RESEARCH ARTICLE OPEN ACCESS

Pedestrian Detection in Images using YOLO

Mana Teja Sharma
*(Computer Science Department, SRM University/India
Email: manatejasharma@gmail.com)

Abstract:

To ensure the safety of both occupants within a vehicle and individuals outside it, it is crucial for the car's embedded sensors to accurately detect objects in its vicinity. The utilization of Deep Convolutional Neural Network (DCNN) architectures has led to the development of more efficient real-time object detection methods. This study focuses on evaluating various Deep CNN models using images captured by integrated fisheye cameras to precisely locate pedestrians in close proximity to the vehicle. The research involves conducting experiments using an acquired image dataset from a test vehicle equipped with multiple fisheye cameras. Different Deep CNN models, including Tiny-YOLOv3, YOLOv3, and ND ResNet-50, were chosen for pedestrian localization. These selected models were trained and evaluated on the collected fisheye image dataset. Furthermore, a comparison was made with pre-trained CNN architectures designed to localize pedestrians in standard non-fisheye images. The experimental findings highlight that YOLOv3 achieves the most superior performance, while ResNet struggles to accurately detect and localize pedestrians on roadways..

Keywords — **Computer vision, pedestrian detection, YOLO, machine learning.**

I. INTRODUCTION

The advancement of autonomous vehicle technology has been on a steady rise, with projections indicating that a substantial portion of vehicles on the road will achieve full autonomy by 2035 [1]. As the journey towards complete autonomy unfolds, present-day vehicles are equipped with Advanced Driver Assistance Systems (ADAS). Within the realm of autonomous vehicles, ADAS plays a pivotal role, encompassing a spectrum of applications from critical safety functions to less intricate parking systems [2], [3]. An integral aspect of driver support systems is the detection of both static and dynamic objects to avert collisions. This emphasis on collision prevention ensures the safety and security of both vehicle occupants and pedestrians, a paramount goal over the years. Conventional autonomous vehicles often employ a range of sensors, including RADAR (Radio Detection and Ranging), LiDAR (Light Detection and Ranging), and optical cameras, to detect and localize objects. However, utilizing all these sensors for less critical scenarios, such as parking or low-speed situations, might not be cost-effective. Despite a decline in the cost of sensors like LiDAR and RADAR, they remain relatively expensive [4]. Therefore, in such cases, the integration of cost-effective sensors like Cameras and UltraSonic sensors becomes more pragmatic. In situations where cost is a consideration, 2D vision sensors like pinhole cameras or fisheye cameras can be effectively employed for object detection [5]. Fisheye cameras, in particular, provide a wider field of view around the vehicle compared to pinhole cameras. Consequently, fisheye cameras find application in systems like surround view setups [6], [7], [8]. The evolution of Deep Convolutional Neural Networks (CNNs) has significantly enhanced real-time object detection accuracy. Notably, end-to-end learning approaches have revolutionized performance, demanding minimal training in the domains of image analysis and computer vision [9-11]. To further optimize the computational efficiency of generated CNN architectures, tools like TensorRT can be leveraged, facilitating real-time object detection across all car cameras on a single computing resource. Traditionally, researchers have delved into handcrafted feature learning methods. Commonly utilized features include Haar features, Histogram of Oriented Gradient (HOG), Local Binary Pattern, and Integral and Aggregated Channel Features.

Object detection and localization techniques employing these features are frequently applied to detect pedestrians and vehicles. In the automotive sector, researchers have explored diverse algorithms to achieve real-time object localization and detection. For instance, in [12-15], a single rear-view fisheye camera was used to identify vehicles by implementing Support Vector Machines (SVM) within selected candidate regions. Another approach by Jiong et al. [16] employed an ensemble classifier, where a framework incorporating Random Forests and a feature selector was proposed for pedestrian detection. While successful in identifying smaller pedestrians, this framework struggled with occluded pedestrian detection, leading to moderate accuracy. This study focuses on evaluating the feasibility of state-of-the-art object detectors for fisheye image object detection. The research is conducted on a vehicle equipped with a sensor setup illustrated in Fig. 1. The vehicle is outfitted with two pinhole cameras, one each at the front and rear, along with four fisheye cameras, one on each side. Additionally, the vehicle incorporates Ultrasonic sensors at both the front and rear. A key challenge with fisheye data lies in the distortion introduced during dewarping of fisheye images. Unlike regular pinhole camera images where objects, such as pedestrians, appear undistorted, fisheye images exhibit pronounced distortion in order to provide a panoramic view of the surroundings. Consequently, individuals captured on fisheye images exhibit varying proportions and dimensions. Those in close proximity to the camera appear stretched and skewed on the edges, as depicted in Fig. 2, while those at a distance appear smaller than usual. Hence, the detection of objects within fisheye images presents a formidable task. To achieve real-time pedestrian detection, this research evaluates three state-of-the-art object detectors: YOLOv3 [17], Tiny- YOLOv3 [18], and ResNet-50 [19], analyzing the trade-off between accuracy and performance of these chosen detectors.

Fig. 1. Pedestrian detection on a fisheye image [16].

II. METHODOLOGY

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. A basic Artificial Neural Network (ANN) is commonly referred to as a shallow network. Networks with numerous layers between input and output configurations are recognized as Deep Convolutional Neural Networks (DCNNs). This study delves into three specific types of Deep Neural Networks (DNNs): YOLOv3 (You Only Look Once) [17], Tiny-YOLOv3 [18], and Residual Network (ResNet) [19]. Furthermore, machine learning techniques find widespread application across various domains such as image processing, computer vision, natural language processing, medical image analysis, and more [7- 33].

A. ResNet-50

The challenge with deeper networks lies in their accuracy plateauing and subsequently declining significantly as the network's depth increases. ResNet addresses this issue by acquiring and incorporating a "residual" component, which is then added to the existing feature instead of attempting to learn the entire complex feature anew. As a result, this approach leads to reduced complexity in the learning process compared to the straightforward learning of features.

B. YOLOv3

YOLOv3 stands as an advanced object detection system designed to achieve high accuracy while maintaining real-time processing capability. YOLOv3 represents an evolutionary step from its predecessor, YOLO, which operates as a singular neural network, enabling the detection of objects in a single pass. This efficiency is derived from YOLO's approach of treating detections as a regression task, uniting image pixels to generate bounding box coordinates and class probabilities [26, 32]. YOLO divides an input image into a grid arrangement of dimensions S x S, where each grid cell is responsible for detecting a maximum of one object. When an object's center lies within a specific grid cell, that cell takes charge of localizing the object. Additionally, every cell predicts a fixed number (B) of bounding boxes and a number (C) of class probabilities. In simpler terms, upon inputting an image, YOLOv3, operating as a sophisticated CNN, generates an output consisting of a vector of bounding boxes alongside corresponding confidence scores for each predicted class.

C. Tiny-YOLOv3

Tiny-YOLOv3, a derivative of YOLOv3, lives up to its name by being substantially smaller in scale. It boasts fewer than half the number of layers found in YOLOv3, resulting in significantly reduced memory usage during operation. Despite its compact design, Tiny-YOLOv3 maintains comparable accuracy to YOLOv3, all while achieving a higher frame rate. Comprising 13 convolutional layers— markedly fewer than YOLOv3—and 8 max-pooling layers, Tiny-YOLOv3 stands as a streamlined variant. The main distinction between the two networks lies in their structure. YOLOv3, featuring a Feature Pyramid Network (FPN)-like arrangement, is tailored for object detection across three different scales. Conversely, Tiny-YOLOv3 focuses on detecting objects across two scales. Additionally, Tiny-YOLOv3 incorporates fewer skip connections compared to YOLOv3. Nonetheless, both networks operate on similar foundational principles.

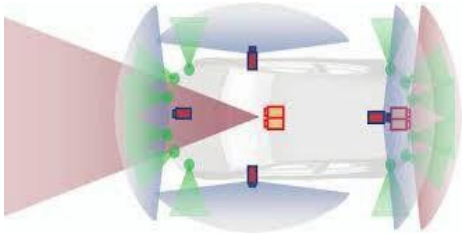

Fig. 2: Layout of the testing vehicle: Four fisheye cameras (indicated in red with a blue border), two pinhole cameras, and Ultrasonic sensors. [11].

D. Fisheye Image Dewarping

A fisheye camera is employed to create an all-encompassing view of the environment, achieved through inducing pronounced visual distortion. In the context of autonomous vehicles, these fisheye cameras are frequently positioned around the vehicle to generate a 360-degree perspective of the surroundings. To rectify the distortions inherent to fisheye images, a process of dewarping is necessary. In Figure 3, a standard fisheye image is juxtaposed with its dewarped counterpart. Extensive research has been conducted across diverse scenarios to address fisheye image dewarping. In this context, we delve into the dewarping technique employed within the Open Source Computer Vision (OpenCV)

Library, providing a broad overview.

Fig.3 A fisheye image

Figure 3 illustrates a test image featuring a black and white chessboard pattern. These images serve as inputs for the calibration algorithm. The architecture of the algorithm is

designed to identify distinct points like straight lines and square corners. These identified points are then correlated with real-world coordinates, and distortion coefficients are deduced by approaching this dataset as a mathematical challenge. *To address the current challenge, an experimentalapproach is adopted, involving the assessment of three Deep Neural Networks (DNNs) on the acquired fisheye dataset. The selected DNNs, namely YOLOv3, Tiny-YOLOv3, and ResNet-50, are evaluated for their pedestrian detection performance.*

E. Data Collection

Video sequences were gathered by driving the test car on various roads using all four fisheye cameras. These sequences, initially in "h264" format, were transformed into individual frames in "jpeg" format. Each frame exhibited dimensions of 1080x1280 pixels in the color space. An initial collection of 30,000 frames encompassed imagery from all cameras. However, upon thorough examination, over half of this data proved non-usable due to the absence of pedestrians and the small dimensions of distant objects. Consequently, an additional 70,000 frames were obtained from bustling streets, utilizing fisheye cameras to guarantee the inclusion of at least one pedestrian in each frame and to ensure closer proximity of objects to the car. In total, approximately 100,000 frames were amassed, but only 15,000 frames proved viable for the learning process. Worth noting is that the data collection locale experienced over 16 hours of sunlight daily during the gathering period, thereby rendering the dataset limited to a single environmental setting.

1*)* Image Dewarping Image dewarping involves rectifying the perspective of an input image to eliminate the lens-induced distortion effects. In this case, all 15,000 images underwent dewarping to restore them to their undistorted forms. The applied dewarping procedure resulted in an output image size of 2160x2560 pixels, achieved from an initial fisheye image dimension of 1080x1280 pixels.

2) Data Labeling The complete set of 15,000 dewarped frames underwent labeling through utilization of a graphical annotation tool. For every individual image, a corresponding annotation file was generated. Each of these annotation files is composed of rows, where each row corresponds to an object present within the image.

3) Training The acquired dataset was partitioned into training and testing subsets, adhering to an 80-20 distribution. From the dewarped fisheye dataset comprising 15,000 images, 12,500 were allocated for training, and 2,500 for testing. These selections were randomized, yet subject to a constraint: images from the same video sequence could belong to either the training or testing set, but not both. For training, each network underwent transfer learning, a choice driven by the aim to reduce both training time and computational demands. This approach also considered the relatively smaller number of images available for training the neural networks. Training occurred on a multi-GPU setup encompassing two Tesla K80 GPUs, each with a 12 GB RAM capacity. Detailed information regarding parameter selection and training specifics for each network is provided in subsequent sections.

YOLOv3 The network architecture for YOLOv3 is specified in reference [15]. The network's detectors and convolutional layers were adjusted for the purpose of localizing a single class, specifically pedestrians. In the training process, we initiated with a pre-trained model, as mentioned earlier. This model was then trained on the fisheye training dataset utilizing the pre-existing weights. Images were resized to dimensions of 608 x 608 pixels and inputted into the model iteratively until the cost value stabilized. To ensure the training process, a batch size of 1024 was selected, accompanied by a consistent learning rate of 0.001. Despite the capability of the chosen GPU to accommodate larger batch sizes, we opted for a maximum batch size of 1024, as larger batch sizes exhibited suboptimal performance for Deep Neural Networks (DNNs). The network underwent training for 200 epochs until the point where the loss value saturated, reaching an approximate training loss of 0.38.. Tiny-YOLOv3 The network architecture of Tiny-YOLOv3 is delineated in reference [10]. The network's detectors and convolutional layers were similarly adjusted to identify and localize a singular class, pedestrians. Employing a comparable

approach to YOLOv3, we initiated training using a pre-trained model. Subsequently, the network was trained on the fisheye training dataset utilizing these pre-established weights. The dimensions of the fisheye images were altered to 416 x 416 pixels, and the resized images were then employed as input for the Tiny-YOLOv3 model. In line with YOLOv3, a batch size of 1024 was selected, accompanied by a learning rate of 0.001. The network underwent training for 530 epochs until the loss value stabilized, ultimately culminating in an approximate training loss of 0.42. ResNet-50 The network architecture of ResNet-50 is outlined in reference [11]. For our implementation, we substituted the feature extractor of YOLOv3 with ResNet-50, configuring the YOLOv3 detector to focus on a single class—pedestrians. To initiate network training, we utilized pre-trained weights from the ResNet-50 model, excluding the detection layer, which was initialized randomly. Subsequently, the network underwent training on the fisheye training dataset employing these new weights. Image dimensions were adjusted to 256 x 256 pixels for training input, using the resized images. Consistent with the approach employed for other networks, we employed a batch size of 1024 and maintained a steady learning rate of 0.001. The training process encompassed 560 epochs until the point of loss saturation. Ultimately, the training loss approximated 0.6 at the conclusion of the training phase. Evaluation Metrics Evaluation metrics such as Precision, Recall, Average Precision, and Intersection Over Union (IOU) were selected for assessment. The evaluation process encompasses several fundamental concepts, including: True Positive rate (TP): A correct detection False Positive rate (FP): A wrong detection False Negative rate (FN): A ground truth not detected True Negative rate (TN): Does not apply.

III. RESULTS AND DISCUSSION

The subsequent Figures, labeled as Figs. 5, 6, 7, and 8, illustrate the distribution of True Positives (TP), False Positives (FP), and False Negatives (FN) for the three networks. The test dataset encompasses a total of 4,246 distinct objects. Across all three graphs, the green line corresponds to the ground truth, the blue line represents true positives, the red line signifies false positives, and the yellow line denotes false negatives. Upon analyzing these presented graphs, it becomes evident that YOLOv3 achieves the highest true positive rate among the three models, whereas ResNet-50 exhibits the highest false negative rate across various Intersection Over Union (IOU) thresholds. Additionally, a noticeable trend is the gradual decline in model performance as the IOU threshold increases.

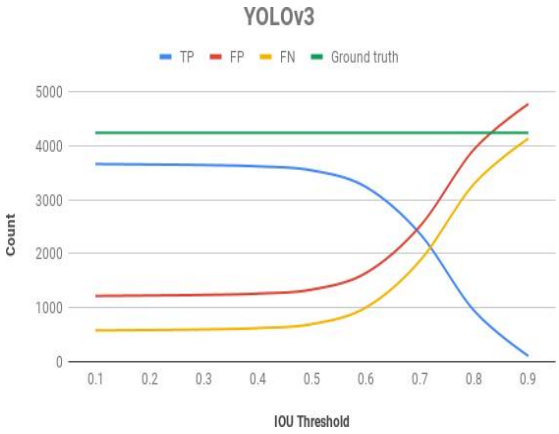

Fig. 5 Detection metrics of YOLOv3.

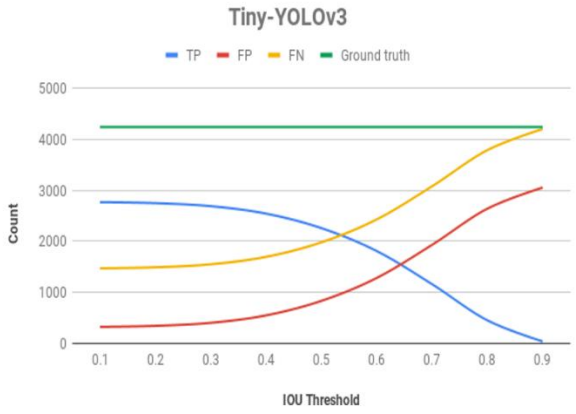

Fig. 6 Detection metrics of Tiny-YOLOv3.

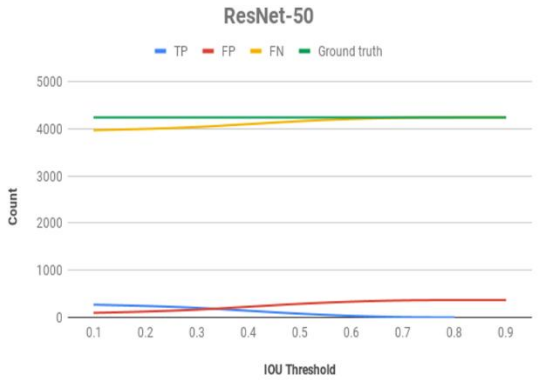

Fig. 7 Detection metrics of ResNet-50.

Based on the provided Figures above, we computed both Precision and Recall metrics. These calculations were performed at an IOU threshold with a scale difference of 0.1. Furthermore, Mean Average Precision (mAP) was determined for the three networks. This mAP was calculated using three distinct IOU thresholds: 0.75, 0.50, and 0.25. The ensuing table showcases the mAP values (expressed as percentages). The results are tabulated in Table I, while Figures 9 and 10 graphically depict the obtained outcomes.

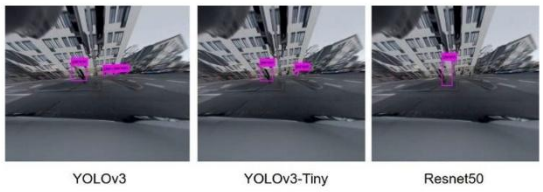

Fig. 9 Left: YOLOv3 detecting all ground truth. Center: Tiny missing one object. Right: ResNet Missing two objects.

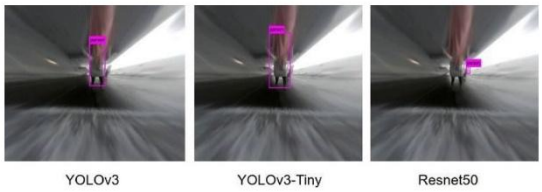

Fig. 10 Model predictions on closer objects.

Throughout the entirety of the project, the dataset emerged as a pivotal challenge. The process of dewarping fisheye images introduced a measure of distortion, with objects positioned at the periphery of the dewarped images exhibiting more pronounced distortion than those in the central region. This disparity in distortion resulted in the models struggling to detect objects positioned at the sides, consequently creating blind spots. Despite the fisheye cameras boasting a 160-degree field of view, the effective detection range narrowed down to 90 to 120 degrees due to the center-focused detection, leading to blind spots surrounding the vehicle. In terms of real-time performance, optimizing the models assumes critical importance to enhance detection speed, allowing for scalability across multiple cameras. Furthermore, the optimization of networks is crucial to curbing memory consumption, a vital consideration when deploying models onto embedded systems. It appears that the implementation of models in the darknet version might not be suitably optimized. In this context, leveraging alternative library implementations such as TensorFlow, renowned for its optimization techniques, holds potential advantages, addressing both adaptability and computational performance.

TABLE I. MEAN AVERAGE PRECISION (MAP).

Threshold	YOLOv3	Tiny-YOLOv3	ResNet-50
IOU = 0.75	**18.22**	5.87	0.0
IOU = 0.50	**87.13**	51.61	11.4
IOU = 0.25	**83.11**	82.66	19.31

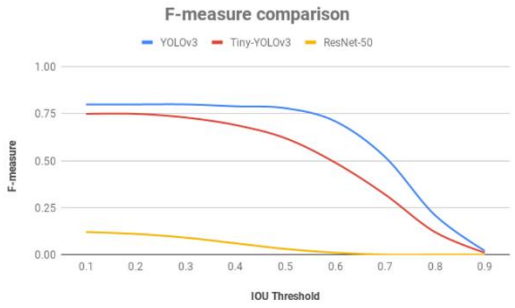

Fig. 8 F-measure comparison of three models.

IV. CONCLUSION

Based on the outcomes and subsequent examination, it can be inferred that the null hypothesis, which posits that the three DNNs exhibit similar detection performance, can be refuted. As such, we can embrace the

alternative hypothesis that the detection performance of the three DNNs varies. Our efforts in training YOLOv3 model variants to identify pedestrians within dewarped fisheye images have proven fruitful. However, it is important to note that the ResNet-50 model has fallen short in generating satisfactory detections.

REFERENCES

[1] K. Bimbraw, "Autonomous cars: Past, present and future - a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," *ICINCO 2015 - 12th International Conference on Informatics in Control, Automation and Robotics, Proceedings*, vol. 1, pp. 191–198, Jan. 2015. doi: 10.5220/0005540501910198.

[2] N.-N. Zheng, S. Tang, H. Cheng, Q. Li, G. Lai, and F.-Y. Wang, "Toward Intelligent Driver-Assistance and Safety Warning Systems," *IEEE Intelligent Systems*, vol. 19, pp. 8–11, Mar. 2004. doi: 10.1109/MIS.2004.1274904.

[3] A. Shaout, D. Colella, and S. Awad, "Advanced Driver Assistance Systems - Past, present and future," in *2011 Seventh International Computer Engineering Conference (ICENCO'2011),* Dec. 2011, pp. 72–82. doi: 10.1109/ICENCO. 2011.6153935.

[4] T. B. Lee, Why experts believe cheaper, better lidar is right around the corner, en-us, 00000, Jan. 2018. [Online]. Available: https://arstechnica. com / cars/2018/01/driving- around- without- a- driver-lidar- technology- explained/(visited on 04/21/2019).

[5] B. Zhang, V. Appia, I. Pekkucuksen, Y. Liu, A. U. Batur, P. Shastry,

S. Liu, S. Sivasankaran, and K. Chitnis, "A Surround View Camera Solution for Em- bedded Systems," in 2014 *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Jun. 2014, pp. 676–681. doi: 10.1109/CVPRW.2014. 103.

[6] F. Nielsen, "Surround Video: A Multihead Camera Approach," *The Visual Computer*, vol. 21, pp. 92–103, Feb. 2005, 00034. doi: 10.1007/s00371-004- 0273-z.

[7] T. Celik et al., "Solar-Powered Automated Road Surveillance System for Speed Violation Detection," in *IEEE Transactions on Industrial Electronics*, vol. 57, no. 9, pp. 3216-3227, Sept. 2010, doi: 10.1109/TIE.2009.2038395.

[8] K. Husyeyin et al., "Unsupervised Change Detection in Multitemporal Multispectral Satellite Images Using Parallel Particle Swarm Optimization," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 5, pp. 2151-2164, May 2015, doi: 10.1109/JSTARS.2015.2427274.

[9] K. Rjhijwan et al., "Change Detection in Multispectral Landsat Images Using Multiobjective Evolutionary Algorithm," in *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 3, pp. 414-418, March 2017, doi: 10.1109/LGRS.2016.2645742.

[10] A. Moghimi, et al., "Distortion Robust Relative Radiometric Normalization of Multitemporal and Multisensor Remote Sensing Images Using Image Features," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1-20, 2022, Art no. 5400820, doi: 10.1109/TGRS.2021.3063151.

[11] A. Moghimi et al., "Comparison of Keypoint Detectors and Descriptors for Relative Radiometric Normalization of Bitemporal Remote Sensing Images," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4063-4073, 2021, doi: 10.1109/JSTARS.2021.3069919.

[12] F. Nielsen et al., "Self-adaptive hybrid PSO-GA method for change detection under varying contrast conditions in satellite images," *2016 SAI Computing Conference (SAI),* 2016, pp. 361-368, doi: 10.1109/SAI.2016.7556007.

[13] S. Awad, "Dynamic Multiple Description Wavelet Based Image Coding Using Enhanced Particle Swarm Optimization," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 1, pp. 125- 138, Feb. 2015, doi: 10.1109/JSTSP.2014.2331911.

[14] H. Grahn et al., "Handwriting image enhancement using local learning windowing, Gaussian Mixture Model and k-means clustering," 2016 *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT*), 2016, pp. 305-310, doi: 10.1109/ISSPIT.2016.7886054.

[15] H. A. Shehu et al., "Deep Sentiment Analysis: A Case Study on Stemmed Turkish Twitter Data," in *IEEE Access*, vol. 9, pp. 56836-56854, 2021, doi: 10.1109/ACCESS.2021.3071393.

[16] A. Cheddad, et al., "Object recognition using shape growth pattern," *Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis*, 2017, pp. 47-52, doi: 10.1109/ISPA.2017.8073567.

[17] V. Sai Vineeth et al., "Forecasting Sales of Truck Components: A Machine Learning Approach," *2020 IEEE 10th International Conference on Intelligent Systems* (IS), 2020, pp. 510-516, doi: 10.1109/IS48319.2020.9200128.

[18] T. Mohammed et al., "Self-sampled image resolution enhancement using dual- tree complex wavelet transform," 2009 *17th European Signal Processing Conference*, 2009, pp. 2017-2021.

[19] B. Karabatak et al., "Unsupervised Change Detection using Thin Cloud- Contaminated Landsat Images," 2018 *International Conference on Intelligent Systems* (IS), 2018, pp. 21-25, doi: 10.1109/IS.2018.871047.

[20] Y. Armid et al., Unsupervised Change Detection in Landsat Images with Atmospheric Artifacts: A Fuzzy Multiobjective Approach, *Mathematical Problems in Engineering*, pp. 1-30, 2018.

[21] S. Thummanapally, S. Rijwan and J. Hall, "CArDIS: A Swedish Historical Handwritten Character and Word Dataset," in *IEEE Access*, vol. 10, pp. 55338- 55349, 2022, doi: 10.1109/ACCESS.2022.3175197.

[22] J. Hall et al., ARDIS: a Swedish historical handwritten digit dataset. Neural Comput & Applic 32, 16505–16518 (2020). [12] S. Cayır, S., Solmaz, G. et al., MITNET: a novel dataset and a two-stage deep learning approach for mitosis recognition in whole slide images of breast cancer tissue. *Neural Comput & Applic* 34, 17837–17851 (2022).

[23] H. Cicek et al., UAV detection in airborne optic videos using dilated convolutions. *J Opt* 50, 569–582 (2021).

[24] A Hilmkil et al., SHIBR—The Swedish Historical Birth Records: a semi-annotated dataset. Neural Comput & Applic 33, 15863–15875 (2021).

[25] N. Lavesson et al., DIGITNET: A Deep Handwritten Digit Detection and Recognition Method Using a New Historical Handwritten Digit Dataset*, Big Data Research*, Volume 23, 2021.

[26] M. S. Leeson, B. Kole, E. L. Hines, Meta-heuristic algorithms for optimized network flow wavelet-based image coding, *Applied Soft Computing,* Volume 14, Part C, 2014, Pages 536-553.

[27] A. Yavariabdi, Evolutionary multiobjective multiple description wavelet-based image coding in the presence of mixed noise in images, Applied Soft Computing, Volume 73, 2018, Pages 1039-1052.

[28] H. Cicek et al., FastUAV-NET: A Multi-UAV Detection Algorithm for Embedded Platforms. *Electronics* 2021, 10, 724. [19] M. F. Demir, A. Cankirli, "Real-Time Resistor Color Code Recognition using Image Processing in Mobile Devices," *Int. Conference on Intelligent Systems* (IS), 2018, pp. 26-30.

[29] M. S Leeson, "A Reduced Uncertainty-Based Hybrid Evolutionary Algorithm for Solving Dynamic Shortest-Path Routing Problem", *Journal of Circuits, Systems and Computers,* vol. 25, no. 6, 2015.

[30] F. G. Yasar et al., "Underwater human body detection using computer vision algorithms," 2018 26th *Signal Processing and Communications Applications Conference* (SIU), 2018, pp. 1-4, doi: 10.1109/SIU.2018.8404305.

[31] A. B. Usakli et al., Unsupervised Satellite Change Detection Using Particle Swarm Optimisation in Spherical Coordinates, *Int. J. Sci. Res. Inf. Syst. Eng.(IJSRISE*), 2017.

[32] O. Kilic et al., MITNET: a novel dataset and a two-stage deep learning approach for mitosis recognition in whole slide images of breast cancer tissue. *Neural Computing and Applications*, 2023.

[33] E. Tekin et al., Tubule-U-Net: a novel dataset and deep learning-based tubule segmentation framework in whole slide images of breast cancer, *Scientific Reports*, vol. 1, pp. 1-10, 2023.