N1+N2	89.64	95.02	60.08	**78.55**	89.36	**75.29**	33.41	70.48	58.14	76.98	**84.97**	37.21	**70.76**
S1	71.61	88.70	**72.86**	66.67	**91.77**	25.92	54.11	46.02	16.15	6.78	54.71	3.91	49.93
N1	**96.97**	**100.00**	24.10	12.74	27.40	32.88	**91.48**	**77.41**	50.53	**87.17**	53.84	32.83	57.28
N2	87.51	97.66	27.45	32.96	3.90	67.27	15.16	10.77	**68.17**	68.47	12.91	**43.56**	36.63
FCN-12	20.32	11.61	8.34	12.69	33.36	20.41	10.01	9.68	11.27	2.24	1.89	13.86	12.97
FCN-6	-	46.58	12.97	-	-	29.55	11.65	50.96	-	-	20.32	-	28.64
FCN-1	-	87.62	-	55.34	-	63.46	30.68	-	60.27	-	80.37	-	62.96
S2	-	-	-	-	-	-	-	46.67	33.80	4.76	-	11.72	24.24

Table 1: N1+N2 is our method. S1, S2 correspondingly refer to the method of Armeni et al. *[1] and Qi* et al. *[20]*

IV. COMPARISON OF OUR SYSTEM WITH THE FCN

We compared our 3D point pall parsing system to a 3D Completely Convolutional Network(FCN), which is analogous to our approach because both styles label different corridor of the scene. still, for large 3D scenes, the spatial connections are veritably complex, and our system handles this better.

Then's how we did the comparison

• **What We Compared** We used the VGG- 16 model, which is a well- known image processing model, and acclimated it to work with 3D data as a 3D FCN.

• **Comparison Process** We trained both our system and the 3D FCN on the same dataset and tested them under the same conditions.

• **Results** Our system outperformed the 3D FCN in terms of bracket delicacy. The results are shown in Table 1.

We compared our system with a 3D Completely Convolutional Network(FCN) to see how well each handles 3D point shadows for large- scale scenes.

Then's what we set up 1. Why the FCN is Poorer

• Point shadowsvs.2.5 D Data Our 3D point shadows are different from simpler2.5 D data like RGBD images. Point shadows have large quantities of data, are noisy, have uneven point viscosity, and are disordered, making them harder to dissect.

• Need for a Deeper Network To get good results with a 3D FCN, you need a important deeper and more complex network, which requires a lot further computational power and time.

2. Why Our Method is More

• Effective Object Localization Our system uses the DQN(Deep Q- Network) to control an eye window that searches for and locks onto objects more directly and efficiently. This approach requires lower computational power and time compared to the deeper networks demanded for the FCN.

FCN- 1, FCN- 6 and FCN- 12 mean to classify just one class, six classes and twelve classes at a time, independently. FCN- 1 fails to fete the border, and the overall accu-racy isn't satisfactory. In FCN- 6 and FCN- 12, the accu-racy reduces to a veritably low position. likewise, in FCN- 12, the loss function does not meet.

RNN Helps to Raise Precision

When using only Network 1, the delicacy for relating walls in the point pall is relatively low. This happens because walls are anatomized latterly, after other objects have been linked. During this after stage, the eye window might inaptly include points from near objects, like a table or president, which leads to crimes in classifying the wall.

Network 2(RNN) fixes these problems. It goes back andre-checks the points, perfecting the delicacy for wall discovery and other objects that were parsed latterly. By correcting miscalculations and redefining the points, Network 2 makes sure that objects are classified rightly indeed when they're close to each other.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

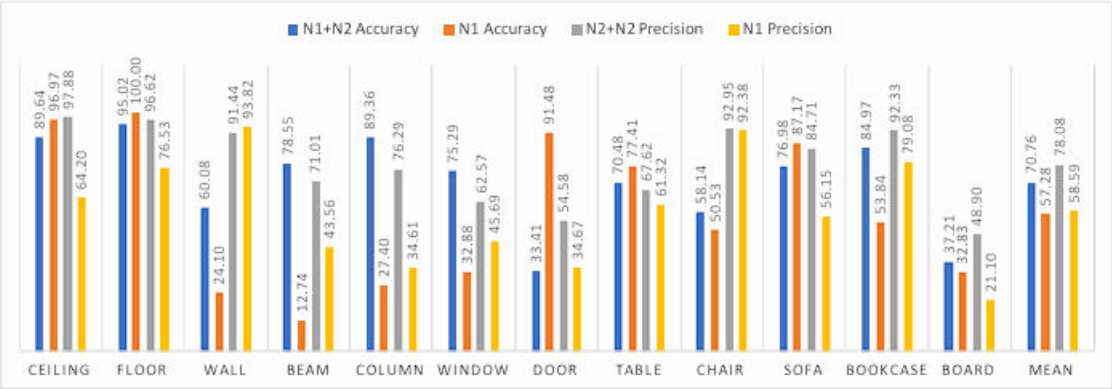

Figure 6: Accuracy and Precesion of our method. N1, N2 represent Network 1 and Network 2.

Influences of the DQN on Point Cloud Parsing

The effectiveness of the DQN(Deep Q- Network) in point pall parsing is told by how long it runs. For the first dataset, running the DQN for 5000 duplications per class takes an normal of 13 hours to localize all 12 classes.

The longer the DQN runs, the better the delicacy generally becomes, as it allows the eye window to explore further countries of the room. still, not all areas are inversely useful for observation; the eye window might waste time looking at insignificant corridor. To ameliorate effectiveness, the authors plan to develop a new underpinning learning strategy for better and faster localization in unborn work.

3D CNN is Not Well Trained T

he main issue with our system is that the 3D CNN(Convolutional Neural Network) isn't well- trained. Unlike in DQN operations for games, where conduct always get clear feedback from the terrain, object discovery from small point shadows is much harder. The 3D CNN struggles to learn detailed features from these small and frequently analogous-looking objects. As a result, the eye window occasionally makes miscalculations, similar as confusing a small bookcase for a table or mistaking a glass door for a window. These inaccuracies can lead to crimes in object localization and bracket, like including part of the wall when locking onto a window. To ameliorate the system's performance, training the 3D CNN on a larger dataset could help it more separate between objects and enhance overall bracket delicacy.

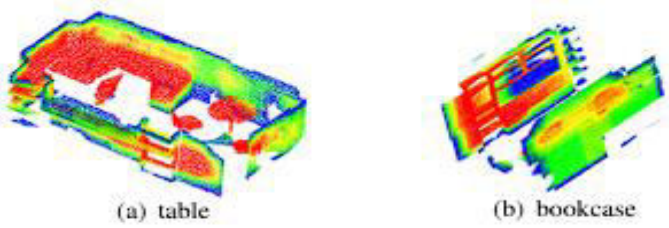

Figure 7: Activity thermodynamic diagram of an eye window.

V. CONCLUSION

In this paper, we introduce a new system called 3DCNN- DQN- RNN for automatically assaying large 3D point shadows. Then's a simple breakdown

1. 3D CNN This part of the system excerpts important features from 3D data, like shapes, colors, and spatial connections, and turns them into a useful representation of the scene.
2. DQN The Deep Q- Network(DQN) controls an eye window to look at different corridor of the 3D scene. It helps the system find and concentrate on different objects grounded on prices.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

3. Residual RNN This intermittent Neural Network(RNN) processes the information from the eye window to directly classify the objects in the scene.

Overall, our system not only identifies objects but also helps in localizing, detecting, and reacquiring them.

VI. FUTURE PLANS

unborn Plans In the future, we will use a new fashion called A3C(Asynchronous Advantage Actor- Critic) to ameliorate how multiple eye windows work together, making the process of finding and classifying objects indeed more and more effective.

ACKNOWLEDGMENTS

We'd like to admit the support from the National Natural Science Foundation of China for funding this exploration under Grant 41371324.

REFERENCES

[1] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese. 3d semantic parsing of large scale indoor spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,, 2016.

[2] N.Chehata, L. Guo, and C. Mallet. Airborne lidar feature se lection for urban classification using random forests. Inter national Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 38(Part 3):W8, 2009.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep con volutional nets and fully connected crfs. International Con ference on Learning Representations (ICLR), 2015.

[4] K. Fukushima. Neocognitron: A self-organizing neu ral network model for a mechanism of pattern recogni tion unaffected by shift in position. Biological cybernetics, 36(4):193–202, 1980.

[5] R. Girshick. Fast r-cnn. Proceedings of the IEEE Interna tional Conference on Computer Vision, pages 1440–1448, 2015.

[6] H. Guan, Y. Yu, Z. Ji, J. Li, and Q. Zhang. Deep learning based tree classification using mobile lidar data. Remote Sensing Letters, 6(11):864–873, 2015.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. pages 770–778, 2016.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural computation, 9(8):1735–1780, 1997.

[9] A.E.Johnson and M.Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. IEEE Transactions on pattern analysis and machine intelligence, 21(5):433 449, 1999. 2

[10] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena. Se mantic labeling of 3d point clouds for indoor scenes. Ad vances in Neural Information Processing Systems, pages 244–252, 2011.

[11] M. Kragh, R. N. Jørgensen, and H. Pedersen. Object de tection and terrain classification in agricultural fields using 3d lidar data. International Conference on Computer Vision Systems, pages 188–197, 2015.

[12] F. Lafarge and C. Mallet. Creating large-scale city mod els from 3d-point clouds: a robust approach with hybrid representation. International journal of computer vision, 99(1):69–85, 2012.

[13] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521(7553):436–444, 2015.

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. pages 3431–3440, 2015.

[15] A. Martinovic, J. Knopp, H. Riemenschneider, and L. VanGool. 3dall the way: Semantic segmentation of urban scenes from start to end in 3d. Proceedings of the IEEE Con ference on Computer Vision and Pattern Recognition, pages 4456–4465, 2015.

[16] D. Maturana and S. Scherer. 3d convolutional neural net works for landing zone detection from lidar. IEEE Inter national Conference on Robotics and Automation (ICRA), pages 3471–3478, 2015.

[17] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pages 922–928, 2015.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. NIPS Deep Learn ing Workshop, 2013.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M.G.Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep rein forcement learning. Nature, 518(7540):529–533, 2015.

[20] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[21] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classi f ication on 3d data. pages 5648–5656, 2016.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 7, July 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1207056 |

[22] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. 2005 IEEE Computer So ciety Conference on Computer Vision and Pattern Recogni tion (CVPR'05), 2:994–1000, 2005.
[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR), 2015.
[24] S. Song and J. Xiao. Sliding shapes for 3d object detection in depth images. pages 634–651. Springer, 2014.
[25] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. pages 808–816, 2016.
[26] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. Proceedings of 30th IEEE Conference on Com puter Vision and Pattern Recognition, 2017.
[27] Z. Wang, L. Zhang, T. Fang, P. T. Mathiopoulos, X. Tong, H. Qu, Z. Xiao, F. Li, and D. Chen. A multiscale and hierar chical feature extraction method for terrestrial laser scanning point cloud classification. IEEE Transactions on Geoscience and Remote Sensing, 53(5):2409–2425, 2015.
[28] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1912–1920, 2015.
[29] B. Yang, Z. Dong, Y. Liu, F. Liang, and Y. Wang. Com puting multiple aggregation levels and contextual features for road facilities recognition using mobile laser scanning data. ISPRS Journal of Photogrammetry and Remote Sens ing, 126:180–194, 2017.
[30] F. Yu and V. Koltun. Multi-scale context aggregation by di lated convolutions. International Conference on Learning Representations (ICLR), 2016.
[31] L. Zhang, X. Huang, B. Huang, and P. Li. A pixel shape index coupled with spectral information for classifi cation of high spatial resolution remotely sensed imagery. IEEE Transactions on Geoscience and Remote Sensing, 44(10):2950–2961, 2006.
[32] L. Zhang, L. Zhang, D. Tao, and X. Huang. Tensor discrim inative locality alignment for hyperspectral image spectral spatial feature extraction. IEEE Transactions on Geoscience and Remote Sensing, 51(1):242–256, 2013.
[33] Z. Zhang, L. Zhang, X. Tong, P. T. Mathiopoulos, B. Guo, X. Huang, Z.Wang,andY.Wang. Amultilevelpoint-cluster based discriminative feature for als point cloud classifica tion. IEEE Transactions on Geoscience and Remote Sensing, 54(6):3309–3321, 2016.
[34] Q.-Y. Zhou and U. Neumann. Complete residential ur ban area reconstruction from dense aerial lidar point clouds. Graphical Models, 75(3):118–125, 2013

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details