

Enhancing Salient Object Detection Through VGG16–VGG19 Architectures

Abubaker Aljerbi¹, Bashir Ghariba², Omar Marey²,

Abdullah Al Masrub², Abdelrahman Arbi²

¹ Department of Computer Science, Faculty of Information Technology, Alasmarya Islamic University, Zliten, Libya

² Department of Electrical and Computer Engineering, Faculty of Engineering, Elmergib University, Al Khums, Libya

¹abo.aljirbi@asmarya.edu.ly

Abstract

Convolutional Neural Networks (CNNs)-based deep learning techniques have significantly advanced Computer Vision (CV), particularly in Salient Object Detection (SOD). This study investigates two CNN architectures; VGG16 and VGG19 for SOD implementation using the MSRA-10K dataset. The models were trained on 1,160 and 2,700 images to assess the impact of dataset size on detection accuracy. Furthermore, two feature fusion strategies were evaluated: the averaged feature fusion (AvgVGG1619) and the maximum feature fusion (MaxVGG1619). The results indicate that VGG16 generally achieves superior performance, whereas VGG19 performs better on images lacking salient regions. Increasing the dataset size improved the evaluation metrics and significantly reduced the Mean Absolute Error (MAE). Among all models, MaxVGG1619 demonstrated the highest overall accuracy, even when trained on fewer images. Overall, these findings demonstrated the benefits of model fusion and dataset scaling in enhancing CNN-based SOD performance, and demonstrate the effectiveness of combining VGG models, particularly through using maximum feature selection, for robust and accurate salient object detection.

Keywords: Deep Learning, Convolutional Neural Networks, Salient Object Detection, VGG16, VGG19, Feature Fusion, MSRA-10K.

1 INTRODUCTION

The exponential growth of data in the recent years has driven significant advancements in Artificial Neural Networks (ANNs), enabling them to outperform traditional machine learning techniques in learning capability, feature representation, and analytical accuracy. ANNs are computational models that mimic the structure and operation of the human brain, they consist of interconnected processing units (neurons) that collaboratively learn complex nonlinear relationships from input data. When trained on sufficient data, these networks can perform perceptual and cognitive tasks with remarkable efficiency, similar to human intelligence [1]. Among various ANN architectures, Convolutional Neural Networks (CNNs) have emerged as a fundamental deep learning (DL) approach, particularly effective for grid-structured data such as images. CNNs are biologically inspired by the visual cortex and are capable of automatically learning hierarchical spatial representations. They consist primarily of convolutional, pooling, and fully linked layers, where convolutional operations extract localized features and deeper layers integrate them into higher-level abstractions. This hierarchical feature extraction enables CNNs to recognize complex patterns and achieve state-of-the-art performance in a wide range of Computer Vision (CV) tasks, including image classification, segmentation, and object detection [2, 3].

A particularly important application of CNNs is Salient Object Detection (SOD), which has an objective of locating the most visually significant regions in an image that align with the Human Visual System (HVS). SOD plays an essential role in several applications, including object recognition, visual tracking, image retrieval, and scene understanding. It reduces redundant information while improving computational efficiency and interpretability. However, despite the considerable progress in this field, accurately segmenting salient objects remains challenging especially in complex, cluttered, or low-contrast scenes [4, 5]. To address these challenges, advanced CNN architectures such as the Visual Geometry Group (VGG) Networks have been widely utilized. The VGG16 and VGG19 models, characterized by their deep convolutional layers and small receptive filters, were initially designed for large-scale image recognition tasks in the ImageNet challenge. Their uniform architecture and hierarchical design make them well-suited for SOD, as they efficiently extract both low-level and high-level semantic features [6].

In this research, VGG16 and VGG19 architectures are employed for SOD implementation and analysis. The models are trained on multiple datasets of varying sizes to evaluate the effect of dataset volume on detection performance. Furthermore, two feature fusion techniques are introduced: the averaging of output features (AvgVGG1619) and the selection of maximum output features (MaxVGG1619). These fusion strategies are investigated to enhance the detection accuracy, robustness, and quality of salient region localization. By leveraging the hierarchical strength of VGG networks and combining their outputs through optimized feature fusion, the aim is improving the overall accuracy and resilience of deep learning-based salient object detection systems. The importance of this research lies in assessing the performance of VGG-based CNNs to accurately identify and highlight salient regions within images while minimizing the prediction errors. Furthermore, combining the output feature maps of both models through averaging and maximum feature selection strategies is explored as a means to improve overall accuracy and reduce output loss. The rest of this paper is structured as follows, the following section presents a review of related work. It is followed by a description of the methodology section, which describes the datasets, experimental setup, training procedures for VGG16, and VGG19 on the SOD datasets, and the evaluation metrics used. The experimental results section presents and analyses the training outcomes of both models under different dataset sizes and compares their performance. The Discussion section interprets and contextualizes the obtained results. Finally, the Conclusion and Recommendation section summarizes the key findings, provides final insights, and outlines directions for future research.

2 RELATED WORK

Salient Object Detection (SOD) in computer vision has received considerable attention from researchers over the years. A wide variety of saliency detection methods has emerged over the past two decades [7]. Conventional salient object detection approaches are primarily based on hand-crafted local features, which encode low-level visual cues and global features [8 -13]. Traditional computer vision methods were based mainly on hand-crafted features including Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and colour histograms. Even though these features demonstrated effectiveness in constrained scenarios, their design required substantial domain-specific knowledge and exhibited limited applicability across diverse tasks and datasets. Li et al. proposed a saliency detection method that captures multi-scale hierarchical features using a deep convolutional neural network [14]. Wang et al. developed a method for saliency prediction that integrates local estimations with a global search process [15]. Separate deep convolutional neural networks are utilized to learn local feature representations and capture global contextual contrast. To improve salient object detection, Zhao et al. developed a deep learning framework that fuses multiple levels of contextual information [16]. They utilized two distinct convolutional neural networks to independently capture global and local contextual information. The study by Lee et al. considered a feature fusion strategy, incorporating both CNN-based and manually engineered features [17]. To merge the extracted features, a united fully connected neural architecture was applied for the purpose of saliency estimation. Liu et al. [18] developed a two-stage deep neural network in which a coarse prediction map was initially generated, followed by a subsequent network that hierarchically and progressively refined the prediction details. A deep contrast network was suggested in [19]. Two complementary components were included in that network, a pixel-level fully convolutional stream and a segment-wise spatial pooling stream.

FCN-based methods are another solution for SOD [19, 20]. These methods achieve a significant improvement over patch-wise deep networks because FCNs can capture richer, multi-scale information. Hu et al. [21] introduced a method for generating precise object boundaries and compact saliency maps by learning a level set function. Luo et al. [22] proposed a network architecture based on a (4×5) grid structure to combine global and local information, then used a fusing loss of cross-entropy and boundary IoU inspired by Mumford and Shah. Zhang et al. introduced a novel sibling network architecture coupled with a structural loss function to enhance saliency prediction [23]. Despite the considerable progress that has been achieved in the last five years, there is still substantial room for improvement over common CNN models for SOD.

3 METHODOLOGY

3.1 Semantic Segmentation Technique

Image segmentation is the process of partitioning a digital image into multiple meaningful regions or objects, where each region consists of a set of pixels with similar characteristics. This process simplifies the representation of an image, making it easier to analyze and interpret its visual content [24]. Among the various segmentation techniques, semantic segmentation has emerged as one of the most important tasks in computer vision. It provides scene understanding by assigning a semantic class label to every pixel in an image, thereby performing pixel-level classification. Unlike traditional image classification, semantic segmentation identifies both the category and the precise location of objects within an image.

Semantic segmentation has many real-world applications, including medical image analysis, autonomous driving, robotics, remote sensing, environmental monitoring, and intelligent surveillance. It has become a fundamental component of many modern computer vision systems because of its ability to accurately delineate object boundaries and classify image regions. To improve the performance of semantic segmentation, deep learning models, particularly Convolutional Neural Networks (CNNs), have been widely adopted. The following section introduces the pre-trained CNN architectures, **VGG16** and **VGG19**, which have been employed to enhance the accuracy and effectiveness of semantic segmentation tasks [25].

3.2 Visual Geometry Group (VGG) Networks

VGG, which stands for **Visual Geometry Group**, is a family of deep convolutional neural network (CNN) architectures developed by researchers at the **University of Oxford** in **2014**. They represented a significant advancement in CNNs developed by Karen Simonyan and Andrew Zisserman [26]. It is characterized by a uniform design employing small (3×3) convolutional kernels, systematically stacked to increase network depth, thereby facilitating hierarchical feature extraction for large-scale image recognition tasks. It is a standard deep CNN architecture with multiple layers. The word “deep” refers to the number of layers in VGG-16 and VGG-19, they consist of 16 and 19 convolutional layers. The idea of the model was initially proposed in 2013, the actual model was later submitted to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. The ILSVRC was an annual competition for evaluating algorithms of image classification, and object detection on a large-scale level. VGG is classical CNN architecture that utilizes small filters. During training, the input to VGG-Net is a fixed-size $(224 \times 224 \times 3)$ RGB image [27].

3.2.1 VGG-16

VGG-16 network contains 13 convolution layers, 5 max-pooling layers, and 3 fully connected layers [26]. The VGG-16 network has an image input size of (224×224) , and can classify images into 1,000 object classes. The major difference between the VGG-16 network and other networks is the use of a series of convolution layers with small receptive fields (3×3) in the first layers instead of a few layers [28]. Figure 1 shows the general structure of VGG-16.

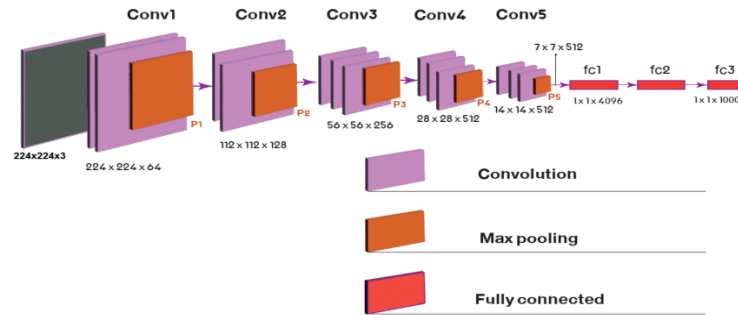


Figure 1: VGG-16 Architecture.

3.2.2 VGG-19

VGG-19 network consists of 16 convolution layers, 5 pooling layers and 3 fully linked layers. It has an input image size of (224x224x3). This network has ability to classify images into 1,000 classes; this increases the depth of the network and contributes to learning more features for Salient Object Detection [26]. Figure 2 shows the general structure for VGG-19.

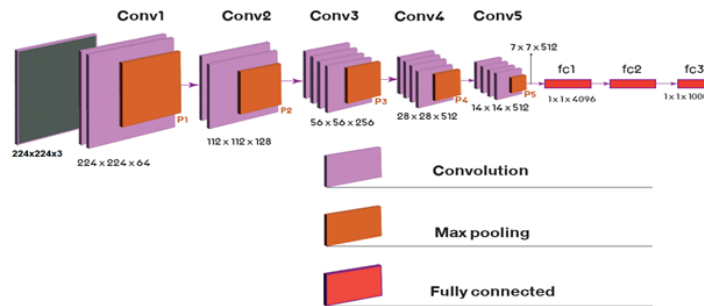


Figure 2: VGG-19 Architecture.

The input image passes over a sequence of convolution layers with (3x3) convolutional filters. Both the convolution stride and padding are set to one pixel. Whereas five max-pooling layers are used after convolution layers. Each max-pooling is executed over (2x2) pooling filter, and a stride value equal 2. Three FC layers follow a series of convolution layers with the first two layers have 4,096 channels each, while the third one has 1,000 channels.

3.3 Datasets

3.3.1 MSRA-10K

This dataset is designed for salient object detection and comprises 10,000 images with pixel-level saliency labeling, sourced from Microsoft Research Asia (MSRA). All the images have a fixed bounding box [12].

3.3.2 ECSSD

The ECSSD dataset stands for "Extended Complex Scene Saliency Dataset". This dataset consists of complex scenes, presenting textures, common structures, and complex natural contents in real-world images. It consists of 1,000 semantically meaningful natural images with pixel-level ground truth annotations, making it suitable for evaluating salient object detection algorithms in complex scenes [29].

3.3.3 MSRA-B

MSRA-B stands for Microsoft Research Asia Dataset B. It is developed by Microsoft Research Asia for the salient object detection. The dataset contains thousands of images, each annotated to highlight salient objects, thereby it is used by researchers and developers in training and evaluating their models for recognizing important visuals in various scenes [12].

3.3.4 THUR-15k

THUR15K (often written as THUR-15K or THUR15K dataset) is a salient object detection (SOD) dataset introduced by researchers at Tsinghua University (THUR). It is used as an important resource for the development and evaluation of computational models focused on visual attention mechanisms. This dataset contains approximately 15,000 images in JPG format and was created to facilitate the evaluation of sketch-based image retrieval. This dataset has also been divided into five subset datasets, including a butterfly, coffeeMug, dogjump, giraffe and plane [10].

3.4 Evaluation Metrics

There are several ways are used to measure agreements between human annotation (i.e., ground truth) and SOD models. Five metrics are adopted for evaluating the SOD in this work. These metrics evaluate **precision**, **accuracy**, **structural similarity**, and **constancy** of predicted saliency maps compared to ground truth. This section summarizes the most widely used SOD evaluation metrics.

3.4.1 Precision – Recall

The precision–recall (PR) metric is calculated by comparing the saliency map with the corresponding ground-truth mask using equation (1):

$$\text{Precision} = \frac{TP}{TP+FP}, \quad \text{Recall} = \frac{TP}{TP+FN} \quad (1)$$

Where TP, TN, FP and FN are true positive, true negative, false positive and false negative, respectively [30].

3.4.2 F-Measure

The F-measure is a key performance metric for evaluating classification models, particularly in situations where class imbalance demands a more comprehensive measure of predictive effectiveness. The standard F-Measure is computed as the harmonic mean of precision and recall as shown in the following equation:

$$\text{F-measure} = \frac{(1+B^2) \text{Precision} \times \text{Recall}}{B^2 \text{Precision} + \text{Recall}} \quad (2)$$

Where is the value of B^2 selected empirically to equal 0.3 to give more weight to precision. Recall rate is not as important as precision [31].

3.4.3 S- Measure

The S-measure is a structure-based evaluation metric proposed to address the limitations of conventional pixel-based metrics (like MAE and F-measure). This metric provides an evaluation of the structural similarity between the real-valued saliency map and the binary ground truth. It considers object-aware (So) and region-aware (Sr) structure similarities as shown in equation (3):

$$s - measure = \alpha So + (1-\alpha) Sr \quad (3)$$

where α is the balance parameter, $\alpha = [0-1]$, empirically set to 0.5 [32].

3.4.4 MAE

MAE denotes the Mean Absolute Error. It computes the average pixel-wise absolute difference between the ground-truth (GT) mask and the normalized saliency map (M). However, this metric does not account for true negative pixels [11]. Equation (4) shows how to calculate the MAE metric.

$$MAE = \frac{1}{W \times H} \sum_{i=1}^w \sum_{j=1}^H ABS(GT(i, j) - M(i, j)) \quad (4)$$

Where W and H refer to the width and height of the given image.

4 EXPERIMENTAL RESULTS

4.1 Model Training

The well-known dataset MSRA-10K is used for training the pre-trained models. In the training stage, a set of trained networks was created for each of VGG16, VGG19, where each model was trained on a set of 1160 images, then it was later trained on a larger set of 2700 images, and the results were compared. The VGG16 and VGG19 models were trained using hyper parameters with specific value for each of them. Each parameter was adjusted to suit the training to give accurate results and faster response. Table (1) gives the parameter values.

Table 1: Training Parameters

Parameters	Values
Epochs	10
Mini-batch	10
Learning rate	0.001
Optimizer	SGDM

4.1.1 Training VGG16, VGG19 Model on 1160 images:

Networks VGG16 and VGG19 were trained on 1160 images, and the training progress was noted, the model accuracy increased, and the loss value decreased. Table (2) lists the accuracy values obtained at the end of training and the time taken for the training process.

Table 2: Training VGG16,VGG19 on 1160 images

Model	Accuracy	Epochs	Iterations	Training Time
VGG16	91.20 %	10	1160	483 min 21 sec
VGG19	90.80 %	10	1160	494 min 30 sec

Figures (3) and (4) show the results of the accuracy and loss curves for training VGG16 and VGG19 on 1160 images. The graphs show that the accuracy of the models increases as the training process progresses and the number of epochs increases.



Figure 3: Accuracy and loss curves for training VGG16 on 1160 images

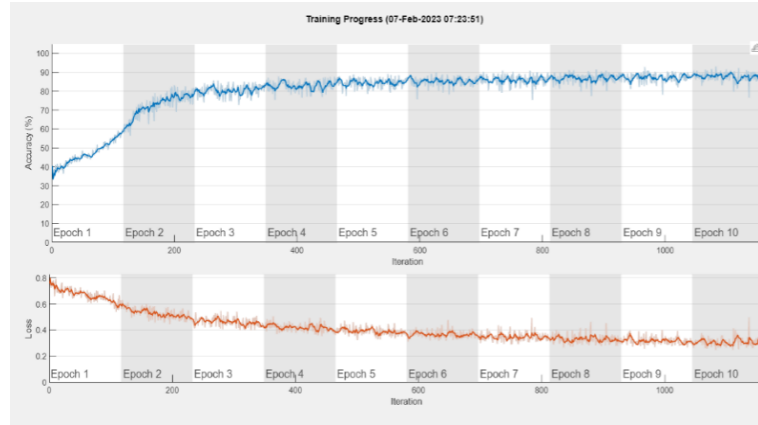


Figure 4: Accuracy and loss curves for training VGG19 on 1160 images

4.1.2 Training VGG16, VGG19 Model on 2700 images

Networks VGG16 and VGG19 were trained on 2700 images, and the training progress was noted, the model accuracy increased and the loss value decreased. Table (3) lists the accuracy values obtained at the end of training and the time taken for the training process.

Table 3: Training VGG16,VGG19 on 2700 images

Model	Accuracy	Epochs	Iterations	Training Time
VGG16	92.50 %	10	2700	2812 min 43 sec
VGG19	93.30 %	10	2700	2834 min 0 sec

The results of the accuracy and loss curves for training both VGG16 and VGG19 on 2700 images are shown in figures (5) and (6). The graphs show that the accuracy of the models increases as the training process progresses and the number of epochs increases, and it is also noted that the accuracy and efficiency of the models increase when they are trained on more images.

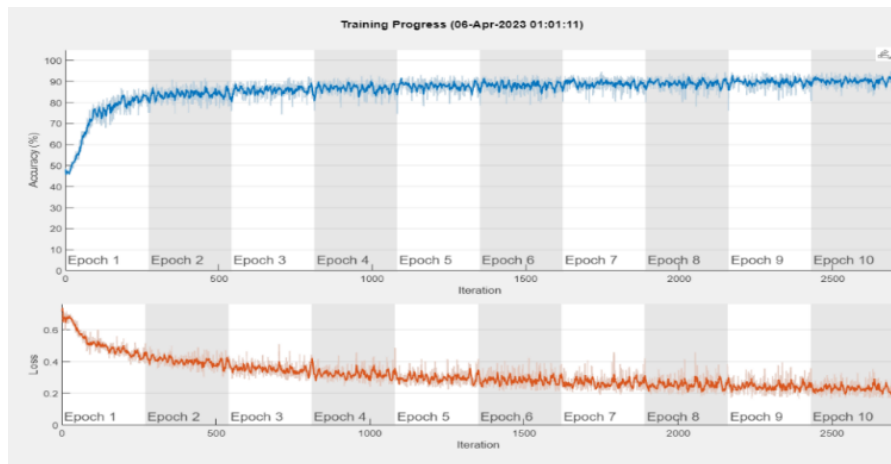


Figure 5: Accuracy and loss curves for training VGG16 on 2700 images

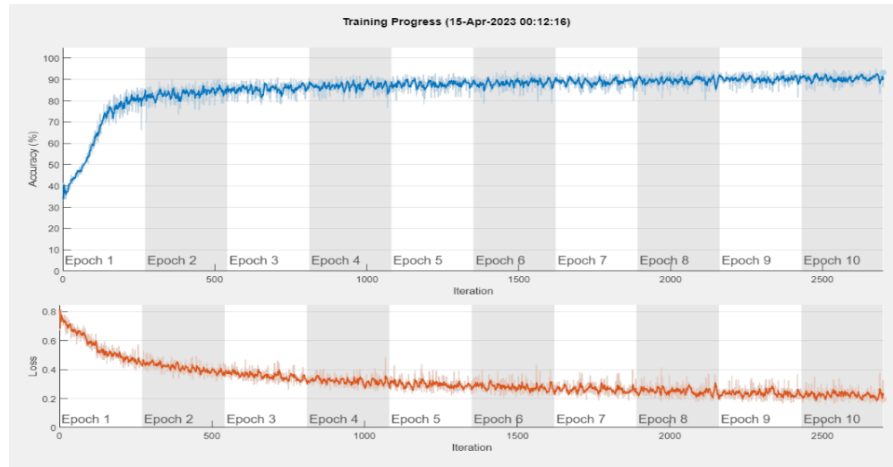


Figure 6: Accuracy and loss curves for training VGG19 on 2700 images

4.2 Model Testing

During the testing phase, sample datasets were selected to evaluate all trained models. For each query image, the salient object was detected from the output of the deep model's last layer. Post-training model testing was performed using three datasets: (ECSSD, MSRA-B, and THUR-15k) to test the performance of the models and the accuracy of the final map.

4.3 Models Combining

Combining trained VGG16, VGG19 models to obtain more accurate results. The outputs of the two models trained on 1160 images, 2700 images, and the features obtained were combined. Two methods were used to combine them: using the average output features (AvgVGG1619), and using the maximum output features (MaxVGG1619), and then obtaining the features map that combines the values and gives the best performance.

4.3.1 Average of Output Features (AvgVGG1619)

The two VGG16, VGG19 networks trained on 1160 as well as those trained on 2700, were combined by taking the average output features for all pixels in final map locations for both networks. This process produced the combined networks: AvgVGG1619(1160) and AvgVGG1619(2700).

4.3.2 Maximum of Output Features (MaxVGG1619)

The two VGG16, VGG19 networks trained on 1160, as well as those trained on 2700, were combined by taking the maximum output features for all pixels in final map locations for both VGG16 and VGG19. The two combined networks MaxVGG1619(1160) and MaxVGG1619(2700) were obtained from the combination process.

5 DISCUSSION

5.1 Quantitative Comparison of the Models for SOD

To identify the best model for SOD. The two models of VGG16 and VGG19 were trained on a MSRA-10K dataset and later tested using three well-known datasets (ECSSD, MSRA-B, and THUR-15k).

5.1.1 Results of Training the Models on 1160 images

Table (4) illustrates the values of the evaluation metrics for the two VGG models trained on 1160 images. These values demonstrate that the VGG16 performs better in general and has a lower MAE rate, although the VGG19 outperforms when using the THUR dataset.

Table 4: Comparison of quantitative scores for VGG16,VGG19 trained on 1160 images

Metrics \ Models	VGG16 (1160)			VGG19 (1160)		
	ECSSD	MSRA	THUR	ECSSD	MSRA	THUR
Precision	0.8126	0.8386	0.7252	0.8258	0.9128	0.7652
Recall	0.7419	0.6061	0.6718	0.6080	0.5038	0.6844
F-measure	0.7951	0.7704	0.7122	0.7627	0.7688	0.7449
S-object	0.8645	0.8408	0.8420	0.8491	0.8362	0.8555
S-region	0.6630	0.6274	0.5843	0.5673	0.5940	0.6153
S-measure	0.7638	0.7341	0.7132	0.7082	0.7151	0.7354
MAE	0.1036	0.1359	0.1276	0.1256	0.1449	0.1151

5.1.2 Results of Training the Models on 2700 images

Table (5) describes the values of evaluation metrics of VGG models when trained on 2700 images and tested on the previously described datasets. The results show that the VGG16 generally achieves superior performance and exhibits a lower MAE, whereas VGG19 outperforms VGG16 only on the THUR dataset. It is observed that both VGG models have benefited from increasing the number of training images by increasing the model efficiency and decreasing the MAE. Notably, VGG19 demonstrates a greater reduction in MAE compared with VGG16.

Table 5: Comparison of quantitative scores for VGG16,VGG19 trained on 2700 images

Metrics \ Models	VGG16 (2700)			VGG19 (2700)		
	ECSSD	MSRA	THUR	ECSSD	MSRA	THUR
Precision	0.8151	0.8192	0.7033	0.7931	0.8888	0.7506
Recall	0.8144	0.7213	0.8177	0.7725	0.5552	0.8135
F-measure	0.8149	0.7943	0.7267	0.7883	0.7805	0.7642
S-object	0.8691	0.8570	0.8512	0.8578	0.8435	0.8625
S-region	0.7198	0.7277	0.6251	0.6777	0.6130	0.6793
S-measure	0.7945	0.7924	0.7382	0.7678	0.7283	0.7709
MAE	0.0898	0.1171	0.1157	0.1039	0.1325	0.1004

5.1.3 Results of Combining the Models (AvgVGG1619)

Table (6) shows the evaluation metrics for the two models after combining them using the average output features (AvgVGG1619), the results generally show relatively better results.

Table 6: Comparison of quantitative scores for combining the VGG using AvgVGG1619

Metrics \ Models	Avg VGG1619 (1160)			Avg VGG1619 (2700)		
	ECSSD	MSRA	THUR	ECSSD	MSRA	THUR
Precision	0.8189	0.8712	0.7453	0.8039	0.8476	0.7259
Recall	0.6753	0.5553	0.6784	0.7931	0.6380	0.8155
F-measure	0.7806	0.7701	0.7287	0.8014	0.7879	0.7448
S-object	0.8706	0.8508	0.8619	0.8774	0.8646	0.8707
S-region	0.6551	0.6540	0.6356	0.7398	0.7215	0.6833
S-measure	0.7629	0.7524	0.7488	0.8086	0.7931	0.7770
MAE	0.1146	0.1404	0.1214	0.0969	0.1271	0.1080

5.1.4 Results of Combining the Models (MaxVGG1619)

Table (7) shows the evaluation metrics obtained by combining VGG16 and VGG19 using max outputs (MaxVGG1619), The results generally demonstrate superior performance compared with the individual VGG16 and VGG19.

Table 7: Comparison of quantitative scores for combining the VGG using MaxVGG1619

Metrics \ Models	Max VGG1619 (1160)			Max VGG1619 (2700)		
	ECSSD	MSRA	THUR	ECSSD	MSRA	THUR
Precision	0.8126	0.8401	0.7308	0.8064	0.8121	0.7120
Recall	0.7442	0.6279	0.7377	0.8347	0.7217	0.8731
F-measure	0.7957	0.7793	0.7324	0.8128	0.7893	0.7437
S-object	0.8609	0.8442	0.8519	0.8623	0.8564	0.8529
S-region	0.6807	0.6709	0.6476	0.7224	0.7280	0.6617
S-measure	0.7708	0.7576	0.7498	0.7924	0.7922	0.7573
MAE	0.0965	0.1225	0.1053	0.0849	0.1054	0.0877

5.1.5 Comparison of MAE Values for Models Tests

Figure (7) shows the MAE values obtained when testing the VGG16 and VGG19 models on the ECSSD, MSRA-B, and THUR-15K datasets. Overall, VGG16 demonstrates superior performance compared with VGG19. The best tests were observed for the ECSSD, followed by the THUR, and then the MSRA. However, the reduction in MAE achieved by VGG19 when increasing the number of training images to 2,700 is greater than that observed for VGG16. Furthermore, a relative improvement is obtained when the outputs of the two models are combined using the AvgVGG1619 method. The lowest MAE value is achieved when combining the two models using the Max VGG1619 method.

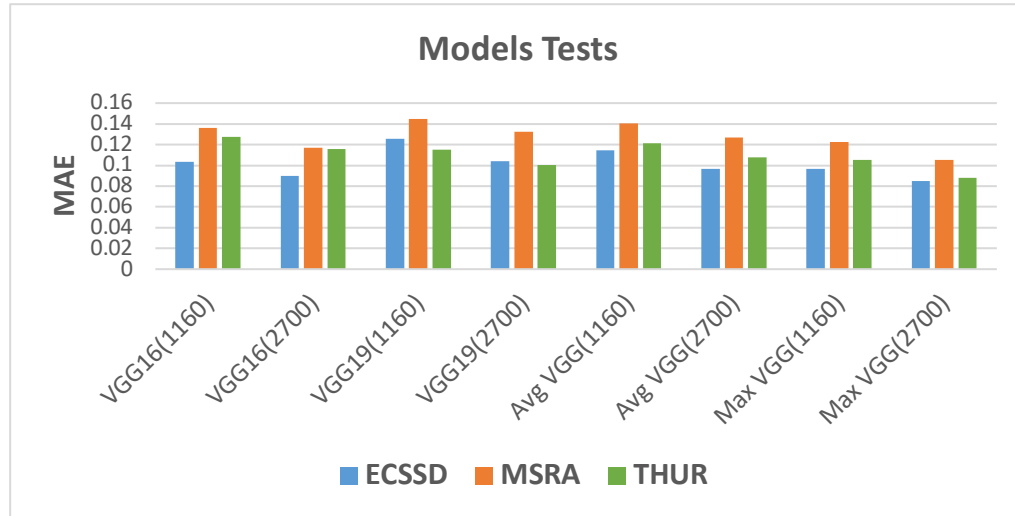


Figure 7: Comparison of MAE Values for Models Tests

5.2 Qualitative Comparison of the Models for SOD

Qualitative results were obtained by evaluating the VGG16 and VGG19 networks trained on 1,160 and 2,700 images, as well as their combined versions using the averaging (AvgVGG1619) and maximum (MaxVGG1619) methods. The evaluation was conducted on sample images from the ECSSD, MSRA-B, and THUR-15K datasets. Figure (8) shows a qualitative comparison of models' evaluation using the indicated datasets, whereas Figure (9) shows a qualitative comparison of models' evaluation using images captured by mobile phone. These results demonstrate the outstanding performance of the proposed models.

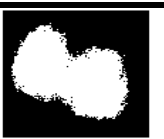
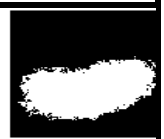
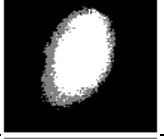
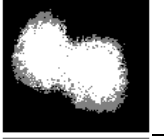
	ECSSD		MSRA-B		THUR-15K	
Image						
Ground truth						
VGG16 (1160)						
VGG19 (1160)						
Avg VGG (1160)						
Max VGG (1160)						
VGG16 (2700)						
VGG19 (2700)						
Avg VGG (2700)						
Max VGG (2700)						

Figure 8: Evaluation qualitative comparison (datasets images)

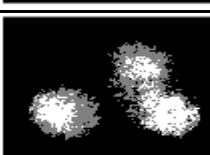
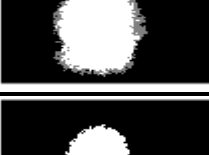
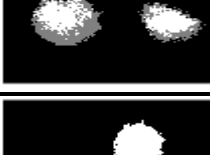
	phone camera images				
Image					
VGG16 (1160)					
VGG19 (1160)					
Avg VGG (1160)					
Max VGG (1160)					
VGG16 (2700)					
VGG19 (2700)					
Avg VGG (2700)					
Max VGG (2700)					

Figure 9: Evaluation qualitative comparison (phone camera images)

6 CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

This study evaluated VGG16, VGG19, and their feature-level fusion strategies for salient object detection using the MSRA-10K dataset for training and ECSSD, MSRA-B, and THUR-15K for evaluation. Results showed that VGG16 generally outperforms VGG19 in terms of accuracy and MAE, whereas VGG19 performs better in scenes without clearly defined salient objects. Increasing the training dataset size consistently improved performance across all evaluation metrics.

Moreover, fusing VGG16 and VGG19 features enhanced detection performance, with the maximum fusion approach (MaxVGG1619) achieving the highest accuracy and lowest MAE across most datasets. These findings demonstrate that dataset scaling and feature-level model fusion significantly enhance CNN-based SOD performance and identify MaxVGG1619 as a robust and effective approach for diverse visual scenes.

6.2 Recommendations and Future Work

Based on these findings, several directions are recommended for future work to further enhance salient object detection performance. Exploring more advanced backbone architectures, such as ResNet, DenseNet, EfficientNet, or Vision Transformers may provide improved feature representation while reducing computational complexity. In addition, integrating multi-scale feature extraction and attention mechanisms could strengthen the model's performance in complex and cluttered scenes. Expanding training to larger and more diverse datasets, including cross-dataset learning and domain adaptation, is also expected to improve the model robustness and generalization. From an application perspective, optimizing the proposed framework for real-time performance and deployment on resource-constrained platforms, such as mobile or embedded systems, is an important future direction. Furthermore, the proposed feature fusion strategy can be extended to related computer vision tasks, including object segmentation, visual attention modelling, medical image analysis, and perception systems for autonomous driving, broadening its practical impact.

In conclusion, this work demonstrates the effectiveness of combining deep CNN models and leveraging larger training datasets to improve salient object detection (SOD). The proposed MaxVGG1619 framework provides a strong baseline for future SOD research and practical applications in real-world computer vision systems.

REFERENCES

- [1]. Abdel-Jaber, H., et al., A review of deep learning algorithms and their applications in healthcare. *Algorithms*, 2022. **15**(2): p. 71.
- [2]. Alzubaidi, L., et al., Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *journal of Big Data* 2021. **8**: p. 1-74.
- [3]. Yamashita, R., et al., Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 2018. **9**: p. 611-629.
- [4]. Al-Saffar, A.A.M., H. Tao, and M.A. Talab. Review of deep convolution neural network in image classification. in *International conference on radar, antenna, microwave, electronics, and telecommunications (ICRAMET)*. 2017. IEEE.
- [5]. Liu, J.-J., et al. A simple pooling-based design for real-time salient object detection. in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019.

- [6]. Zhang, L., et al. A bi-directional message passing model for salient object detection. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- [7]. Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11), 1254-1259..
- [8]. Klein, D.A. and S. Frintrap. Center-surround divergence of feature statistics for salient object detection. in 2011 International conference on computer vision. 2011. IEEE.
- [9]. Xie, Y., H. Lu, and M.-H. Yang, Bayesian saliency via low and mid level cues. *IEEE Transactions on Image Processing*, 2012. **22**(5): p. 1689-1698.
- [10]. Cheng, M.-M., et al., Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 2014. **37**(3): p. 569-582.
- [11]. Perazzi, F., et al. Saliency filters: Contrast based filtering for salient region detection. in 2012 IEEE conference on computer vision and pattern recognition. 2012. IEEE.
- [12]. Liu, T., et al., Learning to detect a salient object. *IEEE Transactions on Pattern analysis and machine intelligence*, 2010. **33**(2): p. 353-367.
- [13]. Jiang, H., et al. Salient object detection: A discriminative regional feature integration approach. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.
- [14]. Li, G. and Y. Yu. Visual saliency based on multiscale deep features. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
- [15]. Wang, L., et al. Deep networks for saliency detection via local estimation and global search. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
- [16]. Zhao, R., et al. Saliency detection by multi-context deep learning. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
- [17]. Lee, G., Y.-W. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [18]. Liu, N. and J. Han. Dhsnet: Deep hierarchical saliency network for salient object detection. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [19]. Li, G. and Y. Yu. Deep contrast learning for salient object detection. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [20]. Kruthiventi, S.S., et al. Saliency unified: A deep architecture for simultaneous eye fixation prediction and salient object segmentation. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- [21]. Hu, P., et al. Deep level sets for salient object detection. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
- [22]. Luo, Z., et al. Non-local deep features for salient object detection. in Proceedings of the IEEE Conference on computer vision and pattern recognition. 2017.
- [23]. Zhang, P., et al., Salient object detection by lossless feature reflection. *arXiv preprint arXiv:1802.06527*, 2018.
- [24]. He, H.-J., C. Zheng, and D.-W. Sun, Image segmentation techniques, in *Computer Vision Technology for Food Quality Evaluation*. 2016, Elsevier. p. 45-63.
- [25]. Ghariba, B., M.S. Shehata, and P. McGuire, Salient object detection using semantic segmentation technique. *International Journal of Computational Vision and Robotics*, 2022. **12**(1): p. 17-38.
- [26]. Simonyan, K. and A.J.a.p.a. Zisserman, Very deep convolutional networks for large-scale image recognition. *arXiv: 1409.1556*., 2014.
- [27]. Zhou, T., et al., RGB-D salient object detection: A survey. *Computational Visual Media*, 2021. **7**: p. 37-69.
- [28]. Tammina, S., Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP)*, 2019. **9**(10): p. 143-150.
- [29]. Yan, Q., et al. Hierarchical saliency detection. in Proceedings of the IEEE conference on computer vision and pattern recognition. 2013.
- [30]. Rother, C., V. Kolmogorov, and A. Blake, " GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 2004. **23**(3): p. 309-314.

- [31]. Achanta, R., et al. Frequency-tuned salient region detection. in 2009 IEEE conference on computer vision and pattern recognition. 2009. IEEE.
- [32]. Fan, D.-P., et al. Structure-measure: A new way to evaluate foreground maps. in Proceedings of the IEEE international conference on computer vision. 2017.