Faster R-C.N.N: Towards Real-Time Oject Recognition

Ribhav Sabharwal*, Shivam Jaiswal*, Saksham Tyagi, Samarth Jaiswal, Vaishali Deshwal

MEERUT INSTITUTE OF ENGINEERING AND TECHNOLOGY

***Abstract:* - Futuristic object recognition grids are based on regional focusing algorithms to generate hypotheses about the position of objects. Advances such as Fast R-C.N.N [5] and SPPnet [7] has shortened the execution times of these recognition networks and show that region focus computation is an obstacle. In this research, a Region Proposal Network (RPN) is proposed that has full convolution functionality with the recognition grid and permits quasi-free region applications. The RPN is a completely convolved grid that simultaneously forecasts object boundaries and objectivity values at each location. The RPN continually trains to create best-in-class regional proposals utilized by Fast R-C.N.N used for recognition. A simple alternative enhancement allows us to train Fast R-C.N.N and RPN to exhibit complicated qualities. The recognition engine ran at 5.0 fps (counting all phases) on the GPU for the very deep model VGG-16 [19], PASCAL VOC 2007 (73.19% mAP) and 2012 (70.41% mAP) at 300 frame rate.**

Keywords – Machine Learning, Fast R-CNN, Region Proposed Networks, grids, Recognition

1. INTRODUCTION

The advancement of region-based approaches like the one proposed in [22] and the development of region-based complex neural networks, namely R-C.N.N [6], have been the guiding force behind current progress in the field of object recognition. While building region-based convolutional neural networks was computationally expensive [6], the sharing of convolutions among proposals led to a significant reduction in costs [7, 5]. The up-to-date version of fast R-C.N.N [5] accomplishes near real-time performance by leveraging very deep grids [19] and circumventing the computational overhead incurred by local proposals. Thus, in modern recognition systems, the proposal stage is no longer a bottleneck for computation.

The methods for suggesting regions are typically founded on inexpensive characteristics and logical reasoning processes. Among these methods, Selective Search (SS) [22] is widely utilized and works by merging super pixels in a greedy manner based on low-level features. However, in CPU implementations, Selective Search is significantly slower, taking about 2 seconds per frame, compared to efficient recognition networks [5]. On the other hand, Edge Box [24], which takes approximately 0.2 seconds per frame, currently delivers the best balance between value and speed for region suggestion. Nevertheless, the time taken for region suggestion is still as much as that required by the recognition network.

The speedy region-based C.N.Ns commonly utilize Graphical Processing Units, whereas the region suggestion approaches utilized in our research rely on CPUs. Hence, comparing the runtimes of these methods would be unjust. One way to enhance the efficiency of our proposed approach is by adapting it for GPUs, which can be a viable technical solution. However, this re-implementation overlooks the computational trade-offs that are significant for downstream sensor networks.

In this article, we demonstrate that the modified deep mesh algorithm offers a sophisticated and effective approach to computing the reconnaissance network calculation, which was previously challenging to accomplish. To achieve this, we present a novel RPN that utilizes convolutional levels and advanced object recognition grids [7, 5]. This approach evenly distributes the computational load during testing, resulting in a negligible cost (e.g., only 10.0 ms per frame) for computing the proposal.

Our remark is that regions suggestions can also be generated by Convolutional Attribute (Conv) maps and implemented in region-based. In addition to these conversion functions, we add two more conversion layers to create the RPN. One encodes each location in the conv-map into a brief feature vector (e.g.., 256 - d) and the second outputs the object at every location in the conv-map. Scores and regression limit for k-region nominations on that location associated with different scope and aspect ratios (k = 9.0 is a standard value).

We can use a Fully Convolutional Network (FCN) type to train end-to-end for producing recognition cues, and recommend integrating RPN into R-C.N.N networks for fast object recognition [5]. For this purpose, we suggest a simple training structure that alternates between region proposal task matching and object recognition matching, while keeping the proposal unchanged. This approach quickly leads to the creation of an integrated network with shared conversion functionality for both processes.

Comparing to the PASCAL repository for VOC recognition [4], the RPN integrated with fast R-C.N.N achieves superior recognition accuracy than the selective search's robust base in combination with fast R-C.N.N. On the other hand, this procedure removes almost all the computational load of SS during testing. The execution time of the proposal is only 10ms. Using the exclusive very deep model from [19], the recognition method maintains a 5.0 fps (consisting of all steps) on the GPU, making it a real-world object recognition system in terms of swiftness and precision. (73).

2. RELATED WORK

Several recent studies have proposed methods to find specific or class-independent bounding boxes using deep networks [21, 18, 3, 20]. A fully associated (fc) layer is built by the OverFeat technique [18] to predict frame coordinates for localization tasks performed by a sole object. A conv layer is then obtained from a fc layer through conversion to recognize various objects of a particular class. Multi-box approach is used for object recognition in deep neural networks by producing region suggestions with a final fully connected layer that predicts multiple boxes at once, which can number up to 800. This process is often implemented by the R-C.N.N system. The proposed grid is applied to a sole image or to several large image sections (e.g.., 225,225) [20]. MultiBox and OverFeat are discussed in increased detail further in the context of methods.

The calculation of curved joints [18, 7, 2, 5] attracts more and more attention for effective and precise visual identification. The OverFeat article [18] calculates conversion attributes from image for organization, locating and recognition. Size compatible clustering (SPP) [7] on collective conversion characteristic maps has been expected for effective region-based object recognition [7, 16] and semantic distribution [2]. End-to-end indicator training is empowered by Fast R-C.N.N [5] for common conversion functions and impresses with its precision and swiftness [25][26][27].

3. PROPOSED METHODOLOGY

3.1 REGION PROPOSED NETWORKS

The purpose of the RPN is to produce object proposals along with their respective objectness scores from an input image, irrespective of its dimensions. To achieve fast object recognition with the R-C.N.N network [5], we suppose that both networks use a mutual set of convolutional layers. During our researches, we evaluated the shareability of four convolutional layers in the Z.F. model by Zeiler and Fergus [23] and twelve convolutional layers in the VGG model by Simonyan and Zisserman [19] [28].

To produce proposals, we overlay a minor grid on the product of the last shared convolutional layer and leverage a fully connected network to predict proposal scores. This is accomplished by connecting a spatial window of size m x m from the convolutional feature map of the input to the network [29].

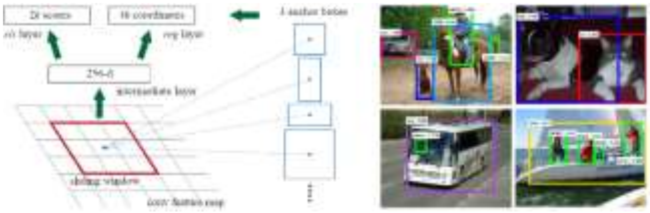


Figure 1
Leftward: Regional Proposal Network (RPN)
Rightward: Examples of recognitions using RPN statements in the PASCAL VOC test 2007

The process involves associating each sliding window with a low-dimensional vector, which has a dimension of 256 for ZF and 512 for VGG. These vectors are then inserted into two fully connected layers: a box classification layer (referred to as ".cls") and a d box regression layer (referred to as ".reg"). The effective field of view of the input picture is quite huge, measuring 173 pixels and 229 pixels for IF and VGG, respectively, when n = 3.0. This mini-grid, which functions like a S.W (Sliding Window), spreads the fully associated layers across all spatial positions, as depicted in Figure 1 (on the left). To achieve this, a Conv m X m layer is utilized, alongside two 1.0 X 1.0 Conv sibling layers (for. cls and .reg, correspondingly). The output of the m X m-conv layer is passed through a ReLU[15] activation function [30].

Transformation invariant hook

The RPN and MultiBox techniques are two methods for generating proposals in object detection. The RPN predicts k proposed regions at every position of the sliding window, resulting in 4k outputs from the reg layer in which the coordinates of k-boxes are encoded. The cls layer generates 2000 probabilities for each statement to estimate the object/non-object probability. Hooks are used to restrict the proposals with respect to k reference boxes, and with 4 scales and 4 aspect ratios, each sliding window position has k=16 hooks. The RPN is translation invariant for both the hooks and the function that computes the hints for the hooks, which is a significant advantage [31].

In comparison, the MultiBox technique [20] generates 800 non-translatable hooks using k-means. As the declaration must be converted when translating an object to a picture, the MultiBox hook necessitates a $[[(4) + (1)] \times 800]$ dimensional resultant layer, whereas the RPN requires only one $[(4 + 2) \times 9]$ dimensional resultant layer. This results in fewer parameters and reduces the risk of over-fitting small databases similar to PASCAL VOC [32].

A loss function for training sample suggestions

To train the Regional Proposal Network (RPN), hooks are assigned binary class labels indicating whether they correspond to an object or not. Positive labels are allocated to two sorts of hooks: those with high intersection with ground truth boxes and the hook with the highest intersection for each ground truth box. If an hook has low intersection with all ground truth boxes (IoU score < 0.3), it is assigned a negative label. Hooks that do not fall into the positive or negative categories are ignored during training [33][34].

We then minimize the multitasking loss objective function as in Fast R-C.N.N [5], which includes a regression loss and a classification loss as follows [35][36].

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{\text{reg}}}\sum_i p_i^* L_{\text{reg}}(t_i, t_i^*).$$

In the Region Proposal Network (RPN), the hook index i and its corresponding predicted probability pi play an important role. A positive hook is assigned a ground truth label pi* of 1, while a negative hook has a label of 0. The predicted bounding box coordinates are represented by vector ti, and ti* corresponds to the coordinates for the positive hook. The classification loss, Lcls, uses the log loss function. The regression loss, Lreg(ti; ti*), is computed using the robust loss function R defined in [5] for the parameterized bounding box coordinates. The outcomes of the classification and regression layers, {pi} and {ti}, are regularized by Ncls and Nreg, respectively, along with a balance parameter λ.3.For the regression, we assume a four-coordinate parametrization according to [6] [37][38].

$$t_x = (x - x_a)/w_a, \quad t_y = (y - y_a)/h_a, \quad t_w = \log(w/w_a), \quad t_h = \log(h/h_a)$$
$$t_x^* = (x^* - x_a)/w_a, \quad t_y^* = (y^* - y_a)/h_a, \quad t_w^* = \log(w^*/w_a), \quad t_h^* = \log(h^*/h_a)$$

The prediction, hook, and ground truth values of a box are calculated using the center coordinates (x, y), width (w), and height (h) of the box [39][40].

While previous methods for bounding box regression relied on feature maps [7, 5] and clustered features of regions of varying sizes, our approach differs in that it utilizes features with the identical spatial size (m x m) in the charcteristic map for regression. A set of k bounding box regressors is trained, with each regressor accounting for different sizes and aspect ratios. Unlike previous methods, these regressors have unique weights, enabling the prediction of boxes of varying sizes even when feature size and scaling are constant [41].

Improvement

The fully convolutional mesh implementation can be used to train the RPN end-to-end through stochastic gradient descent (SGD) and backpropagation in a passive voice construction [12]. In line with the image-centric sampling approach proposed in [5], each mini-stack consists of positive and negative hooks randomly selected from an image. Instead of optimizing the loss function for all hooks, a subset of 256 hooks is randomly chosen from the picture to compute the ministack loss function. The negative and positive samples have a maximum ratio of 1:1, and if an image has less than 128 positive samples, the ministack is complemented with negatives [42][43].

New layers are initialized by drawing weights randomly from a Gaussian distribution with a S.D. [Standard Deviation] of 0.009. The common conversion layers and all other layers are prepared using a model trained for ImageNet categorization [17], similar to the standard method [6]. To conserve memory, layers from network IF and conv3 1 or higher are placed on the VGG network [5]. The PASCAL dataset is used for the first 60,000 mini-batches with a learning degree of 0.001, followed by a learning degree of 0.0001 for the next 20,000 mini-batches [44].

Sharing of convolution functions for area suggestions and object recognition

Our previous efforts have been directed towards training a network that generates region proposals without considering C.N.Ns for object recognition based on those proposals. However, to use these proposals, it is essential to incorporate C.N.Ns. In our approach, we utilize Fast R-C.N.N [5] as our recognition network.

In order to accomplish this goal, it is necessary to acquire knowledge of the transformation layers that are common to both the RPN and Fast R-CNN networks, which have undergone separate training. However, integrating these two networks into a single entity and optimizing it using backpropagation is a complex undertaking, as the process of training Fast R-CNN involves utilizing predetermined object proposals, and it is unclear whether modifying the proposal mechanism would lead to successful convergence of the Fast R-CNN training. While joint optimization is a promising topic for future research, we have devised a practical four-stage training algorithm that enables us to learn joint features by alternating between optimizing RPN and Fast R-C.N.N.

The Region Proposal Network (RPN) is trained using a two-step process. Initially, the RPN is initialized with a pre-trained ImageNet model and then fine-tuned to improve its performance for the region proposal task. In the second step, the RPN generates proposals which are then used to train a Fast R-C.N.N detection network. The ImageNet model, which has been pre-trained, is used to initialize the detection network, but it doesn't share any layers with the RPN at this stage. In the third step, only the RPN-specific layers are optimized using the detection network while keeping the shared layers between the two networks fixed. Once the RPN is optimized, the shared layers are jointly fine-tuned, and the Fast R-C.N.N fully connected layer is optimized.

Implementation Details

To conduct our experiments, we trained and evaluated our proposed object recognition and region networks on images of a fixed scale of [7, 5]. To ensure consistency, we resized the pictures so that the shorter side was set to s = 600 pixels. Although using multiple scales to extract features can enhance accuracy, we found that this approach resulted in a tradeoff between speed and accuracy that was not ideal. It is important to mention that in the case of the ZF and VGG neural network models, the cumulative stride value at the concluding convolutional layer is 16 pixels when applied to a scaled image, while for a standard PASCAL image with a resolution of approximately 500x375 pixels, the stride value is approximately 10 pixels. While larger strides can still produce good results, smaller strides can improve accuracy.

Our proposed algorithm employs hooks with three scales that have frame areas of 5121, 2561, and 1281 pixels and three aspect ratios of 2:1, 1:2, and 1:1. Upon analysis, we observed that our algorithm has the ability to leverage hooks with larger frames than the underlying receptive field to generate predictions for larger proposals, even if the object's centre is the only visible portion. This is due to the fact that such predictions are not deemed impossible, we can still roughly estimate the extent of the object. By using this design, our solution avoids the need for multiscale features or multiscale sliding windows to predict large regions, resulting in significant runtime savings. As shown in Figure 1 (right), Our approach achieves exceptional results on various scales and aspect ratios. Furthermore, we have included a table presenting the mean proposal size for each hook acquired through training with the ZF mesh at a resolution of 600 pixels.

anchor	128^2, 2:1	128^2, 1:1	128^2, 1:2	256^2, 2:1	256^2, 1:1	256^2, 1:2	512^2, 2:1	512^2, 1:1	512^2, 1:2
proposal	188 × 111	113 × 114	70 × 92	416 × 229	261 × 284	174 × 332	768 × 437	499 × 501	355 × 715

It is important to handle hook boxes that extend beyond the image boundaries with caution to avoid stalling during training. During training, such hooks should be ignored to prevent them from contributing to the error term and hindering convergence. In a typical 1000x600 image, there are approximately 20,000 hooks, of which only about 6,000 are used for training after excluding transboundary outliers. During testing, The complete convolutional Region Proposal Network (RPN) is utilized to generate proposal boxes that cover the entire image and are clipped at the edges of the image.

In order to handle significantly duplicated Region Proposal Network (RPN) proposals, the technique of non-maximum suppression (NMS) is utilized, where the classification score is considered. By setting the NMS IoU threshold at 0.7, about 2,000 potential regions are identified per image, leading to a reduction in proposal numbers without any adverse impact on the ultimate recognition accuracy. The recognition process involves the selection of the highest ranked N candidate regions, and a Fast R-CNN is subsequently trained, utilizing 2,000 RPN proposals, with different numbers of proposals assessed during the testing phase.

4. EXPERIMENTS

To evaluate the efficiency of our suggested approach, we carried out experiments on the extensively utilized recognition benchmark of 2007 PASCAL VOC in our study [4]. This dataset comprises of over 10,000 images divided into training and test sets, and features more than 21 object categories. Furthermore, to evaluate the effectiveness of our approach, we conducted experiments on the widely-used 2012 PASCAL VOC benchmark and presented the results for several baseline models.

To conduct our experiments, we employed two pre-trained models trained on the ImageNet dataset, namely Z.F net [23] and VGG-165 [19]. The Z.F net model comprises six convolutional layers and four fully connected layers, while the VGG-165 model consists of 13 convolutional layers and three fully connected layers.

We assessed the efficacy of our approach mainly by utilizing the mean average precision (mAP) metric, which is commonly employed to evaluate object recognition systems. Our findings indicate the efficacy of the proposed method and offer valuable perspectives into the performance of various models and architectures on the PASCAL VOC benchmark.

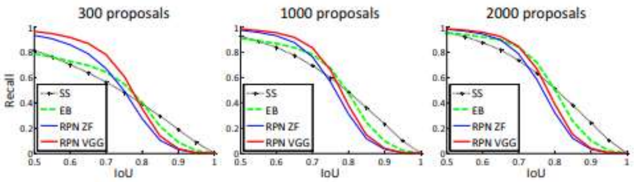
Table 1: The performance evaluation of Fast R-CNN and VGG16

method	# proposals	data	mAP (%)	time (ms)
SS	2k	07	66.9†	1830
SS	2k	07+12	70.0	1830
RPN+VGG, unshared	300	07	68.5	342
RPN+VGG, shared	300	07	69.9	**198**
RPN+VGG, shared	300	07+12	**73.2**	**198**

The results were carried out on the PASCAL VOC 2007 assessment set through recognition experiments. The training data comprised two sets, namely "07" and "07+12", referring to VOC 2007 train.val and the combined VOC 2007 train.val and VOC 2012 train.val datasets, respectively. The train time proposed by RPN using Fast R-C.N.N was 2k. Furthermore, the symbol "†" indicates that the value specified was mentioned in [5], and according to the benchmark provided in this article, the number is higher, with an average of 68.01 ± 0.32 obtained in six run [45][46].

Recognition accuracy and operating time of the VGG-16. The results of VGG-16 for proposal and recognition are presented in Table 1. When VGG is combined with RPN, the Fast R-CNN achieves a non-shared function accuracy of 68.49%, which is comparatively superior to the SS baseline. According to the table, the higher precision of the assumption generated by VGG and RPN compared to SS can be attributed to the rigorous training of RPN and the improved networking. The collective feature modification resulted in a superior outcome of 69.89% compared to SS's baseline, even with almost free offerings. To enhance the performance of the RPN and recognition grid, we conducted additional training on the junction dataset extracted from the combined train.val sets of PASCAL VOC 2007 and 2012, following [5]. PAM is 73.18%. According to the results presented in Table 3 of the PASCAL VOC 2012 trial set, our technique demonstrates a mean average precision (mAP) of 71.02%. This performance metric was obtained after training our approach on the unification set, which comprises the VOC 2007 train, validation, and test sets, along with the VOC 2012 train and validation sets. This performance surpasses that achieved by the method described in reference [5] [47][48].

IoU callback parsing. The next step involves calculating the recall of propositions across various IoU fractions in comparison to the ground truth cases. However, it is important to keep in mind that the Recall-to-IoU metric is not a precise measure of recognition accuracy, as its correlation with recognition precision is limited and approximate [8, 7, 1]. Using this metric to identify the proposal technique is more accurate than calculating it.



One-Stage Recognition vs. Two-Stage Proposal + Recognition.

The method proposed in the OverFeat paper [18] utilizes regressors and classifiers to detect objects in input feature maps by sliding windows. In contrast, our approach consists of a two-stage cascade that involves separate classification and object-specific recognition. While OverFeat uses a scaled pyramid with aspect ratio sliders to simultaneously determine the location and category of objects, our RPN method employs square sliders (3x3) to suggest regions with different aspect ratios and scales. However, although both methods use sliding windows, our RPN + Fast R-C.N.N cascade is more comprehensive, as it involves a second stage where the suggested regions are refined to improve object detection. In this step, regional qualities are adaptively clustered from suggested regions, leading to more accurate object recognition. To compare single-stage and two-stage systems, we simulated the OverFeat system using the single-stage Fast R-C.N.N, eliminating potential differences in implementation details. Our results indicate that the single-stage system produces a mean average precision (mAP) of 53.9%, which is 4.8% lower than the two-stage system's 58.7% achieved with the ZF model. These findings demonstrate the effectiveness of using a cascade of region proposals and object recognition to improve object detection, as previous studies have also reported. Additionally, the one-step system is slower due to the higher number of suggestions that need to be processed.

5. CONCLUSION AND FUTURE WORK

The Regional Proposal Network (RPN) was developed by us as a means of generating regional proposals in a highly efficient and precise manner. By sharing its complex function with the downstream recognition grids, by making the region suggestion step highly efficient, it is now possible for a deep learning-powered object recognition system to operate at an impressive speed ranging between 5 to 17 frames per second, without incurring significant computational costs. Furthermore, the implementation of the RPN improves the quality of the region proposals, the implementation resulted in a notable improvement in the precision of detecting objects. In summary, the RPN is a valuable tool that enables fast and accurate object recognition through the efficient generation of high-quality region proposals.

Real-time object detection is a crucial technology that offers numerous benefits and promising applications in diverse fields. A research paper on this topic can emphasize several advantages, such as enhanced safety and increased efficiency. For example, real-time object detection can improve safety by identifying and tracking objects in real-time, which is particularly relevant in the automotive industry to prevent collisions between vehicles. Additionally, object detection can enhance efficiency by detecting defects in real-time during manufacturing processes, leading to higher product quality and less waste. Overall, real-time object detection has the potential to revolutionize multiple industries, making it a technology worth exploring further.

References

[1] N. Chavali, H. Agrawal, A. Mahendru, and D. Batra. Object-Proposal Evaluation Protocol is 'Gameable'. arXiv: 1505.05836, 2015.

[2] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In CVPR, 2015.

[3] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object recognition using deep neural networks. In CVPR, 2014.

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.

[5] R. Girshick. Fast R-C.N.N. arXiv:1504.08083, 2015.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object recognition and semantic segmentation. In CVPR, 2014.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV. 2014.

[8] J. Hosang, R. Benenson, P. Doll´ar, and B. Schiele. What makes for effective recognition proposals? arXiv:1502.05082, 2015.

[9] J. Hosang, R. Benenson, and B. Schiele. How good are recognition proposals, really? In BMVC, 2014.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. arXiv:1408.5093, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[12] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard,W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1989.

[13] K. Lenc and A. Vedaldi. R-C.N.N minus R. arXiv:1506.06981, 2015.

[14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.

[15] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In ICML, 2010.

[16] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun. Object recognition networks on convolutional feature maps. arXiv:1504.06066, 2015.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575, 2014.

[18] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and recognition using convolutional networks. In ICLR, 2014.

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[20] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object recognition. arXiv:1412.1441v2, 2015.

[21] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object recognition. In NIPS, 2013.

[22] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A.W. Smeulders. Selective search for object recognition. IJCV, 2013.

[23] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In ECCV, 2014.

[24] C. L. Zitnick and P. Doll´ar. Edge boxes: Locating object proposals from edges. In ECCV, 2014.

[25] Pramanik, Sabyasachi, et al. "A novel approach using steganography and cryptography in business intelligence." Integration Challenges for Analytics, Business Intelligence, and Data Mining. IGI Global, 2021. 192-217.

[26] Narayan, Vipul, et al. "FuzzyNet: Medical Image Classification based on GLCM Texture Feature." 2023 International Conference on Artificial Intelligence and Smart Communication (AISC). IEEE, 2023.

[27] Narayan, Vipul, et al. "Deep Learning Approaches for Human Gait Recognition: A Review." 2023 International Conference on Artificial Intelligence and Smart Communication (AISC). IEEE, 2023.

[28] Mall, Pawan Kumar, et al. "FuzzyNet-Based Modelling Smart Traffic System in Smart Cities Using Deep Learning Models." Handbook of Research on Data-Driven Mathematical Modeling in Smart Cities. IGI Global, 2023. 76-95.

[29] Mall, Pawan Kumar, et al. "Early Warning Signs Of Parkinson's Disease Prediction Using Machine Learning Technique." Journal of Pharmaceutical Negative Results (2022): 4784-4792.

[30] Srivastava, Swapnita, et al. "An Ensemble Learning Approach For Chronic Kidney Disease Classification." Journal of Pharmaceutical Negative Results (2022): 2401-2409.

[31] Sawhney, Rahul, et al. "A comparative assessment of artificial intelligence models used for early prediction and evaluation of chronic kidney disease." Decision Analytics Journal 6 (2023): 100169.

[32]Paricherla, Mutyalaiah, et al. "Towards Development of Machine Learning Framework for Enhancing Security in Internet of Things." Security and Communication Networks 2022 (2022).

[33]Tyagi, Lalit Kumar, et al. "Energy Efficient Routing Protocol Using Next Cluster Head Selection Process In Two-Level Hierarchy For Wireless Sensor Network." Journal of Pharmaceutical Negative Results (2023): 665-676.

[34]Narayan, Vipul, A. K. Daniel, and Pooja Chaturvedi. "E-FEERP: Enhanced Fuzzy based Energy Efficient Routing Protocol for Wireless Sensor Network." Wireless Personal Communications (2023): 1-28.

[35 NARAYAN, VIPUL, A. K. Daniel, and Pooja Chaturvedi. "FGWOA: An Efficient Heuristic for Cluster Head Selection in WSN using Fuzzy based Grey Wolf Optimization Algorithm." (2022).

[36] Faiz, Mohammad, et al. "IMPROVED HOMOMORPHIC ENCRYPTION FOR SECURITY IN CLOUD USING PARTICLE SWARM OPTIMIZATION." Journal of Pharmaceutical Negative Results (2022): 4761-4771.

[37] Babu, S. Z., et al. "Abridgement of Business Data Drilling with the Natural Selection and Recasting Breakthrough: Drill Data With GA." Authors Profile Tarun Danti Dey is doing Bachelor in LAW from Chittagong Independent University, Bangladesh. Her research discipline is business intelligence, LAW, and Computational thinking. She has done 3 (2020).

[38] Narayan, Vipul, et al. "Enhance-Net: An Approach to Boost the Performance of Deep Learning Model Based on Real-Time Medical Images." Journal of Sensors 2023 (2023).

[39] Ojha, Rudra Pratap, et al. "Global stability of dynamic model for worm propagation in wireless sensor network." Proceeding of International Conference on Intelligent Communication, Control and Devices: ICICCD 2016. Springer Singapore, 2017.

[40] Shashank, Awasthi, et al. "Stability analysis of SITR model and non linear dynamics in wireless sensor network." Indian Journal of Science and Technology 9.28 (2016).

[41] Gupta, Sandeep, Arun Pratap Srivastava, and Shashank Awasthi. "Fast and effective searches of personal names in an international environment." Int J Innov Res Eng Manag 1 (2014).

[42]Awasthi, Shashank, Naresh Kumar, and

Pramod Kumar Srivastava. "An epidemic model to analyze the dynamics of malware propagation in rechargeable wireless sensor network." Journal of Discrete Mathematical Sciences and Cryptography 24.5 (2021): 1529-1543.

[43]Tyagi, Neha, et al. "Data Science: Concern for Credit Card Scam with Artificial Intelligence." Cyber Security in Intelligent Computing and Communications. Singapore: Springer Singapore, 2022. 115-128.

[44] Srivastava, Arun Pratap, et al. "Fingerprint recognition system using MATLAB." 2019 International conference on automation, computational and technology management (ICACTM). IEEE, 2019.

[45] Kumar, Neeraj, et al. "Parameter aware utility proportional fairness scheduling technique in a communication network." International Journal of Innovative Computing and Applications 12.2-3 (2021): 98-107.

[46] Awasthi, Shashank, et al. "A New Alzheimer's Disease Classification Technique from Brain MRI images." 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM). IEEE, 2020.

[47] Awasthi, Shashank, et al. "Modified indel treatment for accurate Phylogenetic Tree construction." 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM). IEEE, 2020.

[48] Mohseni, S., Yang, F., Pentyala, S., Du, M., Liu, Y., Lupfer, N., ... & Ragan, E. (2021, May). Machine learning explanations to prevent overtrust in fake news detection. In Proceedings of the International AAAI Conference on Web and Social Media (Vol. 15, pp. 421-431).