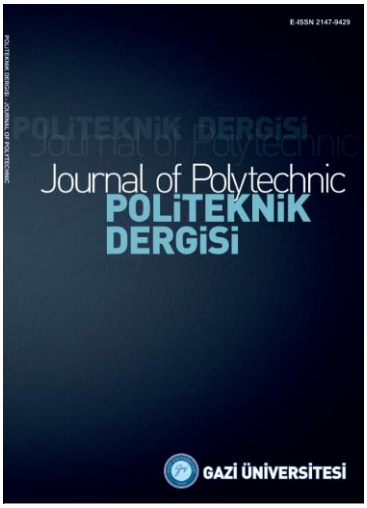

POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.org.tr/politeknik

Branch and end points detection in cerebral vessels images using deep learning object detection techniques

Derin öğrenme nesne tespit teknikleri kullanılarak serebral damar görüntülerinde dal ve uç noktalarının tespiti

Yazar(lar) (Author(s)): *Samet KAYA[1], Berna KİRAZ[2], Ali Yılmaz ÇAMURCU[3]*

ORCID[1]: 0009-0007-0964-686X

ORCID[2]: 0000-0002-8428-3217

ORCID[3]: 0000-0003-1409-9905

To cite to this article: Kaya S., Kiraz B. ve Çamurcu A.Y., "Branch And End Points Detection in Cerebral Vessels Images Using Deep Learning Object Detection Techniques", ***Journal of Polytechnic***, 28(2): 639-648, (2025).

Bu makaleye şu şekilde atıfta bulunabilirsiniz: Kaya S., Kiraz B. ve Çamurcu A.Y., "Branch And End Points Detection in Cerebral Vessels Images Using Deep Learning Object Detection Techniques", ***Politeknik Dergisi***, 28(2): 639-648, (2025).

Erişim linki (To link to this article): http://dergipark.org.tr/politeknik/archive

DOI: 10.2339/politeknik.1492002

Branch and End Points Detection in Cerebral Vessels Images Using Deep Learning Object Detection Techniques

Highlights

- *Detection of vessel branching and endpoints using deep learning object recognition methods.*
- *Creating a branch and end point dataset that can be used for deep learning.*
- *Comparison of deep learning methods.*
- *Quantitative characterization of brain vessels.*

Graphical Abstract

The success of deep learning methods on images of cerebral vessels, which are skeletonized with classical image processing methods and have branching and endpoints, is demonstrated in this study.

	HFHS13							
	Fast R-CNN		Faster R-CNN		RetinaNet		RPN	
	Raw	NMS IoU 0.50	Raw	NMS IoU 0.50	Raw	NMS IoU 0.50	Raw	NMS IoU 0.50
Average Precision (AP) IoU=0.50	0.921	0.95	0.903	0.939	0.243	0.243	0.914	0.945
Average Recall (AR) IoU=0.50	0.621	0.572	0.621	0.558	0.264	0.264	0.626	0.575

Figure. Scores of Deep Learning Object Detection Methods

Aim

To demonstrate that branching and endpoints in vascular networks can be detected with deep learning object detection methods and to compare between methods.

Design & Methodology

A grand-truth dataset was created from the branches and endpoints detected with the classical image processing method, and then the results were revealed by testing deep learning algorithms trained with this dataset.

Originality

There is no published grand-truth dataset for the detection of branching and endpoints of two-dimensional brain vessels. In addition, detection and comparison with different deep learning object recognition methods has not been performed.

Findings

Detecting the branching and endpoints of veins, which can be done with image processing, can also be done with deep learning object recognition methods with very high success.

Conclusion

This work contributes to the advancement of medical imaging analysis by demonstrating the effectiveness of deep learning on challenging tasks such as brain vessel detection. The findings have potential implications for improving clinical diagnosis and research in neurovascular disorders, paving the way for more efficient and accurate analysis of complex biological images in the future.

Declaration of Ethical Standards

The authors of this article affirm that the materials and methods utilized in this study do not necessitate permission from an ethics committee and/or legal-special permissions.

Branch and End Points Detection in Cerebral Vessels Images Using Deep Learning Object Detection Techniques

Araştırma Makalesi / Research Article

Samet KAYA[1]*, Berna KİRAZ[2], A. Yılmaz ÇAMURCU[2]

[1] Department of Software Engineering, Fatih Sultan Mehmet Vakif University, Türkiye
[2] Department of Artificial Intelligence and Data Engineering, Fatih Sultan Mehmet Vakif, University, Türkiye
(Geliş/Received : 02.06.2024 ; Kabul/Accepted : 01.09.2024 ; Erken Görünüm/Early View : 04.09.2024)

ABSTRACT

In this study, we introduce a cutting-edge methodology for detecting branching and endpoints in two-dimensional brain vessel images, employing deep learning-based object detection techniques. While conventional image processing methods are viable alternatives, our adoption of deep learning showcases notable advancements in accuracy and efficiency. Following meticulous cleaning and labeling of the raw dataset sourced from laboratory environments, we meticulously convert it into the COCO format, ensuring compatibility with deep learning algorithms for both training and testing phases. Utilizing four deep learning object detection methods: fast R-CNN, faster R-CNN, RetinaNet and RPN within the Detectron2 framework, our study achieves remarkable results. Evaluation using the intersection over union (IoU) method underscores the robust performance of our deep learning approach, boasting a success rate surpassing 90%. This breakthrough not only enhances neuroimaging analysis but also holds immense potential for revolutionizing diagnostic and research practices in neurovascular studies.

Key Words: Object Detection, Deep Learning, Branch and End Point Detection.

Derin Öğrenme Nesne Tespit Teknikleri Kullanılarak Serebral Damar Görüntülerinde Dal ve Uç Noktalarının Tespiti

ÖZ

Bu çalışmada iki boyutlu beyin damarı görüntülerinde dallanma ve uç noktaları tespit etmek derin öğrenme tabanlı nesne algılama tekniklerini kullanıldığı,son teknoloji bir metodoloji tanıtıyoruz. Geleneksel görüntü işleme yöntemleri geçerli alternatifler olsa da, derin öğrenme kullanarak uyguladığımız yöntem doğruluk ve verimlilik açısından kayda değer başarım sergilemiştir. Laboratuvar ortamlarından aldığımız ham veri setini titizlikle temizleyip etiketledikten sonra COCO formatında veri seti oluşturarak, hem eğitim hem de test aşamalarında derin öğrenme algoritmalarıyla uyumluluk sağlıyoruz. Derin öğrenme nesne tespiti için kullandığımız, Detectron2 çerçevesinde bulunan fast R-CNN, faster R-CNN, RetinaNet ve RPN ile dikkat çekici sonuçlar elde ettik. Birleşim üzerindeki kesişim (IoU) yöntemini kullanarak yaptığımız değerlendirmeyle modelleri karşılaştırdık ve %90'ı aşan bir başarı oranına sahip olan derin öğrenme model performanslarının altını çiziyoruz. Çalışma sadece nörogörüntüleme analizini geliştirmekle kalmıyor, aynı zamanda nörovasküler çalışmalarda teşhis ve araştırma uygulamalarında devrim yaratma konusunda da büyük bir potansiyel taşımaktadır.

Anahtar Kelimeler: Nesne Algılama, Derin Öğrenme, Dal ve Uç Nokta Algılama.

1. INTRODUCTION

Advances in medical imaging technology have enabled the acquisition of high-resolution two-dimensional brain vessel images [1][2][3]. Accurate examination of blood vessels is crucial for characterizing alternations in vascular structure due to disease. This analysis can provide crucial insight into the state of the disease, its progression or the effectiveness of potential treatments [2][4]. This wealth of information inside the images that often cannot be analyzed manually by humans in biological studies [5]. Critical tasks in analyzing such images include detecting branching and end points within the vasculature, as these features have significant diagnostic and research value [6][7]. Traditionally, this process has relied on conventional image processing techniques, which often require extensive manual intervention and lack the scalability needed for large-scale analysis.

Unlike traditional object detection methods, deep learning leverages the power of neural networks to automatically identify relevant features from raw data, potentially offering improved accuracy and efficiency in detecting vascular structures. If we don't take scalability into account and there is no trainable dataset, classical object detection may be a better approach. For all that deep learning approaches have outperformed image processing

*Sorumlu Yazar (Corresponding Author)
e-mail : skaya@fsm.edu.tr

approaches in terms of performance as the amount of information increases [8][9].

A paper presents a computational tool designed for the quantitative analysis of vascular networks [10]. Leveraging advanced mathematical models or machine learning techniques, the tool enables precise measurement and characterization of vascular structures, aiding researchers in understanding vascular morphology, function, and pathology. The tool computes various morphological and spatial parameters, such as the area covered by a vascular network and the number of vessels.

A 2012 study presented an innovative method for the automatically detecting and classifying bifurcation and branching points in retinal vessel images, which are crucial for the diagnosing diseases like diabetic retinopathy. The approach combines morphological operations with local gradient information to accurately identify and classify vascular network structures [8].

Another study [11] uses a multi-tasking network architecture to facilitate the detection of connections in the retinal vasculature, enabling multiple tasks to be performed simultaneously within a unified framework. This approach optimized the efficiency and accuracy of identifying critical features in retinal images, facilitating effective disease diagnosis and treatment planning.

In a study of cerebral vessels, the "Vessel Metrics" software tool uses Python-based techniques to perform automated analysis of vascular structures in confocal imaging [5]. The application enables efficient and precise measurement of vascular properties, facilitating research in various fields such as biology and medicine.

There are many deep learning object detection algorithms [9]. In particular, we use the popular four deep learning object detection method within the Detectron2 framework a state-of-the-art technique known for their robust performance in detecting objects in images [12].

By leveraging sophisticated algorithms, it has achieved accurate segmentation and characterization of brain blood vessels, providing valuable insights for clinical diagnosis and research. The work has employed a convolutional neural network (CNN) for automated analysis of whole brain vasculature, demonstrating its efficacy in segmenting intricate vascular structures with high accuracy and efficiency [1].

One research applied machine learning methods, specifically convolutional neural networks, for analyzing the complete mouse brain vasculature. This study utilized deep learning methods, in particular convolutional neural networks (CNNs), to comprehensively analyze the entire mouse brain vasculature. Using CNNs, the research achieved robust segmentation and extracted intricate vascular network lengths and bifurcation points [13].

A comprehensive protocol is presented for immunofluorescent labelling of mouse brain vessels using tissue sections [14]. Then, the authors made unbiased measurements of vessel density, branching, and tortuosity using two- or three-dimensional digital image processing algorithms, respectively. The study utilized advanced image processing algorithms, including machine learning and computer vision techniques, to conduct an unbiased analysis of mouse brain endothelial networks using 2D or 3D fluorescence images.

In this work, we an innovative method employing deep learning-based object detection to automate the identification of branch and end points in two-dimensional brain vessel images. In classical object recognition, methods were used to find and classify the branching and bifurcation points of veins [8].

There are steps to be followed for object recognition with deep learning [15] [16]. For this study to facilitate the deep learning model's training and evaluation, we firstly preprocess the raw dataset from laboratory experiments to ensure cleanliness and accuracy. The dataset is then rigorously labelled to provide ground truth annotations for training the object detection algorithm. To increase compatibility with the deep learning framework, the labelled dataset is then converted into COCO format, a widely used standard for object detection tasks [17]. The evaluation of our proposed method is robust, focusing on its performance in accurately detecting branching points and endpoints in brain vessel images. We employ the union over intersection (IoU) metric, a reliable measure that quantifies the degree of overlap between predicted and ground truth annotations, providing a strong indicator of detection accuracy. The findings of our study demonstrate the significant success achieved by the deep learning-based approach, with detection accuracy exceeding 90%. These results emphasize the capability of deep learning as a valuable tool for automating the analysis of brain vessel images, offering improved efficiency and accuracy compared to traditional image processing techniques.

Overall, this work contributes to the advancement of the field of medical imaging analysis by demonstrating the effectiveness of deep learning in addressing challenging tasks, such as brain vessel detection, with potential implications for improving clinical diagnosis and research in neurovascular disorders. Moreover, the findings underscore the promising potential of integrating deep learning methodologies into medical imaging analysis and offer valuable insights to improve diagnostic accuracy and treatment strategies in neurovascular disorders.

2. METHODS

In this research, we introduce an approach to automatically detect branches and endpoints in two-dimensional brain vessel images by employing object detection methods based on deep learning. The following sections provide detailed insights into the proposed approach.

2.1. Construction of Brain Vessel Dataset

Raw images need to be transformed into a data set for deep learning algorithms to work with them. The data preparation process can be divided into three parts: (1) image skeletonization; (2) label objects (their location and class) on images; (3) convert the images into the COCO format.

2.1.1. Construction image skeletonization

In microscope images of the mouse brain, the vessels are easily visible. In order to be processed by computer vision techniques, some pre-processing is required. An exemplary image is given in Figure 1.

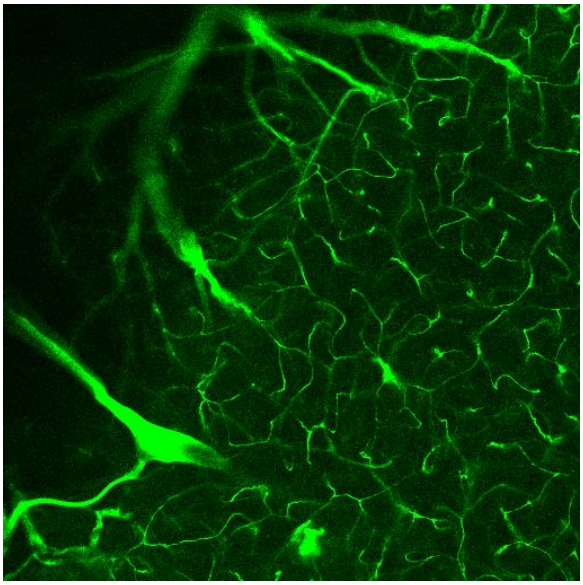

Figure 1. An exemplary image used in the experiments.

Images of the mouse brain taken with a microscope have so much noise that it is almost impossible to find the vasculature. Firstly, we denoise images and then, we perform segmentation and skeletonization methods, respectively. Five consecutive pre-processing steps, as shown in Figure, are performed to obtain skeletonized images.

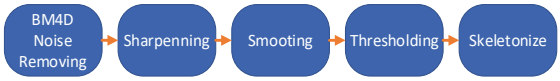

Figure 2. Image Pre-Processing Steps

In the pre-processing steps, we first remove the noise from the images. Since no reference image exists for the noise removal, we use the block matching algorithm (BM4D), which takes the reference from within itself [18][19]. Then, we sharpen and smooth process to make the veins more prominent. The next step is to segment the veins. Among there are many proposed methods for this process [20][21]. However, in order not to shift the focus of the study, we only examine Otsu thresholding [22]. Lastly, we convert the segmented images into a binary skeleton using the 2D thinning algorithm [23]. After all these procedures, the images are suitable for finding branch and end points. The skeletonized image is shown in Figure 3. The image consists of a binary vessel network.

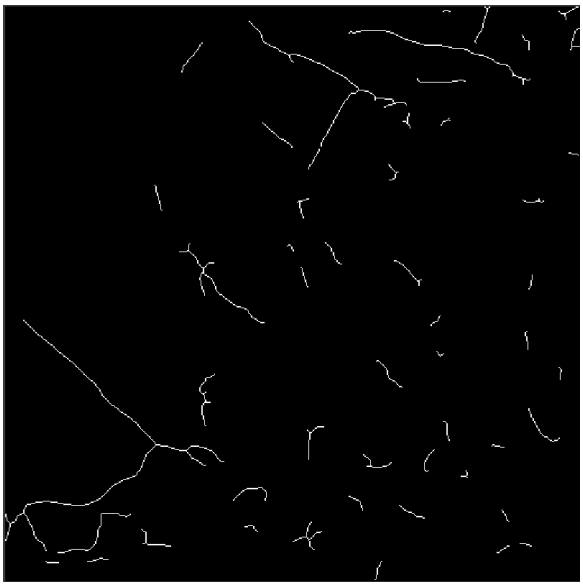

Figure 3: Skeletonized image obtained by the pre-processing methods.

2.1.2. Object labelling

To detect the objects (branch points and endpoints), we employ the Hit-or-Miss transform algorithm, which is a morphological operation used for pattern matching in image processing. It aims to detect patterns characterized by specific configurations of foreground and background pixels. The algorithm identifies regions in the input image where the pattern matches, using a structuring element that defines the pattern to be detected. This process involves performing two morphological operations: erosion and complementation. This algorithm is particularly useful for tasks such as shape detection in binary images, object recognition and image analysis. Its efficiency lies in its ability to accurately find complex patterns in an image [24].

The hit-miss algorithm finds branches and endpoints accurately and precisely. In this method, 3x3 spatial filters are moved over the entire image. Then, we find the exact centers of the branch and endpoints of the skeletonized vessels accurately.

Branches and endpoints require different filters. We consider 8 filters for endpoints and 16 filters for branch points. All these filters are given in Figure 4 and Figure 5. We develop a program using Python that finds and visualizes the points by moving the filters spatially on the image with the hit-mis algorithm [25].

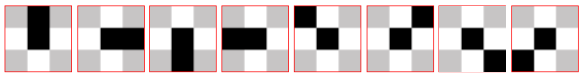

Figure 4: 8 Filters Used for End Points.

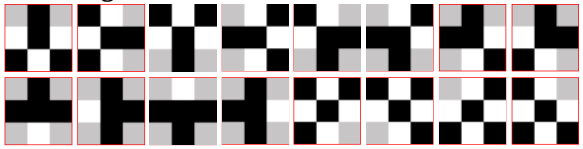

Figure 5: 16 Filters Used for Branch Points.

The visualization of the branch points and endpoints of the vascular skeleton image in Figure 3 is shown in Figure 6. In this figure, the endpoints are marked with green circles and the branching points marked with red circles.

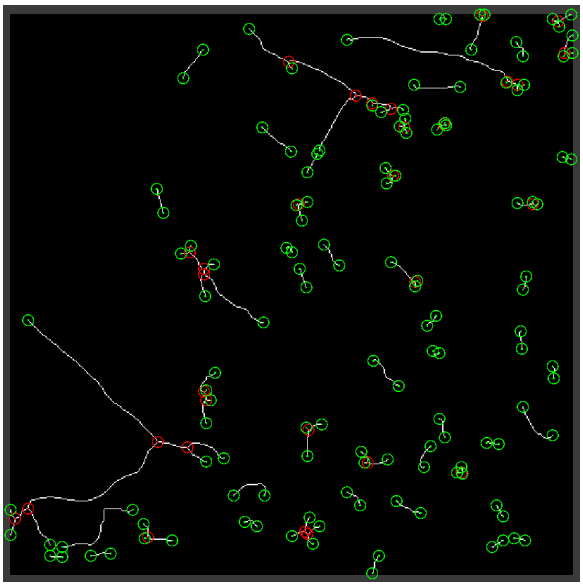

Figure 6. Branch points and endpoints in the exemplary image

2.1.3. Image conversion to COCO format

There are many data markup formats for object recognition. Some of the most widely used are PASCAL VOC (Visual Object Classes), YOLO (You Only Look Once), LabelImg XML and Coco [26][27][28]. For dataset annotations, we prefer Microsoft's Common Objects in Context (COCO) format [17]. COCO is a dataset format developed by Microsoft for tasks such as object localization and semantic segmentation. It is widely used for training and evaluating deep learning models.

Traditional image processing with 3x3 pixel sized filters had found all objects correctly. For the deep learning object recognition method, these objects should not be included in the frame. Therefore, we enclose the spatially detected branch and end points in each image in a 5x5 pixel frame. "B" and "T" represent branch points and end points, respectively.

It was difficult to observe the accuracy of the markings in the Coco formatted dataset, so we saved images of the raw branch and end points framed so that we could quickly observe the position and type accuracy of the markings. In such experiments, we observed that objects of two different types in all the data sets we used were correctly labeled.

The mouse cortex images obtained from cleared mouse brain samples processed through the 3DISCO procedure [3], namely HFHS13, is used in the experiments. This dataset has 96 images with the size of 512×512, and it is randomly divided into two sets, 80% of the images as training set and 20% as test set.

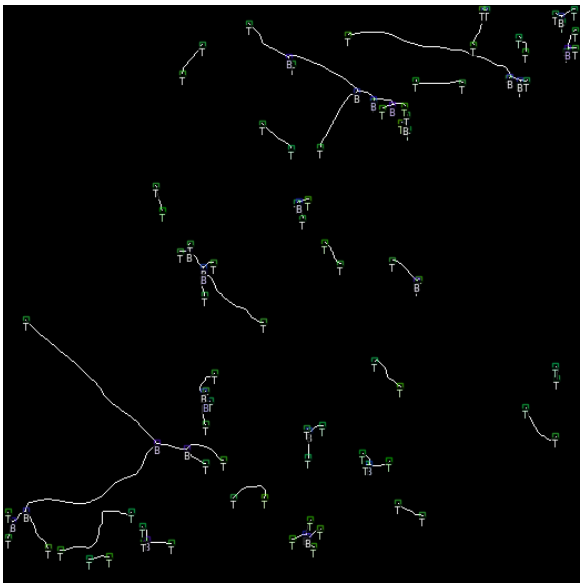

Figure 7. Marked branch and end points

2.1.4. Deep learning object detection algorithms

We apply 16 and 8 filters to the whole image to detect the branch points and endpoints, respectively. The time complexity of the hit-miss algorithm is given in Equation (1)

$$\binom{3 \times 3}{Filter\ Size} x \left(\binom{8}{End\ Point\ Filters} + \binom{16}{Branch\ Point\ Filters} \right) \binom{n \times n}{Image\ Size}$$

$$216 \times n^2 \qquad (1)$$

$$O(n^2)$$

As can be seen from the complexity calculation, the time complexity increases exponentially as the image size grows.

Traditional image processing methods are often based on fixed algorithms and manually defined features. These methods operate on specific rules and thresholds that can be effective for certain types of images but are usually inadequate when dealing with complex and variable structures. In biomedical imaging, for example, these methods have notable limitations when analyzing tissues such as the rat brain.

The limitations of traditional methods can be categorized as follows:

Feature Selection: Traditional methods require manual definition of features (e.g., edges, textures, colors). This requires customized, case-specific algorithms and reduces flexibility.

General Applicability: Fixed rules can be sensitive to image changes (e.g., lighting, contrast differences), making

it difficult to achieve consistent results across different datasets.

Human Intervention: These methods often require manual intervention, which is time-consuming and error-prone. They also require specialized knowledge, which limits their application.

Complexity and Scalability: Traditional methods find processing complex and large datasets challenging. They may need help to process large amounts of data efficiently[29].

Deep learning offers a powerful alternative to overcome these limitations. Deep learning models, especially Convolutional Neural Networks (CNNs), can automatically learn and identify complex image features. The main advantages of deep learning are:

Feature Learning: Deep learning models excel in automatically extracting meaningful features from raw data, significantly reducing the need for human intervention and manual feature engineering.

General Applicability and Adaptability: Deep learning models demonstrate remarkable adaptability to various datasets and conditions. When trained on extensive and diverse datasets, these models exhibit a more general and robust performance.

High Accuracy: Deep learning generally provides higher accuracy than traditional methods in image classification, segmentation, and recognition tasks.

Efficiency and Scalability: Deep learning models trained with GPUs and large datasets can process large amounts of data quickly and efficiently.

In this context, deep learning is considered an advance in accuracy and efficiency as it addresses the limitations of traditional methods. This is particularly beneficial in fields such as biomedical imaging, where more accurate, efficient, and scalable solutions can significantly improve the quality of research and applications[30][31].

In the investigation, because of its complexity given at Equation (1), we employ the deep learning-based object detection methods instead of the hit-miss algorithm. We consider four object detection methods within the Detectron2 framework [12]: Fast Region-Convolutional Neural Network (fast R-CNN), Fast Region-Convolutional Neural Network (Faster R-CNN), RetinaNet, Region Proposal Network (RPN).

In the Detectron2 framework, four different models, namely fast-rcnn-R-50-FPN-1x, faster-rcnn-R-50-FPN-1x, retinanet-R-50-FPN-1x, and rpn-R-50-FPN-1x, are applied. Here, R-50 refers to the ResNet-50 backbone. ResNet stands for Residual Network, and ResNet-50 specifically denotes a variant of the ResNet architecture with 50 layers. FPN, Feature Pyramid Network, addresses the challenge of detecting objects at different scales by constructing a pyramid of feature maps with different resolutions and semantic meanings. “1x” indicates the training schedule. In object detection tasks, ’1x’ typically implies that the model is trained for one epoch over the entire training dataset. This contrasts with ’2x’ or ’3x’ schedules where the model is trained for two or three times over the dataset, respectively. The ’1x’ training schedule is often used as a standard baseline.

3. EXPERIMENTS

3.1. Implementation Details

Our image set contains 96 images with 512x512 size. Bounding boxes with 5x5 pixel size are drawn around each object in an image inside dataset. This work uses four models: fast R-CNN, Faster R-CNN, RetinaNet, RPN for object detection. We then presented the training and test results separately with the training model.

We trained and tested all models separately using the same parameters showing at Table 1 with the same dataset.

Table 1. Hyper-parameters of Model

Property	Value
BASE-LR	0.0001
MAX-ITER	10000
BATCH-SIZE-PER-IMAGE	16
NUM-CLASSES	2
PRE-NMS-TOPK-TRAIN	15000
POST-NMS-TOPK-TRAIN	1500

Base-Lr: This denotes the base learning rate which is a hyper-parameter used to determine the rate at which weights are updated initially. The learning rate controls how much the weights will be adjusted during each update.

Max-Iter: This specifies the maximum number of iterations for training. An iteration corresponds to one pass of the entire training dataset. The maximum iteration count determines how many times the model will be trained during the training process.

Batch-Size-Per-Image: This indicates the batch size used per image. Batchsize refers to the number of samples used for one update. Larger batch sizes require more memory but can reduce training time.

Num-Clasess: This specifies the number of classes the trained model will have. Object detection models are trained to detect a predetermined set of object classes.

Pre-Nms-Topk-Train: It specifies the top-k highest scoring regions to be selected before the pre-NMS (Non-Maximum Suppression) step during training. This is used to limit the number of candidate regions in the region proposal process.

Post-Nms-Topk-Train: This indicates the top-k highest scoring regions to be selected after the NMS step during training. It is used to determine the final set of candidate regions and limits the number of candidate regions used in the multi-object detection process.

3.2. Performance Evaluation

The object recognition methods of joint learning provide a confidence percentage for the objects they predict, reflecting the model's certainty in its predictions. Even if a predicted object with a low confidence score is correct, its

low accuracy rate suggests uncertainty and potential unreliability. To ensure robust and reliable results in our study, we filtered out predictions below 70%. This means only objects with a confidence score above 70% are evaluated, effectively excluding low-confidence detections that could introduce noise and inaccuracies.

Each of the evaluated detected objects is then rigorously assessed based on the Intersection over Union (IoU) overlap rate with the correct object in the ground-truth test data. This IoU overlap rate is a key factor in determining the correctness of the detected object. We used average precision and recall with an IoU threshold of 0.5 for model evaluation. An IoU of 0.5 signifies that a detected object is correct if the reference object frame and the predicted object frame overlap by at least 50%. If the overlap is below this threshold, the detected object is matched with another correct object, if available. If no match is found, the evaluation is negatively impacted, reflecting in lower precision and recall scores.

This approach ensures our evaluation metrics are stringent and meaningful, focusing on high-confidence and accurate detections. By setting a high threshold for prediction confidence and using IoU-based evaluation, we aim to enhance the precision and reliability of our object recognition system. This systematic process reduces false positives and improves overall performance, making our system more effective and trustworthy for practical applications where high accuracy is crucial.

We show two types of test results: raw and filtered. In object recognition studies, an object can be detected and framed more than once. In other words, within an image, the same object can be detected and framed more than once, or the frames of different detected objects can intersect at a certain rate. Therefore, a filter is applied afterwards in object recognition studies. The raw results are the results of the object recognition model that we insert directly into the evaluator without any processing and are shown under the "raw" column. The filtered result is the object frames with non max suppression (NMS) applied to the raw output of the model with 0.5 IoU. If two objects of the same type overlap at frames above 50%, we ignore one of them. You can see these results under the "filtered" column. This post-processing step ensures that the final results accurately represent the unique objects detected in the images, minimizing redundancy and increasing the overall precision of the object recognition system.

3.3. Results and Discussion

Table 2 presents the average precision and recall values, while Table 3 gives the numerical values of the detected branches and endpoints. In both tables you can see raw and filtered values.

Table 2. Average Precision and Recall Values Obtained by Four Models on HFHS13 dataset

		HFHS13	
		Average Precision (AP) IoU=0.5	**Average Recall (AR) IoU=0.5**
Fast R-CNN R_50 FPN_1x	**Raw**	0.921	0.621
	NMS IoU 0.5	0.950	0.572
Faster R-CNN R_50 FPN_1x	**Raw**	0.903	0.621
	NMS IoU 0.5	0.939	0.558
RetinaNet R_50 FPN_1x	**Raw**	0.243	0.264
	NMS IoU 0.5	0.243	0.264
RPN R_50 FPN_1x	**Raw**	0.914	0.626
	NMS IoU 0.5	0.945	0.575

The average precision values of all models except RetinaNet, presented in Table 2, are quite high and close to each other. We can also observe that there is an improvement in the values with NMS 0.5 applied. We can also observe that there is an improvement in the values with NMS 0.5 applied.

In the Table 3 where we give the number of detected points, we first give the number of grant-truth branches and endpoints. In each row you can see the predictions of the four models, raw and filtered. It can be more clearly understood on this table that NMS should be applied as post-processing in studies with 5x5 pixel frame size.

Based on the results given in the result tables, the fact that the average precision (AP) in object detection is higher than the average recall (AR) typically indicates that the model is better at precisely identifying and describing detailed items than capturing all instances in the image.

Precision-Recall Trade-off: Precision indicates the accuracy of the model's positive predictions, while recall measures the coverage of positive samples in the data set. Typically, there is a trade-off between precision and recall. Even if a model needs to include some positive samples in the data set, it can achieve high accuracy by making fewer, more accurate, optimistic predictions, which can be done with a lower recall.

Table 3: Branch and End Point Count of Test Images

HFHS13																			
512x512																			
			Fast R-CNN R_50_FPN_1x				Faster R-CNN R_50_FPN_1x				RetinaNet R_50_FPN_1x				RPN R_50_FPN_1x			
File_Name	GT_BP_C	GT_TP_C	R_BP_C	F_BP_C	R_TP_C	F_TP_C	R_BP_C	F_BP_C	R_TP_C	F_TP_C	R_BP_C	F_BP_C	R_TP_C	F_TP_C	R_BP_C	F_BP_C	R_TP_C	F_TP_C
z72	97	146	159	99	190	153	170	98	198	152	236	236	257	257	190	106	191	152
z78	9	70	17	9	104	75	25	10	123	80	65	65	284	284	21	10	122	79
z65	80	143	135	81	177	148	150	81	199	150	190	190	285	285	145	89	187	150
z95	62	142	110	59	183	146	122	65	204	147	176	176	285	285	109	61	194	150
z27	52	118	89	53	143	118	99	52	156	117	172	172	285	285	89	51	162	119
z16	70	182	133	77	216	178	129	77	247	183	162	162	291	291	128	71	225	182
z88	0	6	1	1	9	6	0	0	10	8	93	93	428	423	0	0	8	6
z23	78	163	147	81	201	165	145	80	214	168	187	187	307	307	129	79	218	167
z35	27	71	52	29	87	72	51	26	108	75	169	169	279	279	49	26	91	73
z91	41	149	62	38	183	150	82	41	219	158	111	111	336	336	71	40	222	160
z07	18	46	41	19	63	50	41	21	80	52	217	217	249	249	47	24	75	52
z47	65	132	117	64	158	134	119	64	183	138	133	133	307	307	114	67	186	138
z40	53	119	97	53	148	121	98	53	166	123	141	141	307	307	97	53	166	122
z84	0	6	0	0	6	6	0	0	8	7	131	131	459	452	0	0	6	6
z11	40	114	88	44	137	113	90	46	167	120	157	157	265	265	83	43	149	114
z58	88	165	150	91	201	171	165	93	228	175	179	179	275	275	159	89	216	173
z53	52	167	92	54	207	176	107	54	233	179	101	101	365	365	92	53	216	175
z10	26	82	58	31	120	85	62	29	135	86	173	173	281	281	55	31	136	87
z80	1	11	1	1	13	11	2	1	14	11	137	137	394	391	2	1	12	11
z62	141	114	240	144	153	121	281	149	155	120	323	323	200	200	244	145	156	123

Thresholding: Object detection models often use confidence thresholds to determine when a predicted object will not be considered positive. By setting this, you can adjust the model's sensitivity and refractive operations. A higher light leads to higher sensitivity but lower recall as the model becomes more conservative when making positive predictions.

Detection Difficulty: Some objects in the dataset may be more challenging to detect than others due to size, occlusion, or variability in appearance. The model may prioritize accurately detecting and locating easier-to-detect objects, leading to higher precision but potentially lower recall if it needs to include some harder-to-detect samples.

Class Imbalance: If the dataset is unbalanced, with many more negative examples than positive examples, the model may prioritize minimizing false positives to prevent negative examples from being misclassified as positive. If the model is conservative in making positive predictions, this could lead to higher precision but potentially lower recall.

In summary, the fact that the average precision in object detection is higher than the average recall indicates that the model is more focused on making correct optimistic predictions, even at the expense of potentially missing some positive examples in the dataset. This trade-off is common in many machine learning tasks and must be evaluated based on the constraints of the application and specific requirements.

3.4. Ablation Study

We conducted an ablation study using the KUVESG dataset containing 512x512, 156 images. The KUVESG data set was subjected to the same pre-processing as the HFHS13 data set, and its skeleton was extracted.

The KUVESG dataset has been tested with our model, which was trained with the HFHS13 dataset to evaluate its performance on a different dataset. Table 4 presents the average precision and recall values of the test.

Table 4. Average Precision and Recall Values Obtained by Four Pre-trained Models on KUVESG dataset.

KUVESG			
		Average Precision (AP) IoU=0.5	**Average Recall (AR) IoU=0.5**
Fast R-CNN R_50 FPN_1x	**Raw**	0.916	0.643
	NMS IoU 0.5	0.955	0.598
Faster R-CNN R_50 FPN_1x	**Raw**	0.904	0.613
	NMS IoU 0.5	0.941	0.549
RetinaNet R_50 FPN_1x	**Raw**	0.216	0.261
	NMS IoU 0.5	0.219	0.260
RPN R_50 FPN_1x	**Raw**	0.909	0.623
	NMS IoU 0.5	0.954	0.554

Testing the model trained on the HFHS13 dataset with the KUVESG dataset gives the same result. This shows that our model can work with different datasets.

3.5. Time Comparison

Our aim in this section is to compare the application times of different deep learning models and traditional image processing techniques. This comparison serves to provide insights into the efficiency of these models in terms of application times.

In the methodology for comparison, we give the training time for the HFHS13 dataset and the total object detection time for all test data. Then, we give the test times we did with the KUVESG dataset. The results are shown in Table 5 and Table 6 (TIP – Traditional Image Processing). According to the results, Fast R-CNN, Faster R-CNN, and RPN give similar training and testing times. There is little difference in time when using these models. Although RetinaNet solves in a shorter time, as we have seen before, the prediction scores of the RetinaNet model are pretty bad.

Table 5. HFHS13 (20 images) Train and Test Time Comparison.

Model	**Training Time**	**Test Time (HH:MM:SS:MS)**
Fast R-CNN	2:22:44	00:00:03:781
Faster R-CNN	2:19:14	00:00:03:762
RetinaNet	0:36:48	00:00:01:788
RPN	2:21:38	00:00:03:726
TIP	N/A	00:00:00:780

Table 6. KUVESG (156 images) Test Time Comparison.

Model	**Test Time (HH:MM:SS:MS)**
Fast R-CNN	00:00:23:225
Faster R-CNN	00:00:22:729
RetinaNet	00:00:09:263
RPN	00:00:22:740
TIP	00:00:01:116

The traditional method showed the lowest performance time in object detection. We interpret this as a small dataset. The performance time of the traditional method, for which we have given the complexity formula before, is expected to increase exponentially.

The comparison highlights the trade-offs between deep learning models and traditional image processing techniques. Deep learning models offer scalability and accuracy but have higher training times and complexity. Traditional image processing is faster for small datasets and more straightforward to implement but needs help with scalability and accuracy for larger or more complex datasets. The choice between these approaches should consider the specific requirements and constraints of the application, balancing speed, accuracy, scalability, and resource availability.

4. CONCLUSION

This study presents a new ovel approach to automate the detection of branches and endpoints in two-dimensional brain vessel images using deep learning-based object detection techniques. Traditional methods for analyzing such images often rely on manual intervention and lack scalability for large-scale analysis. However, by leveraging deep learning, this work demonstrates significant advances in automating these critical tasks.

The proposed approach utilizes deep learning models within the Detectron2 framework to accurately detect branching and endpoint features in brain vessel images. Through rigorous pre-processing, labelling, and conversion of the dataset into COCO format, deep learning models are trained and evaluated with a focus on achieving high detection accuracy.

The experimental results show promising results, with detection accuracy exceeding 90% for various deep-learning models. The study demonstrates the effectiveness of deep learning in overcoming the challenges associated with cerebrovascular analysis, offering improved efficiency and accuracy compared to traditional image processing techniques. Furthermore, the study highlights the importance of post-processing techniques such as non-maximum suppression (NMS) to improve detection results and overall performance. This study demonstrates improvements in precision and recall metrics by applying NMS with an intersection-over-union (IoU) threshold of 0.5, contributing to more robust object detection results.

Overall, this work contributes to the advancement of medical imaging analysis by demonstrating the effectiveness of deep learning on challenging tasks such as brain vessel detection. The discoveries hold promise for enhancing clinical diagnosis and advancing research in neurovascular disorders, setting the stage for enhanced precision and efficiency in analyzing intricate biological imagery moving forward.

In future studies, we aim to create a 3D reconstruction model of the entire cerebral vasculature from cross-sectional images of the rat brain by aligning and segmenting each slice images. We are also planning to find branching and endpoints on the 3D reconstructed brain within the scope of the study. This will provide a more detailed and meaningful analysis of the brain vessels and add depth to the research. Thus, more comprehensive and precise results will be obtained for neurovascular research.

SYMBOLS and ABBREVATIONS

GT_BP_C	Ground-Truth Branch Point Count
GT_TP_C	Ground-Truth Tip Point Count
R_BP_C	Raw Branch Point Count
R_TP_C	Raw Tip Point Count
F_BP_C	Filtered Branch Point Count
F_TP_C	Filtered Tip Point Count
TIP	Traditional Image Processing

ACKNOWLEDGEMENT

This work is supported by Fatih Sultan Mehmet Vakif University Scientific Research Projects Coordination Unit under grant number 22022B1Ç01D.

DECLARATION of ETHICAL STANDARDS

The article has been prepared without any conflicts of interest with individuals or institutions, thus, no permissions are necessary from the ethics committee.

AUTHORS' CONTRIBUTIONS

Samet Kaya: conceived the original idea; designed and implemented the model and the computational framework; performed the experiments and analysis of the results; wrote the manuscript.

Berna Kiraz: conceived the original idea; wrote the manuscript; involved in planning and supervised the work.

A. Yılmaz Çamurcu: Involved in planning and supervised the work.

CONFLICT of INTEREST

There is no conflict of interest with any person/institution in the article prepared.

REFERENCES

[1] M. I. Todorov *et al.*, "Automated analysis of whole brain vasculature using machine learning," ***bioRxiv***, pp. 0–34, (2019).

[2] L. Y. Zhang *et al.*, "CLARITY for high-resolution imaging and quantification of vasculature in the whole mouse brain," ***Aging Dis***, vol. 9, no. 2, pp. 262–272, (2018).

[3] E. Özkan *et al.*, "Hyperglycemia with or without insulin resistance triggers different structural changes in brain microcirculation and perivascular matrix," ***Metab Brain Dis***, vol. 38, no. 1, pp. 307–321, (2023).

[4] S. Bollmann *et al.*, "Imaging of the pial arterial vasculature of the human brain in vivo using highresolution 7T time-of-flight angiography," ***Elife***, vol. 11, pp. 1–35, (2022).

[5] S. D. and A. C. and A. S. and G.-W. J. and V. I. and R. K. D. and C. Sarah. J. McGarry, "Vessel Metrics: A python based software tool for automated analysis of vascular structure in confocal imaging," ***bioRxiv***, vol. 151, no. 0026–2862, p. 104610, (2022).

[6] Z. Gu *et al.*, "CE-Net: Context Encoder Network for 2D Medical Image Segmentation," ***IEEE Transactions on Medical Imaging***, vol. 38, no. 10. pp. 2281–2292, (2019).

[7] E. Zudaire, L. Gambardella, C. Kurcz, and S. Vermeren, "A computational tool for quantitative analysis of vascular networks," ***PLoS One***, vol. 6, no. 11, pp. 1–12, (2011).

[8] A. Bhuiyan, B. Nath, and K. Ramamohanarao, "Detection and classification of bifurcation and branch points on retinal vascular network," ***2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)***, pp. 1–8, (2012).

[9] C. Anusha and P. S., "Object Detection using Deep Learning," ***International Journal of Computer Applications***, vol. 182, no. 32. pp. 18–22, (2018).

[10] E. Zudaire, L. Gambardella, C. Kurcz, and S. Vermeren, "A computational tool for quantitative analysis of vascular networks," ***PLoS One***, vol. 6, no. 11, pp. 1–12, (2011).

[11] F. Uslu and A. A. Bharath, "A multi-task network to detect junctions in retinal vasculature," ***Lecture Notes in Computer Science***, vol. 11071 LNCS, pp. 92–100, (2018).

[12] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." (2019).

[13] M. I. Todorov *et al.*, "Machine learning analysis of whole mouse brain vasculature," ***Nat Methods***, vol. 17, no. 4, pp. 442–449, (2020).

[14] M. Freitas-Andrade, C. H. Comin, M. V. da Silva, L. da F. Costa, and B. Lacoste, "Unbiased analysis of mouse brain endothelial networks from two- or three-dimensional

fluorescence images," ***Neurophotonics***, vol. 9, no. 03, (2022).

[15] X. Ji *et al.*, "Brain microvasculature has a common topology with local differences in geometry that match metabolic load," ***Neuron,*** vol. 109, no. 7, pp. 1168-1187.e13, (2021).

[16] J. Kaur and W. Singh, "A systematic review of object detection from images using deep learning," ***Multimedia Tools and Applications***, vol. 83, no. 4. pp. 12253–12338, (2024).

[17] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," ***Computer Vision–ECCV 2014: 13th European Conference***, vol. 8693, pp. 740–755, (2014).

[18] Ş. Sağiroğlu and E. Beşdok, "A novel approach for image denoising based on artificial neural networks," ***Journal of Polytechnic***, vol. 15, no. 2. pp. 71–86, (2012).

[19] Y. Tan, Y., Liu, M., Chen, W., Wang, X., Peng, H., & Wang, "DeepBranch: Deep Neural Networks for Branch Point Detection in Biomedical Images." **IEEE transactions on medical imaging**, pp. 39(4), 1195–1205, (2020).

[20] Z. Kuş *et al.*, "Differential evolution-based neural architecture search for brain vessel segmentation," ***Engineering Science and Technology, an International Journal***, vol. 46. (2023).

[21] N. Akyel, Cihan and Arıcı, "U-Net-RCB7: Image Segmentation Algorithm." ***Journal of Polytechnic***, pp. 1555–1562, (2023).

[22] S. L. Bangare, A. Dubal, P. S. Bangare, and S. T. Patil, "Reviewing otsu's method for image thresholding**,"** ***International Journal of Applied Engineering Research***, vol. 10, no. 9, pp. 21777–21783, (2015).

[23] Y. He, S. H. Kang, and L. Alvarez, "Finding the skeleton of 2d shape and contours: Implementation of hamilton-jacobi skeleton," ***Image Processing On Line***, vol. 11, no. February, pp. 18–36, (2021).

[24] P. Murray and S. Marshall, "A Review of Recent Advances in the Hit-or-Miss Transform," ***Advances in Imaging and Electron Physics***, vol. 175, pp. 221–282, (2013).

[25] S. Kaya, S. Z. Dik, B. Kiraz, M. Aydın, and A. Y. Çamurcu,"BRAINVASCULYZER: 2B Beyin Damar Görüntü Analiz Programı," in **MAS 18th International European Conference on Mathematics, Engineering**, Natural & Medical Sciences, pp. 121–130, (2023).

[26] *Everingham, M., Van Gool, L., Williams, C.K.I. et al., "The PASCAL Visual Object Classes (VOC) Challenge."* ***Int J Comput Vis,*** *88, 303–338 (2010).*

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," ***2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)***, p. 779, (2016).

[28] A. Dave, T. Khurana, P. Tokmakov, C. Schmid, and D. Ramanan, "TAO: A Large-Scale Benchmark for Tracking Any Object," ***Computer Vision ECCV,*** pp. 436–454, (2020).

[29] I. Scholl, T. Aach, T. M. Deserno, and T. Kuhlen, "Challenges of medical image processing," in ***Computer Science-Research and Development***, pp. 5–13. (2011).

[30] J. G. Lee et al., "Deep learning in medical imaging: General overview," ***Korean Journal of Radiology***, vol. 18, no. 4. Korean Radiological Society, pp. 570–584, (2017).

[31] H. P. Chan, R. K. Samala, L. M. Hadjiiski, and C. Zhou, "Deep Learning in Medical Image Analysis," in Advances in Experimental Medicine and Biology, vol. 1213, ***Springer***, pp. 3–2, (2020).