Journal of Theoretical and Applied Information Technology
31st March 2025. Vol.103. No.6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPLEMENTING NEUROMORPHIC COMPUTING USING NEURAL NETWORK TECHNOLOGY FOR DYNAMIC OBJECT DETECTION AND OCCLUSION HANDLING IN AR IMAGES

[1] ARUNA THETHALI, [2] MANDAVA KRANTHI KIRAN

[1] Research Scholar, Department of CSE, GITAM Deemed to be University, Visakhapatnam, India
[2]Department of CSE, GITAM Deemed to be University, Visakhapatnam

ABSTRACT

In recent times, people, objects, and other real-time elements have played a major role in real-time videos. Most scenes are taken behind natural objects like trees, grid wires, wireframes, etc. Identifying, detecting, and tracking the objects from real-time videos is inaccurate since the objects are over-placed on one another. Some earlier research has focused on providing occlusion detection in face recognition to identify human faces and obtained only 80% accuracy. In real-time industries like virtual and augmented reality, video occlusion is high, and objects must be identified accurately. This paper focused on occlusion removal and object detection to improve visualization accuracy. To increase object detection accuracy without occlusion, this paper proposes an SNN-based neuromorphic system to detect objects from occluded images. The mask-RCNN model is applied to segment the input samples before detection to detect the objects accurately. The overall workflow of the proposed SNN-based model uses four steps: data pre-processing, feature extraction, data segmentation, and detection. The entire model is implemented and experimented with AR video images, and the results are verified. The output shows that SNN and MRCNN increased the overall efficiency in object detection and visualization, occlusion removal, reduced time, and computational complexities.

Keywords: *Neuromorphic Computing, Spiking Neural Networks, MR-CNN, Occlusion Removal, Object Detection.*

1. INTRODUCTION

A new technique called augmented reality (AR) enables computer-generated graphics to be overlaid in images or video taken using the camera in real-time [1]. Augmented reality (AR) technology aims to increase the real world with extra virtual content, including images, objects, audio, or other non-visual media, while maintaining real-time user interaction [2]. The two objectives of AR are enhancing the user's perception of the surrounding world and increasing their performance on a task by integrating the real and the virtual world coherently. For the user to feel as though they are interacting in a single realistic environment, the virtual items should ideally appear in a way that allows for natural interaction. For this reason, in the AR world, the effects, such as shadows, illumination, and reflection, that we observe in real life should also appear realistically [3].

By providing additional information or enriching the user experience, this AR technology is frequently used to increase perception. According to recent findings, AR has a great deal of promise in numerous fields, such as industrial [4], medical [5, 6], education [7, 8] and entertainment [9]. Breen et al. [10] state that a seamless integration of real and virtual objects within the user's environment is necessary for AR to achieve maximum potential and complete acceptance. When we try to achieve this effect, three main types of problems arise illumination, tracking, and occlusion. When an object in closer range to the viewer blocks the vision of further distance, this is called occlusion (Figure-1). The virtual objects occlude the real objects in most AR applications, though the opposite can happen occasionally. Occlusion is an essential visual effect that properly blends the real and virtual worlds. For example, incorrect occlusion may occur when the virtual object moves behind a real object [11].

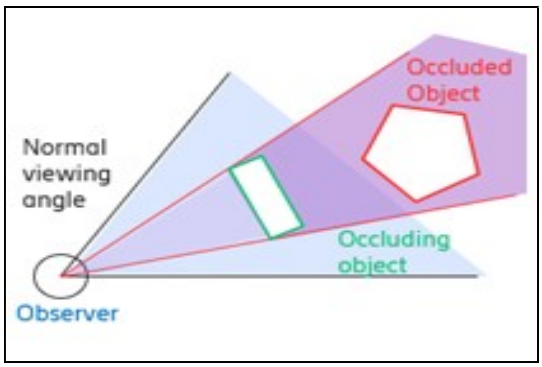

Fig.1.1. Occlusion Effect [3]

In Augmented Reality (AR) applications, precise depth perception depends on dynamic occlusion handling. Thus, it is a vital component to guarantee realistic and immersive AR experiences. Current approaches to address these problems usually have numerous limitations, such as the assumption of a static scene or high computational complexity [12]. Researchers have paid more attention to the occlusion issue to improve the illusion that virtual objects exist in the real scene. When the real objects in the scene are in front of the virtual objects, an occlusion problem is created. When the real objects in the scene occlude virtual objects, users may mistakenly think that the real object is further from the viewpoint than the virtual objects if the occlusion handling is not implemented. This increases eyestrain and the probability of motion sickness; it also makes the user misjudge the spatial properties of items and make errors when trying to snatch objects [13, 14].

For handling occlusion, the model-based approach will be unsuitable when the geometric models of the relevant objects are not obtainable. The depth-based method is only suitable for the static environment, making it useless when the camera or the real environment changes. One of the most challenging problems in AR applications is handling occlusion between real and virtual objects. Occlusion handling is one of the important factors for maintaining visual consistency. Various methods are applied to obtain the depth information needed for occlusion handling in an application scenario involving small indoor areas [15]. User may feel non-realistic and non-immersive AR if occlusion handling is done inaccurately or incorrectly, confusing their vision [3, 16]. Hence, this paper focuses on solving the occlusion problem in AR images using the neuromorphic computing technique. This technique can analyze and understand massive amounts of data more efficiently than traditional computing architectures. It can mimic the human brain's neural architecture and provides unparalleled efficiency for complex and sensory-driven tasks. Because of their efficiency, they are applicable for adaptive learning and real-time processing [17].

In addition, neural networks help handle the problems related to occlusion. For example, Wei et al. [18] used the CNN algorithm by integrating with AR to ensure the accuracy and interactivity of the model in recognition and handling occlusion. To detect the occlusion edges in images, a deep convolutional neural network (CNN) was trained by Sarkar et al. [19]. Since this CNN can visualize intermediate optimized filters and extract hierarchical features (features of features) from data, they are highly suitable for this problem. According to many recent experiments, deep neural network models with biological-inspired features could accurately reconstruct corrupted representations, even for highly occluded objects [20]. However, few studies have focused on handling this occlusion using neuromorphic computing. In this study, we can achieve an excellent architecture to handle occlusion in AR images by combining neuromorphic computing with neural networks. Many traditional object detection models perform poorly while dealing with overlapping objects, which is problematic for visualization and tracker performance. Many existing approaches obtain a lack of accuracy that does not exceed 90% in terms of object identification. This paper contributes the following using MRCNN and SNN models. It improves the speed of implementation. It is used different datasets for training the models. It improves the accuracy of object detection, occlusion removal, and visualization.

2. LITERATURE SURVEY

This section discusses the efficiency of neuromorphic computing and its functionality in applying various real-time applications. Further, the efficiency of the traditional neuromorphic computing-based approaches and their limitations are discussed. Neuromorphic computing is entirely based on the inspirations of the operative characteristics of the human brain, and the idea has been adopted in implementing neuromorphic machines for several years; conversely, it is not confirmed whether the computational performance is entirely based on the biological brain alone. For instance, even though neurons and synapses have been considered the primary units of computers, more types of neural elements may be helpful for

computation, like glial cells. Polykretis, I et al. (2020) [21], Irizarry-Valle Y and Parker A C (2015) [22]. In the neuromorphic computing field, the availability of datasets was considered one of the main challenges. To overcome this, the conversion approaches and conventional datasets into neuromorphic was carried out by Orchard et al. (2015) [23]. In addition to keeping them suitable with the versions of the existing systems, it includes creating information essential to achieving the advantages of neuromorphic systems in real-time applications. Shastri, B. J et al. (2021) [24] focused on the latest advancements in integrated photonic neuromorphic systems. The authors analyzed the present and future issues, and technology-based solutions were proposed to resolve such issues. The authors concluded that the applicability of neuromorphic photonics in enhancing the limitations of data science fields. Schuman, C. D. As Yin, S et al. (2017) [25] stated, a key difference exists between neuromorphic computing and a few future computing fields. The authors said several neuromorphic hardware components were under development, and a few were utilized in the research. Numerous large-scale neuromorphic machines were developed using various approaches and techniques.

Dunleavy and Dede, C. (2014) [26] reviewed augmented realities and their applications in pedagogical approaches by utilizing them as cognitive tools. The authors stated that AR could be the ideal and efficient tool for instructing approaches carried out by academic professionals. The authors demanded further research on applying the strategy to resolve the continuous issues in education while agreeing with the limitations of such approaches. Arena F et al. (2022) [27] presented a detailed augmented reality view. The authors started with the basic elements of AR and explained its real-time applications. The authors mentioned that compared to VR, the design of AR is for permanent utilization in the movement where the gradual replacement of the smartphone is possible. They stated that the real metaverse would have a significant impact on the movements of humans and would isolate people from the real physical world. They added to overcome issues such as privacy, tracking every behavior, surveillance, and constant availability.

Macedo, M. C., and Apolinario, A. L. (2021) [28] deeply reviewed the techniques and methods for occlusion handling problems in AR applications, and the survey was taken from various articles published over the past decades. Widely applying techniques in ordering the real and virtual objects were firmly focused, which is essential for visualizing the hidden objects and building visual displays capable of handling occlusions. Additionally, modern and novel techniques were observed, and present issues and essential future requirements for improving AR technologies were discussed. However, the study did not take into account real-time performance constraints which are critical for AR applications particularly when interacting with them in dynamic environments. Regarding AR applications, the occlusion problem was framed as an alpha-matting problem by Hebborn, A. K. et al. (2017) [29], and they proposed an algorithm. The proposed approach helped differentiate regions such as invisible and visible; it also got even variations by evaluating the reinterpreted value as a blending coefficient, and the approach was found to be better than the present novel techniques. The authors concluded that future works should focus on facing occlusion issues. Similarly. However, the alpha-matting method is highly computationally demanding, making them potentially ill-suited for AR types that require high image speed. Jorge, J. et al. (2019) [30] got inspiration from the proposed algorithm of Hebborn, A. K. et al. (2017), and they suggested a few changes to the alpha matting approach to enhance the quality of the outcomes. To improve the algorithm, various filters were applied to enable the sensors to obtain data, and adaptive dilation and post-processing steps were carried out. Hence, the changes were applied and used to achieve ideal results with stability. This paper is highly efficient since it implements Mask-RCNN to obtain the mask region of the objects. It also uses several important processes like image resizing, bounding box drawing, and different styles of boxes to differentiate objects. Nevertheless, although it seems to show high processing speed, setting a lower learning rate as 0.02 in the context of full model training might incur a possible convergence steady state problem, leading to a model trained with poor generalization ability.

3. PROPOSED METHODOLOGY

This paper proposes an SNN-based neuromorphic system to detect objects from occluded images. The mask-RCNN model is applied to segment the input samples before detection to detect the objects accurately. The overall workflow of the proposed SNN-based model uses four steps: data pre-processing, feature extraction, data segmentation,

and detection. Integrating mask R-CNN and SNN on pre-processed input images increases overall efficiency. Initially, all the input images obtained directly from the AR videos are pre-processed separately and fed to the neural network architecture. The output data from the conventional neural networks is generated into spikes to represent the objects presented in the input images. The objects are obtained by generating the bounding boxes to segment the individual objects present in the video images. The paper's novelty is that feature pyramid networks and ResNet-101 pre-trained models are used to improve efficiency.

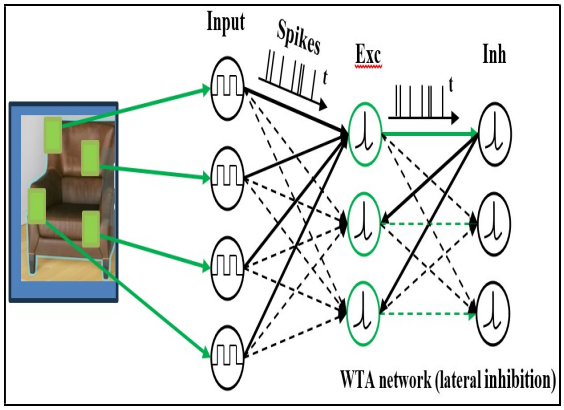

Fig.3.1. SNN Architecture

3.1 Spiking Neural Network

The Spiking Neural Network resembles a biological neural network based on the mathematical descriptions of the biological neurons. Leaky-and-Fire neuron, a spiking neuron model, is mostly adopted due to its simple operations in which binary spikes facilitate communication between neurons. The scalar weights are inspired by the synapses connecting the different neurons. The deep neural network has the potential to learn the weights, whereas the spikes are indistinguishable and distinct; due to these natures, the backpropagation learning algorithm cannot be used on the SNNs. The SNN can be developed by rate coding to create static data, transforming it into a spike domain that becomes a trained DNN. Even though the results were comparatively less than those of the direct learning of temporary data by the SNNs, the usage of the backpropagation algorithm in spikes was overcome by adopting various learning rules in the later days. The weight that links the two neurons is converted based on the presynaptic and post-synaptic firing. This bio-plausible unsupervised learning rule is called Spike-timing-dependent plasticity (STDP). The result obtained by the STDP is not matchable to the supervised learning for simple tasks; over a period of time, the scenario changes. However, the STDP is ineffective in solving real-world problems like object detection. Recently, the SNNs facilitated by supervised learning based on backpropagation have yielded better results. Further, it is enhanced by the SLAYER, an error-based backpropagation model that makes the SNNs learn synaptic weight and axon delays. It allows them to handle large datasets with deep networks. One more method with surrogate gradient learning is used to train the SNNs. In the method above, a Heaviside step function creates a spike in the forward pass, and the surrogate gradient approximates the gradient of the indistinguishable function. The general structure of the SNN architecture is shown in below Fig.3.

It proves that the SNNs are equal to the RNNs and support learning with back propagation over time, which includes this model in one of the widespread deep learning networks. With the help of the surrogate gradient learning rule, many tools like Nengo and Spiking jelly, as well as a multiple Pytorch-based framework, have been developed to train the SNNs. The new framework's automatic differentiation and Strong GPU acceleration have made the deeper spiking neural networks yield state-of-the-art output in classification-based issues. Another study confirms that the neuron parameters are learnable, such as the Parametric-LIF neuron model, which has the learnable time constant, making a fast-learning method less sensitive to starting values.

3.2 Mask R-CNN

The concept of Mask R-CNN is simple: In Faster R-CNN, every candidate object has two outputs: a class label and a bounding box offset. In addition, a third branch will be added to get an object mask as an output. This is why the mask R-CNN is considered a logically acceptable concept. On the other hand, the mask output is different from the other two inputs; it needs an exact spatial layout extraction. Compared to faster R-CNN, some key elements are available in mask R-CNN, like pixel-to-pixel alignment.

3.3 Faster R-CNN

There are two stages in Faster R-CNN: The RPN (Regional Proposal Network) is the first stage, in which the proposal of the candidate object is implemented, and in the second stage, the features

are extracted by utilizing RoIPool from every candidate box and executing categorization and bounding-box regression.

For faster inference, sharing the features of the two stages is possible. Comparing the faster R-CNN should be focused on, and recent and broad approaches for such a task should be observed. A deep neural network called Mask RCNN aims to solve the instance segmentation problem in machine learning or computer vision.

This The Diabetic retinopathy data sets are classified as trained data sets, validation datasets and testing data sets. 60% of the data sets are considered as trained data sets. 20% of the data sets are considered as validation data sets and the remaining 20% of the data sets are considered as testing data sets

The trained datasets and validation data sets are given as input to the YOLOv8 algorithm. This supervised learning is used for training purposes. Object Labeling is done for each input image. Modified CSP Darknet 53 is used as backbone of the classifier. This consists of 53 convolution layers. This backbone is pretrained CNN (Convolution neural network) will extract low, medium, and high-level features of the image. The feature map of the CSP Net base layer is decomposed into two parts.

It can distinguish between various objects in an image or a video. It can accept an image as an input and output object, bounding boxes, classes, and masks. The mask RCNN process has two steps.In the first stage, based on the input image, proposals about the regions where an object might be produced are made. Based on the proposal of the first stage, this second stage predicts the object's class, improves the bounding box, and creates a mask in the object's pixel level. The backbone structure connects both steps. The two stages of the mask-RCNN model are illustrated in Figure-3. The FPN-style deep neural network serves as the backbone, comprising a bottom-up pathway, a top-bottom pathway, and lateral connections. The bottom-up pathway can be any ConvNet, but ResNet or VGG is regularly used to extract features from raw images. A feature pyramid map generated by the Top-bottom pathway is similar in size to the bottom-up path. Convolution and addition of operations between two corresponding levels of the two pathways generate lateral connections. The FPN performs better than single ConvNets because it preserves vital semantic features at several resolution scales.

All FPN top-bottom pathways are scanned by a lightweight neural network known as RPN scans, which suggests the regions that may hold objects. Even though scanning feature maps is an efficient method, there is a need to link features to their raw image location. A set of boxes with predetermined locations and scales relative to images are called anchors. Individual anchors are allocated ground-truth classes and bounding boxes based on IoU value. RPN uses anchors with varying scales bind to different levels of the feature map to find out the size of the bounding box with the object's location in the feature map. Convolving, downsampling, and upsampling, in these cases, would retain features staying in similar relative locations as the objects in the original image without altering them.

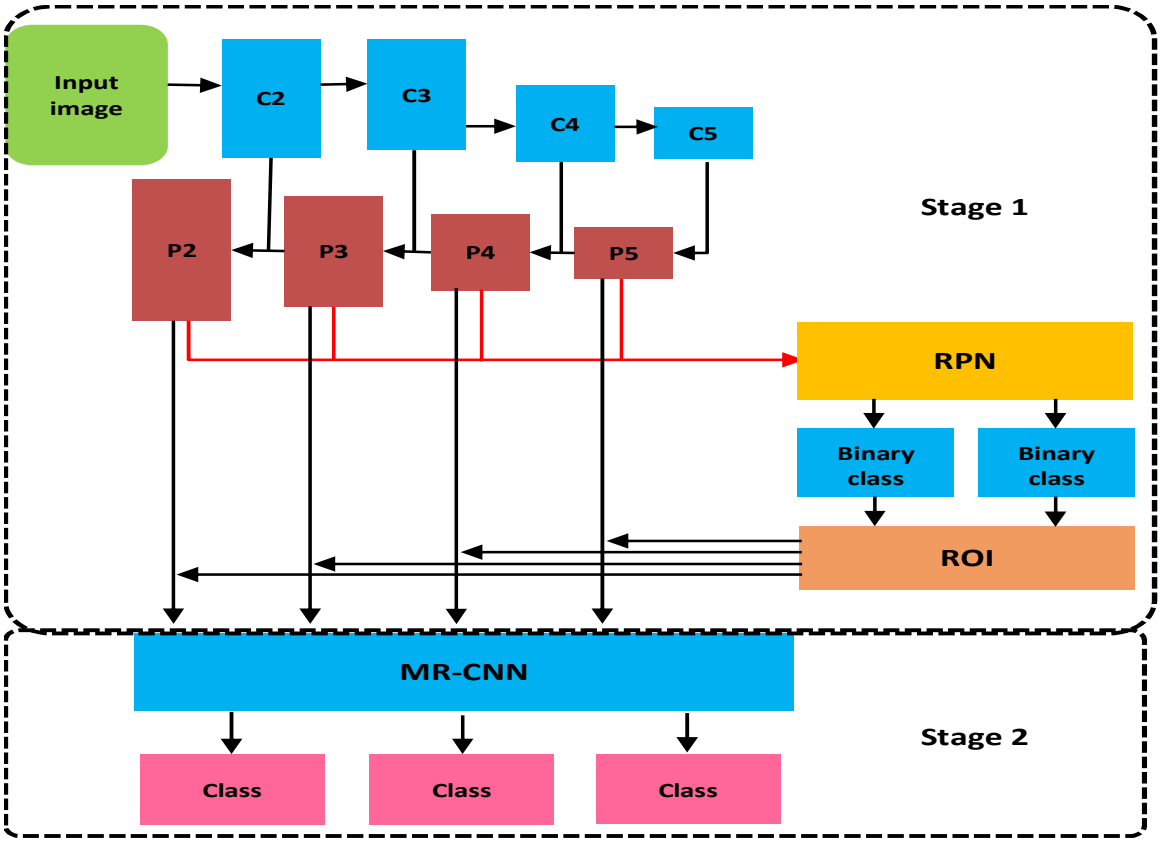

Fig.3.2. Two stages of Mask RCNN

3.4 Two Stages of Mask R-CNN

The two-stage procedure is used in mask R-CNN, like Faster R-CNN: The first stage of the mask, R-CNN, is the same as the faster R-CNN. According to the second stage, as a simultaneous process along with the prediction of the class and box offset, Mask R-CNN creates a binary mask output relating to every ROI. Conversely, in terms of the present systems, mask predictions play a role in determining the classification. The proposed approach is based on the way of fast R-CNN as described by R. Girshick (2015) [31], where regression and the bounding box classification are utilized simultaneously, and according to the training, for every sampled ROI, the definition of a muti-task loss is known as $L = L_{cls} + L_{box} + L_{mask}$. Based on the author's definition, the classification (L_{cls}) and bounding box (L_{box}) losses are similar.

Every ROI has a Km^{2}-dimensional output in the mask branch, where K binary masks of resolution m × m are encoded. Every K class would have one. A per-pixel sigmoid is applied, and L_{mask} is defined as the average binary cross-entropy loss. For a RoI related to ground-truth class k, L_{mask} is only described on the k^{th} mask since other outputs are not associated with the loss. The L_{mask} definition helps create masks for every class in the network, and the classes will not compete with each other.

The class label prediction for choosing the output mask depends on the dedicated classification branch. This distinguishes mask and class prediction. This is not similar to the typical practices when FCNs are utilized for semantic segmentation, Long J et al. (2015) [32], where a per-pixel softmax and a multinomial cross-entropy are generally used. The masks from various classes compete with each other in such cases, but with a per-pixel sigmoid and a binary loss, it will not occur in the proposed case. The experimental

outcomes prove that this formulation is observed to be the key to ideal instance segmentation results.

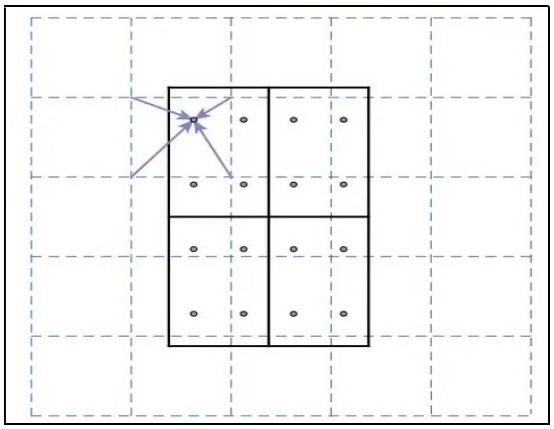

Fig.3.3. Region Of Interest Align

Fig.3.3. illustrates the RoI Align: A feature map is depicted by the dashed grid; the solid lines represent an ROI with 2×2 bins, and the dots represent the 4 sampling points. In the feature map, the values of the sampling points are calculated by RoI Align. The coordinates involved in the RoI, sampling points, or bins are not quantized

3.5 Mask Representation

A mask encodes the spatial layout of an input object. Consequently, Avoiding the collapsing performance of the fully connected layers over the class labels or box offsets into short output vectors is impossible. However, dissimilar to this, the extraction process of the spatial structure of masks can be identified by the pixel-to-pixel communication given by convolutions. In particular, an FCN is used to predict an m × m mask from each RoI. Because of this, every layer in the mask branch can preserve the spatial layout of m × m objects without being collapsed into a vector representation with no spatial dimensions. In terms of mask prediction, compared to previous approaches that do not require fully connected layers.

Some parameters are needed for fully convolutional representation, which is more perfect as validated by experiments. Small RoI features are required for this pixel-to-pixel behavior to be fairly aligned to regulate clear per-pixel spatial communication confidently. This scenario motivated the development of the RoIAlign layer and its key performance in mask prediction.

3.6 Alignment of Region-of-Interest

While considering extracting a small feature map from each RoI, the RoI Pool is a standard operation. Firstly, due to the discrete granularity of the feature map, a floating number is quantized by the RoI pool. The subdivision is performed over the quantized RoI into spatial bins, and they are quantized themselves. Eventually, the feature values covered by each bin are aggregated. The quantization performance happens, for instance, while considering a continuous coordinate x by calculating [x/16], where 16 denotes a feature map stride and [·] denotes rounding. Similarly, the quantization performance happens when separating into bins, for instance, 7×7. The mistakes in the alignments between the extracted features and RoI are identified by quantization. This does not impact the classifications essential to small translations; however, this negatively affects the prediction of pixel-accurate masks.

An RoI Align layer capable of removing the harsh quantization of RoIPool is proposed to meet these, and the alignment of the extracted features and the input is perfectly carried out. The proposed change is quite simple, where any quantization of the RoI boundaries or bins is avoided; for instance, x/16 is taken in place of [x/16]. The bilinear interpolation is utilized in computing the accurate values of the input features at four commonly sampled positions in each RoI bin, and the result is aggregated by using the maximum or average. Figure-5 depicts this briefly. It is observed that the results are independent of the accurate sampling locations or the number of sampled points until the quantization performance is carried out. Significant improvements are possible because of RoI Align. The RoI Warp operation proposed by Dai J. et al. (2016) [33] was considered in the comparison. It was observed that RoIWarp was not similar to RoIAlign, the alignment problem was not considered, and the implementation was quantized RoI like RoIPool. Despite RoIWarp utilizing bilinear resampling inspired by Jaderberg, M. et al. (2015) [34], it achieves the equivalent of RoIPool, as demonstrated by the experiments.

ISSN: 1992-8645 E-ISSN: 1817-3195

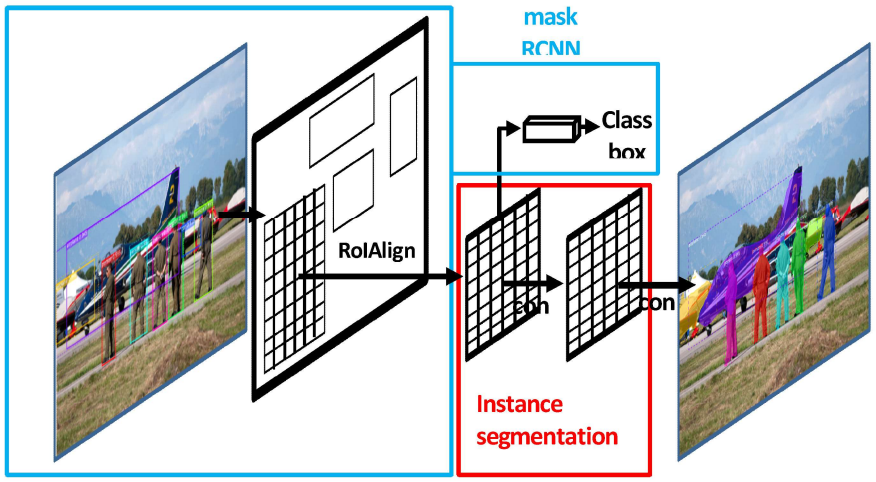

Fig.3.4. Image Pre-Processing Process Using M-RCNN

3.7 Image Pre-Processing

It is the step followed in all neural networks to improve the quality of the input raw data samples. Generally, this step is applied to perform the image denoising process, color conversion, data balance, segmentation, etc. These are the important steps used in the data pre-processing phase to enhance the detection accuracy. Reducing the data dimensionality will help the model produce an exact result. Based on the requirement, the data pre-processing process is performed**.**

3.8 Image Re-Sizing

This paper initially uses gamma adjustment operations and Contrast Limited Adaptive Histogram Equalization (CLAHE) for image processing in the input segmentation dataset. These processed images are then converted into Grayscale and divided into 48×48 patches to use as input. Here, 512×512size input images are deals for dataset segmentation dataset and employ image augmentation techniques, predominantly horizontal and vertical flipping. In this paper, the photos in the input dataset are scaled to a standard dimension of 256 × 256. This paper uses the Adam optimizer to adjust the learning rate and set the epoch number to 100 and the learning rate to 1e−3. To fine-tune the weight and bias scales, this paper sets the normalization factors of the first layer's input set to one and adjusts the layer-wise and connection-wise weight normalization.

3.9 Image Denoising

This paper uses the input dataset where the noise level σ ranges from 0 to 55 for training purposes. Random cropping techniques are used to create patches of 192×256 pixels to augment the data. Grayscale and color images are individually made from these patches. The Adam optimizer is used with an initial learning rate of 1.0×10−5, and the training process works with a batch size of 8. In addition, an early stopping technique is incorporated into the training period, which spans 400 epochs to prevent overfitting. During the fine-tuning phase, the pre-trained denoising models in their spiking versions denoise the input data. During this evaluation, noise was applied to the test images at three different levels (σ ∈ {15, 25, 50}). These noise levels are indicated as σ15, σ25, and σ50, correspondingly. The average of the accumulating membrane potentials is the output treated as the noise map.

3.10 Image Segmentation Using Mask-RCNN

It is the process of dividing the input-colored digital image into several segments. The main focus of this step is to reduce the complexity of the model. In another form, the image segmentation process is performed to label all pixels or elements of the same type. Generally, the image segmentation process is performed using 5 different methods such as threshold-based, edge-based, region-based, clustering-based, and neural network-based segmentation. In this paper, a neural network-based segmentation process is performed to enhance the quality of the input image. The Mask-RCNN model is implemented to segment the pre-processed input image.

3.11 Object Detection

The term object detection deals with locating the objects in the given image and classification of the objects. The SNNs have achieved more remarkable performance than some handmade methodologies by learning semantics deeper and high-level features. SingleShot Multi-box Detector (SSD), a one-step object detection framework, has been developed for embedded real-time applications. By mapping the image pixels, box boundary coordinates are fixed, and classifying the probabilities represents object detection as a classification task. Once the possible probabilities of images are identified, the weight histogram value of each predicted instance is evaluated.

3.12 Weighted Histogram

To estimate the probabilities p_i histogram of weighted events or weighted histogram is used. It is acquired through the random experiment with probability density function g(x), which usually does not match with PDF p (x). For bin I, the sum of weights of events is described as:

$$W_i = \sum_{k=1}^{n_i} w_i(k), \quad \ldots\ldots\ldots\ldots (3)$$

Where $w_i(k)$ is the weight of the k th event in the i th bin and n_i denotes the number of events at bin i . The statistic

$$\hat{p}_i = \frac{W_i}{n} \quad \ldots\ldots\ldots\ldots\ldots (4)$$

is employed to estimate p_i , where n= $\sum_{i=1}^{m} n_i$ denotes the total number of events for the m bins histogram. The event's weight is selected so that the estimate (4) is unbiased,

$$E[\hat{p}_i] = p_i \ldots\ldots\ldots\ldots\ldots (5)$$

3.13 Object Detection Using Spike Neural Network

A backbone and multiple predictor heads are available in the SSD object detection framework. The feature maps generated by the backbone at various scales are given as input to the head to predict the box boundary and associated classes. The spiking neural network, which can perform object detection, can be achieved by interchanging the CNN backbone and normal convolutions in the additional layers with the SNN backbone and spiking convolutions. The feature maps given to the SSD heads are spikes, and the head portion contains only one convolution, making the entire network SNN. The SNN works on 'T' timesteps, resulting in final bounding boxes, and the classes predicted are achieved by the total sum of the 'T' timesteps. We consider the post-processing of output prediction filtrations to be done away from the SNN range in conventional hardware. Excessive predictions, classified as background, make the class imbalance problem. This problem makes the one-shot object detector, SSD, struggle. The one-shot object detector learning can be enhanced with the help of 'focal loss', a term used for cross-entropy loss function. The earlier underperformed SSD, which used hard negative mining, can be overcome by training the network with the focal loss. Three additional convolutions are placed after the spiking backbone, similar to the original SSD, which can generate smaller feature maps. Every block has a 1 × 1 spiking convolution to reduce the number of channels, followed by the 3 ×3 spiking convolution with a sudden increment of 2. The batch normalization is used before every convolution layer.

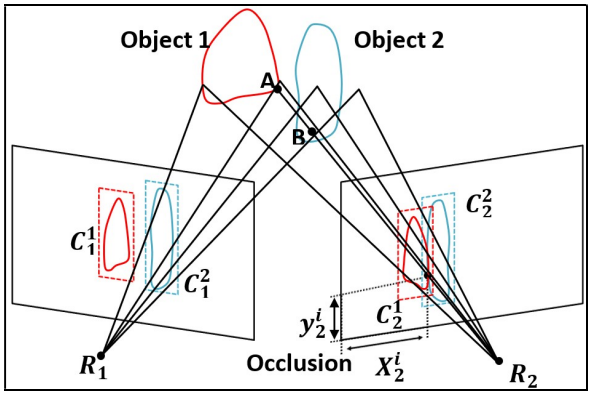

Fig.3.5. Occlusion Estimation in Occluded Image

3.14 Step To Detect Objects From Occlusion Image

The occlusion region between objects A and B is shown in Figure-6. In Figure-6 R_1 and R_2 are the sensors, i, m, and, t denotes the event number, event coordinates, and time, respectively. The terms $e(m_1^i, t)$ and $e(m_2^i, t)$ are the event sensor 1 and 2. The event spatial coordinator is denoted as M. $e(M^i, t)$ represent the 3D event.

C_1^n and C_2^n are the pair of obtained cluster. x_1^i and y_2^i coordination points in R_2. During occlusion, it is impossible to detect the objects. To overcome these limitations, an SNN-based approach is applied in this paper. This proposed model detected all the possible clusters to detect the objects. The main aim of the proposed approach is to detect the occlusion between the objects. It compares the obtained clustering value with the threshold cluster value. If the predicted and actual clusters are the same, then the disparity vector is evaluated, and event matching is observed over several iterations. Otherwise, the cluster match algorithm is applied to pair the uneven clusters in the input. If the model detects a single event candidate, a cluster tracking algorithm reconstructs the 3D clusters. The same function is followed to construct 2D cluster events. Otherwise, the model detects n number of clusters. The same steps followed to the single clusters are applied to each cluster and final occlusion result is detected. Once the positive result is obtained that is, true resultant value is achieved, the total number of valid 3D events among several iterations is evaluated to find the matched one. This clustered evaluation is performed based on the size and position of the objects. If none cluster is identified, the total loop is discarded from the iteration.

4. RESULTS AND DISCUSSIONS

This paper uses Mask R-CNN and SNN models to segment and detect objects in the input Augmented reality (AR) images. The initial step in the object detection is to segment the input image before it is classified into multiple categories. As mentioned above, the segmentation process is performed using a mask R-CNN model. Fig.4.1 depicts the object identified and filtered by the proposed mask-RCNN model. It is clear from Fig.4.1.that the proposed model more efficiently recognizes all the occluded objects in the input image.

Fig.4.1. Occluded Objects

Once the occlude images are filtered, they are boxed and categorized into object types. Thus, the resultant Mask-RCNN-based boxed images are shown in the Fig.4.2. By segmenting the input image, the accuracy of the model is increases. This will lead the mask-RCNN model more accurately boxed region of the objects. Though, the proposed Mask-RCNN model bounding the boxes around the multiple objects in the input samples., it required additional features to specifically boxes the instance. that is, the proposed mask-RCNN model

Fig.4.2. Occlusion Removal

The mask region is generated after generating the bounding box on each class of instance in the input image sample. To generate the mask region, the input (RGB) colored images are converted into shaded images. The extracted mask region from the input data is shown in Fig.4.3. The result represents the four occluded class instances predicted from the input image. These mask shapes are scaled and placed in their respective location. The layer activation process is

performed from the mask generation result to more accurately detect the objects from the samples.

Fig.4.3. Mask Region Detection

Fig.4.4 shows the resultant figures of the layer activation function. This process is entirely performed to further improve the quality of the extracted input image and to reduce the dimensionality. It is obtained by extracting the region of the input images. Through this result, the proposed SNN model more accurately detect the objects with specific identity individually.

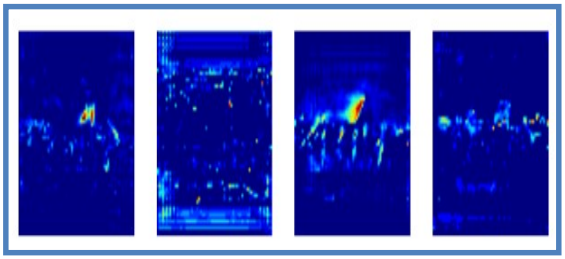

Fig.4.4 Weight Histograms

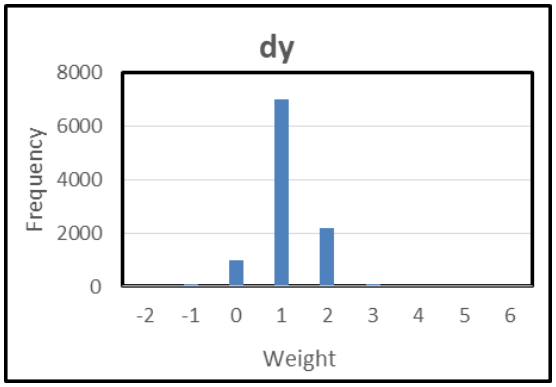

Fig.4.5(a). Weight Histogram For dy

Now, the weighted histogram result of the input image sample is detected and graphically plotted in Fig.4.6. These weight histograms represent the parameters-based loss. For example, **dx** and **dy** are the parameters that represent the spatial plan of the top left and top right bounding boxes. **dW** and **dh** represent width of the bounding box region extracted by the prosed approach. During image segmentation, the training and validation loss rate of different loss component of Mask-RCNN model is evaluated. Fig.4.6(a)shows the overall training and validation class loss rate of Mask-RCNN model. similarly Fig.4.6 (b) and (c) represent the bbox loss and mask loss rate of the MRCNN model.

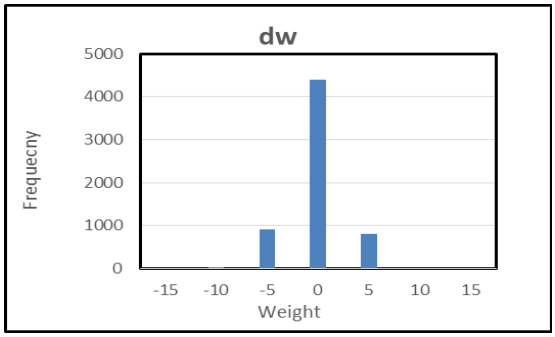

Fig.4.5(b). Weight Histogram For dw

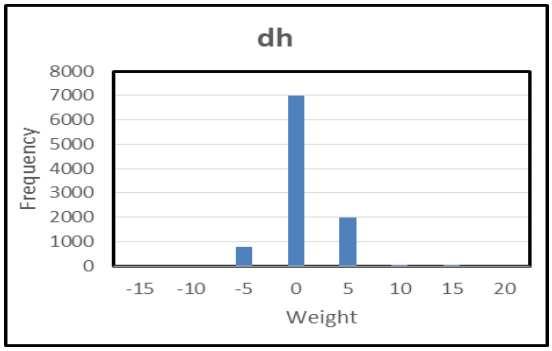

Fig.4.5(c). Weight Histogram For dh

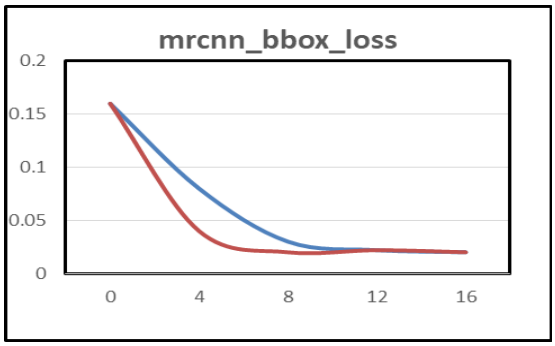

Fig.4.6(a). Loss in Bounding Box

Journal of Theoretical and Applied Information Technology
31st March 2025. Vol.103. No.6

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

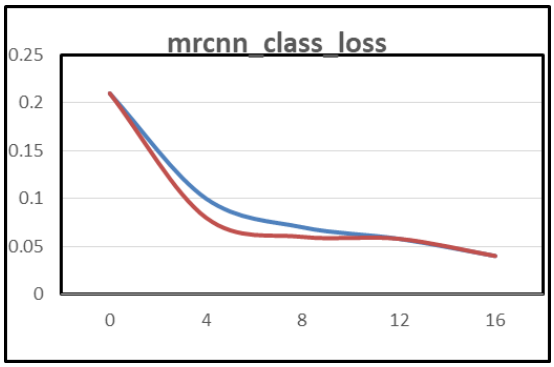

Fig.4.6 (b). Loss In Class

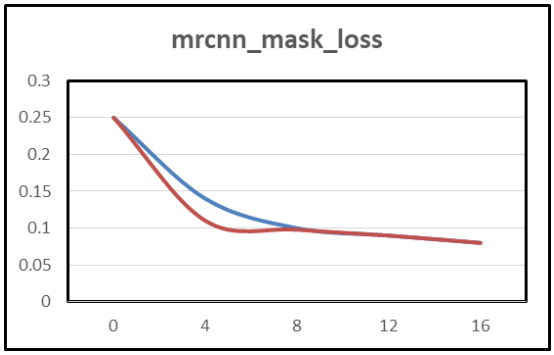

Fig.4.6 (c). Loss in Mask

Fig.4.7. Object Prediction Using SNN

The final prediction result of the proposed SNN model is shown in the Fig.4.7. The proposed model more accurately highlights the various objects in the input AR images. That is, the input images are individually identified based on its respective features. The accuracy result of the proposed model is evaluated and plotted in Fig.4.8

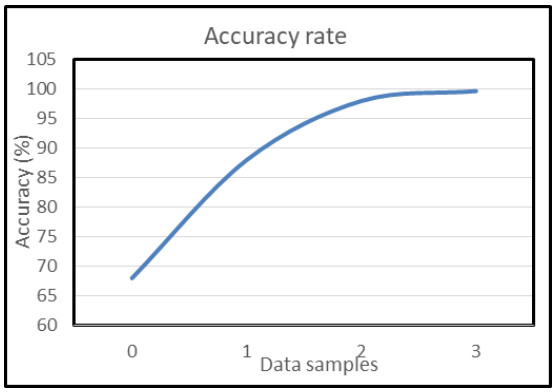

Fig.4.8. Accuracy Result

Fig.4.8indicates the accuracy rate of the model in detecting the objects from the number of samples. The analysis result shows the number of samples increases, and the accuracy rate of the proposed SNN model are also increases. Finally, the proposed model has achieved 99.68% accuracy on detecting objects from the occluded objects in the images. Whereas, Fig.4.9 shows the loss rate of the proposed model on detecting the exact objects from the occluded images. The result of the analysis shows, when number of samples increases, the loss rate of the model is decreases.

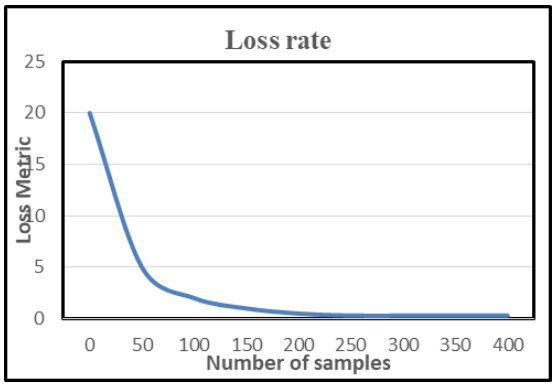

Fig.4.9. Loss Rate

Fig.4.10 shows the accuracy comparison of the proposed and existing models with different time steps. Compared to other methods, the analysis results show that the proposed SNN-based approach consumes less time to reach maximum accuracy. The other traditional methods have consumed more time and achieved less accuracy than the proposed approach. It is clear from the comparison result that the proposed mask-RCNN-based segmentation and SNN-based detection techniques are more suitable for detecting objects from occluded AR images.

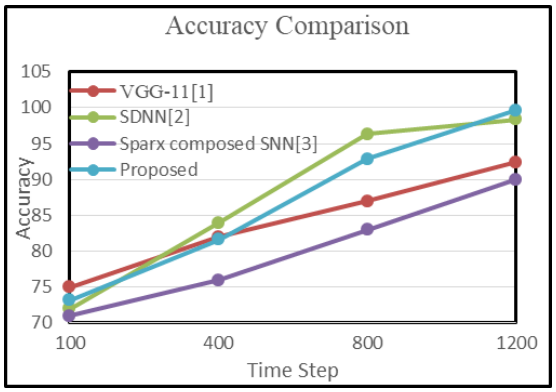

Fig.4.10. Accuracy Comparison

5. CONCLUSION

Occlusion is one of the major problems in object detection and recognition in video images. Earlier research has focused on occlusion detection in face detection and recognition to identify people. Object detection and recognition are also applied in various real-time applications, like virtual reality and augmented reality videos. This paper focused on detecting objects from VR and AR images even by removing occlusions. Hence, it uses various pre-processing methods to initially process the images and multiple neural networks for object detection, recognition, and occlusion removal. It used the Mask-R-CNN model for recurrent object detection concerning masks, detecting the objects, and removing the occlusion. Finally, the SNN detects and recognizes the objects based on their features. From the output, it is noticed that the accuracy of object detection after occlusion removal is high and identified by visualizing the objects and from the histogram values. Finally, the model proposed has reached 99.68% accuracy on the detection of objects from the occluded objects in the images. The proposed model loss rate in finding the precise objects obtained from the occluded images. The analysis results indicate that as the number of samples increases, the model loss rate decreases. Fig 4 (c) shows the comparative accuracy graph of 4 models (VGG-11, SDNN, Sparx Composed SNN and Proposed model) for 100, 400, 800 and 1200 time steps. By default, all models seeded with relatively same neighbouring accuracy values, VGG-11 75% and SDNN lower but improved swiftly, Sparx Composed SNN the least (71%) and Proposed model remained no further distant from SDNN. With time, SDNN outperforms VGG-11 and retains this strong pattern of performance converging to nearly 98% accuracy by the end time step. Meanwhile, Sparx Composed SNN continues its steady improvement, but still brings up the rear, finishing at about 88%. Proposed is also consistently growing and always closely following SDNN attribute, surpassing all the models and achieving the highest accuracy at 1200 time steps, when it reaches ~100%. The vertical comparison means the proposed model is the best model based on accuracy over time as it outperforms the rest on this description.

In future work, real-time VR and AR videos will be recorded from an efficient camera, and this proposed model will verify the real-time performance.

REFERENCES

[1]. Jorge, J., Anjos, R. K. D., & Silva, R. (2019, November). Dynamic occlusion handling for real-time AR applications. In *Proceedings of the 17th International Conference on Virtual-Reality Continuum and its Applications in Industry* (pp. 1-9).

[2]. Macedo, M. C., & Apolinario, A. L. (2021). Occlusion handling in augmented reality: Past, present and future. *IEEE Transactions on Visualization and Computer Graphics*, *29*(2), 1590-1609

[3]. chrome-extension: //efaidnbmnnnibpcajpcglclefindmkaj/https://www.diva-portal.org /smash/get/diva2:1361930/FULLTEXT01.pdf.

[4]. Schmirler, P. D., Nguyen, T. T., Nicoll, A. L., & Vasko, D. (2020). *U.S. Patent No. 10,735,691*. Washington, DC: U.S. Patent and Trademark Office.

[5]. Lopes, D. S., & Jorge, J. A. (2019). Extending medical interfaces towards virtual reality and augmented reality. Annals of Medicine, 51(sup1), 29-29.

[6]. Zorzal, E. R., Sousa, M., Mendes, D., dos Anjos, R. K., Medeiros, D., Paulo, S. F., ... & Lopes, D. S. (2019). Anatomy studio: a tool for virtual dissection through augmented 3D reconstruction. Computers & Graphics, 85, 74-84.

[7]. Herbert, B., Ens, B., Weerasinghe, A., Billinghurst, M., & Wigley, G. (2018). Design considerations for combining augmented reality with intelligent tutors. *Computers & Graphics*, *77*, 166-182.

[8]. Preim, B., & Saalfeld, P. (2018). A survey of virtual human anatomy education systems. *Computers & Graphics*, *71*, 132-153

[9]. Schmidt, S., Bruder, G., & Steinicke, F. (2019). Effects of virtual agent and object

representation on experiencing exhibited artifacts. *Computers & Graphics*, *83*, 1-10.

[10]. Breen, D. E., Rose, E., & Whitaker, R. T. (1995). Interactive occlusion and collision of real and virtual objects in augmented reality. *Munich, Germany, European Computer Industry Research Center*.

[11]. Zhu, J., Pan, Z., Sun, C., & Chen, W. (2010). Handling occlusions in video-based augmented reality using depth information. *Computer Animation and Virtual Worlds*, *21*(5), 509-521

[12]. Du, C., Chen, Y. L., Ye, M., & Ren, L. (2016, September). Edge snapping-based depth enhancement for dynamic occlusion handling in augmented reality. In *2016 IEEE international symposium on mixed and augmented reality (ISMAR)* (pp. 54-62). IEEE.

[13].Fuhrmann, A., Hesina, G., Faure, F., & Gervautz, M. (1999). Occlusion in collaborative augmented environments. *Computers & Graphics*, *23*(6), 809-819.

[14]. Tian, Y., Guan, T., & Wang, C. (2010). Real-time occlusion handling in augmented reality based on an object tracking approach. *Sensors*, *10*(4), 2885-2900.

[15].Ogawa, T., & Mashita, T. (2021, October). Occlusion Handling in Outdoor Augmented Reality using a Combination of Map Data and Instance Segmentation. In *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)* (pp. 246-250). IEEE.

[16].Tian, Y., Guan, T., & Wang, C. (2010). An automatic occlusion handling method in augmented reality. Sensor Review, 30(3), 210-218.

[17].https: // medium. Com / @ aardvarkinfinity/ integrating – augmented – reality – with - neuromorphic-computing – for – enhanced – cognitive –training -75e29b3cfc8e

[18].Wei, M., Tang, J., Tang, H., Zhao, R., Gai, X., & Lin, R. (2021). Adoption of convolutional neural network algorithm combined with augmented reality in building data visualization and intelligent detection. Complexity, 2021, 1-13.

[19].Sarkar, S., Venugopalan, V., Reddy, K., Ryde, J., Jaitly, N., & Giering, M. (2017). Deep learning for automated occlusion edge detection in RGB-D frames. Journal of Signal Processing Systems, 88, 205-217.

[20].Kang, B., & Druckmann, S. (2020). Object recognition under occlusion revisited: elucidating algorithmic advantages of recurrent computation. bioRxiv, 2020-12.

[21].Polykretis, I., Tang, G., & Michmizos, K. P. (2020, July). An astrocyte-modulated neuromorphic central pattern generator for hexapod robot locomotion on intel's loihi. In International Conference on Neuromorphic Systems 2020 (pp. 1-9).

[22].Irizarry-Valle, Y., & Parker, A. C. (2015). An astrocyte neuromorphic circuit that influences neuronal phase synchrony. IEEE transactions on biomedical circuits and systems, 9(2), 175-187.

[23].Orchard, G., Jayawant, A., Cohen, G. K., & Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. Frontiers in neuroscience, 9, 159859.

[24].Shastri, B. J., Tait, A. N., Ferreira de Lima, T., Pernice, W. H., Bhaskaran, H., Wright, C. D., & Prucnal, P. R. (2021). Photonics for artificial intelligence and neuromorphic computing. Nature Photonics, 15(2), 102-114.

[25].Yin, S., Venkataramanaiah, S. K., Chen, G. K., Krishnamurthy, R., Cao, Y., Chakrabarti, C., & Seo, J. S. (2017, October). Algorithm and hardware design of discrete-time spiking neural networks based on backpropagation with binary activations. In 2017 IEEE Biomedical Circuits and Systems Conference (BioCAS) (pp. 1-5). IEEE.

[26].Dunleavy, M., & Dede, C. (2014). Augmented reality teaching and learning. Handbook of research on educational communications and technology, 735-745.

[27].Arena, F., Collotta, M., Pau, G., & Termine, F. (2022). An overview of augmented reality. Computers, 11(2), 28.

[28].Macedo, M. C., & Apolinario, A. L. (2021). Occlusion handling in augmented reality: Past, present and future. IEEE Transactions on Visualization and Computer Graphics, 29(2), 1590-1609.

[29].Hebborn, A. K., Höhner, N., & Müller, S. (2017, October). Occlusion matting: realistic occlusion handling for augmented reality applications. In 2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (pp. 62-71). IEEE.

[30] Jorge, J., Anjos, R. K. D., & Silva, R. (2019, November). Dynamic occlusion handling for real-time AR applications. In Proceedings of the 17th International Conference on Virtual-

Reality Continuum and its Applications in Industry (pp. 1-9).
[31] R. Girshick. Fast R-CNN. In ICCV, 2015.
[32] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).
[33] Dai, J., He, K., & Sun, J. (2016). Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3150-3158).
[34]. Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. *Advances in neural information processing systems*, *28*.