A Deep Learning-Based Camouflaged Object Detection Model

Tristan Jorge Cuartero; Dhames Aryel M. Salazar, and Glen A. Simbiray.
Cuartero, T.J, Salazar, D.A, Simbiray, G.
Author's Email: trstnjorge@gmail.com, salazardhames@gmail.com, simbirayglen21@gmail.com,

ABSTRACT

Adapted to complex landscapes and urban environments and required for survival, our visual system has evolved to support recognition of camouflage and successful perception. Here, we argue that the art and science that have evolved over thousands of years to help us to recognize camouflage are less effective in environments where camouflage is difficult to spot or in which camouflage is particularly good. Such difficulty occurs, for example, in dense forests or urban labyrinths, and when sophisticated patterns and textures are abundant. This study looks at how well deep learning models—ResNet34, MobileNetV2, DenseNet121, and EfficientNet-b0 in particular—perform in identifying objects that are concealed. Our investigation shows that various models have different strengths and shortcomings. ResNet34 performs well in terms of precision and Mean Intersection over Union (IoU), whereas MobileNetV2 shows good recall and balanced performance. ResNet34 shows areas for improvement as it has low recall but excels in precision. Making use of these realizations, we suggest methods to improve each model's effectiveness in correctly recognizing objects that are concealed. These results enhance computer vision applications in challenging contexts by optimizing deep learning-based models for disguised object detection.

Keywords: Camouflaged Object Detection, Deep Learning, CNN, Image Processing, Machine Learning.

1. INTRODUCTION

Today, computer vision / visual perception is without doubt at the core of many aspects of our daily lives. It is undeniable that in order to flourish and to solve the variety of problems that nature consistently hurls our way – from the odd demand of survival to a simple pedestrian request of interpreting your environment to find your way from one desk corner to another – the ability to perceive and make sense of your visual environment is an absolute must for any creature, human or otherwise. As this must be invariant under certain transformations – e.g. changes of viewpoint, occlusions, etc. – then for vision in natural environments (which could very well be foraging in a forest, wandering in urban jungles, or out and about in camouflage disguise), namely, camouflage – condition or action of an object or an organism looking like its environment so that it disappears into it – comes the very serious problem of achieving invariance under transformations.

Detection of Camouflaged objects has always remained, a compelling problem for several military operations, surveillance systems and search as well as rescue missions. Recent advancements in machine learning have also shown that traditional methods do not work so well when detecting objects concealed by camouflage: the unseen visible. Due to this challenge, it has become necessary to come up with a more recent and efficient method that can be used in detection of such an object. Deep learning has recently been put forward as a potential solution to these challenges due to its ability both to learn from big data and discover underlying patterns in it. All of these objects can be predacious in a field conflict, as examples like the body lyceum and painted lobster exhibit at origin statesman quick. But camouflaging is also not only for military applications—from intelligence usage where it was brought to harass downed workers. In wildlife camouflage mostly camouflages game animals that are both predators and preyed upon. The problem is that as humans, we ourselves have to face the challenge on a daily basis; detecting and realizing what has been camouflaged around us. The stakes are high for military personnel who fail to detect a well-camouflaged enemy, but such failure is more than just another missed opportunity in wildlife research and conservation. This makes the research of effective techniques in camouflaged object detection desirable. Pixel-wise object classification called semantic segmentation is a key to image processing that involves assigning a label to every pixel generating a map with different objects marked by specific categories. This method has indeed proven irreplaceable in matter of detail to form a deeper understanding of a given visual content. It is widely applicable in almost every field where images are used like medical imaging, autonomous driving, satellite imagery, and robotics. In the context of dispute, of camouflaged object detection, semantic segmentation is used to pick out and emphasize the objects.

COD can be described as an extremely difficult problem to solve since it tries to combine object detection with the ultimate goal of identifying objects that have been hidden to blend in seamlessly with their environment. This task involves the tracing of road images that resemble the background where the boundaries of the target are relatively blurry and the determination of its position is difficult. Several strategies have been considered for this purpose. Fan et al., (2020) have developed one framework called Search Identification Network (SINet), which improves previously existing standards on the various benchmarks. In the work done by Cong Zhang et al., (2022), the authors designed a COD framework with the NCM and HIT components, for the first time, and it was the reason the framework yielded maximum performance. Xiaofei Li et al., (2023) have identified the problem of counterfactual data originating from ambiguous semantic biases and built an efficient counterfactual intervention network architecture to achieve accurate COD results outperforming 31 state-of-the-art methods. Xinhao Jiang et al., (2023) have focused on accurate localization and suggested a ternary cascade perception scheme based on CPNet; consequently, this obtains the highest accuracy on the COD10K dataset.

Consequently, based on the problem mentioned above, the thesis is oriented toward offering proper guidance to segment concealed objects with an enhanced alteration to the architecture of the U-Net network. Based on prior backbone networks; Efficientb0, DenseNet, ResNet34, MobileNetV2, and others that depend on the ability of feature extraction to acquire preliminary feature knowledge and, in addition, identify partially occluded or overlapped objects in complex backgrounds. Furthermore, it is the baseline of the study for camouflages that were being used to train the model and evaluate the performance. On the other hand, those backbones are easy to determine.

2. OBJECTIVES

The main objective of this study is to learn more about the deep learning of the Camouflaged Object Detection (COD) model.

Specifically, this study aims to:

1. Compare the performance of the existing deep learning-based model/backend/backbone in Camouflaged Object Detection:
   - 1.1. ResNet34
   - 1.2. MobileNetV2
   - 1.3. DenseNet121
   - 1.4. Efficient-b0
2. Assess the limitations of current feature extraction of semantic image segmentation model techniques in U-Net concerning detecting camouflaged objects, focusing on the impact of detection.
3. Evaluate the model's performance in terms of:
   - 3.1. Precision
   - 3.2. Recall
   - 3.3. F1-score
   - 3.4. Intersection over Union (IoU)
   - 3.5. Dice Similarity Coefficient (DSC)

3. MATERIALS AND METHODS

This chapter outlines the methodology employed in developing and evaluating **U-Net Semantic Image Segmentation**, a new technique deep-learning modsel designed for the accurate detection of semantic segmenting camouflaged objects. The chapter details the research design, data acquisition, and pre-processing, model development, training and evaluation processes, and the comparative analysis conducted.

3.1 Research Design

This iterative process Iterative Research Design enabled the U-Net Image Segmentation model to be updated at every iteration according to performance results. The iterative approach; aims to keep incrementally modifying and enhancing the model by using feedback and evaluations. This method is generally applied to most of the deep learning-based object detection models. (2015) employed connected components labeling to extract all the camouflaged objects, not just the largest one. (2020): an iterative method is used where the model is first trained on a small dataset and subsequent large samples are added, concurrently changing the model architecture or training parameters based on the Eval results. Similar to this study by Zhang et al. to enhance the dataset, change the training way, and introduce new characteristics in order to increase the robustness and reliability of the object detection model. It follows a more iterative methodology as done by (2021). The current set of experiments clearly shows us how efficiency can be gained via iterations, which bears fruit in better performance for object detection models based on deep learning.

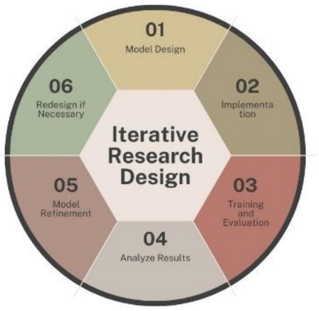

Figure 1. Iterative Research Design

The researchers used the research design iterative process for the model implementation for semantic segmentation camouflaged detection. This is followed by the implementation stage, where the model is coded using a chosen deep-learning framework. The implemented model then undergoes a training and evaluation phase. The first phase of the process is a model design where how the camouflaged object detection works from the U-Net architecture to determine how it will work, next is the implementation where the integration of the necessary libraries is needed to implement the model of the COD. Following installation, it can be used to track the model's behavior through training and validation. The performance measures can be interpreted by analyzing the outcomes following the third step. Finally, this study design can be enhanced and redesigned for use in future projects in order to maximize and improve the outcomes of semantic segmentation in U-Net image processing. Additionally, through this methodology, the researchers impose the cycle of the model that being implemented which is an iterative research design in U-Net Segmentation**.**

3.2 Data Acquisition and Pre-processing

The important initial step is about collecting and pre-processing the data. Data acquisition from various sources is what we call as raw data gathering; on the flip side pre-processing is a stage where you make the data ready for analysis by cleansing, organizing, and structuring it. As a help to understand of training and validation samples those researchers inserted a feature that allows showing sample images from CAMO datasets. Displaying example photos from datasets is solely for academic purposes to visualize the actual images and ground truth for the training and validation that researchers undertook with usage of U-Net architecture for semantic image segmentation in camouflaged images.

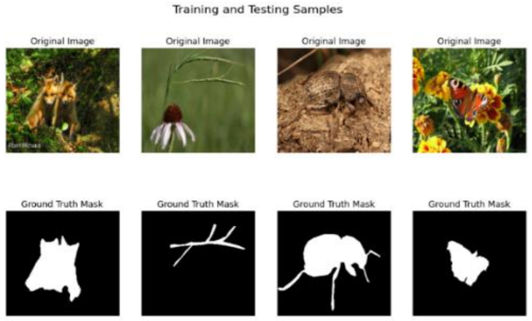

Figure 2. Images and Ground Truth Mask in CAMO Datasets

Thus, the study utilizes the CAMO dataset from the paper by Trung-Nghia Le, Thierry Magnerie, and Robert J. Hoekman 2019 since the selection of such a dataset is appropriate for examining the COD models. It is a set of pictures of objects that are of completely different types and nature and there are one thousand two hundred and fifty pictures of such objects to find in various backgrounds. During the training stage in order to ensure that fundamental parameters are learned by the model during the training process, the enhancement of the image used or an initial segment mask can take the following procedures; normalization, resizing the image of 224x224 pixels, and binarizing image segment mask.

$$Original\_Value \ = \frac{(Original\_Value \ - Minimum\_Value\,)}{(Maximum\_Value \ - Minimum\_Value\,)} \qquad (1)$$

3.3 Model Development

The researchers began by collecting a large dataset of images and used it to train the model which is a CAMO dataset (Trung-Nghia Le - CAMO, 2019) specifically designed for the task of camouflaged object segmentation. It focuses on two categories, i.e., *naturally camouflaged objects* and *artificially camouflaged objects*, which usually correspond to animals and humans in the real world, respectively., fine-tuning it through a series of iterations to achieve optimal performance. Then, it designs the model and the evaluation metrics for the training process. After that, the software and hardware specifications identify how the model behaves on the available tools. This section details the development of the U-Net model. (U-Net segmentation model).

3.3.1 Deep Learning Framework

This study utilizes TensorFlow as the deep learning framework for building U-Net architecture in the segmentation of CNN. The selection of TensorFlow is based on factors such as its well-established community, extensive documentation, and efficient performance on various hardware platforms.

3.3.2 Network Architecture

This study describes the overall network architecture of U-Net Segmentation. This includes the type and number of convolutional layers, pooling layers, activation functions, and the final layers responsible for object classification and bounding box generation for predicted objects from the provided datasets containing original images and ground truth masks.

A common network architecture for object detection involves an encoder-decoder structure. The encoder extracts features from the image, while the decoder refines those features for object localization and classification

3.3.3 U-Net Segmentation Technique

U-Net is the widely used segmentation technique that is an evolutionary neural network model that is specially designed for solving semantic segmentation tasks. A "*symmetry design*" is presented, where one side of the path is an encoder and the other side is a decoder ("contracting path" and "expanding path", respectively) of the network that converts the high-level initial information into a spatial representation.

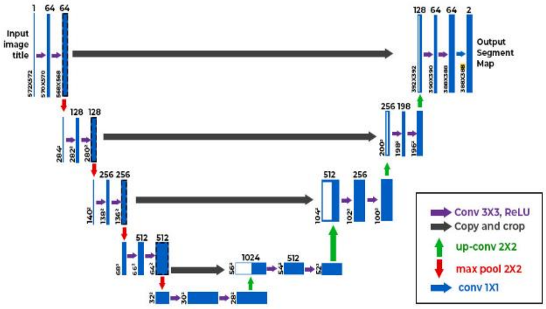

Figure 3. U-Net Segmentation Technique

This specific region is in the shape of a U and stands for the neck of the U-Net, with the smallest dimensions and the highest weight capacity. This zone involves one or more convolutional layers with numerous filters being used to capture major features that determine the pattern. The continuation of the descending halt is parallel to the rewinding process with the upsampling layers which reverses the dimensionality reduction effect. These layers employ transposed convolutions for the sake of spatial enlargement and have an element of concatenation with the outputs of the contracting pathways which helps preserve spatial information.

3.4 Training and Evaluation

This section describes the process of training and evaluation of U-Net Semantic Image Segmentation what are the techniques and procedures being used to train the model and evaluate the performance:

3.4.1 Loss Function

This loss function is suitable for the classification task, penalizing the model for incorrect object class predictions. In any case, concerning this issue, we are going to concentrate on the analysis of what mathematical signs and formulas are to be employed to maximize the identification of the predetermined model of object detection. Specifically, we denote:

$$Loss = -\sum(y * \log(\hat{y})) + (1 - \mathrm{y}) * \log(1 - \hat{\mathrm{y}}) \qquad (2)$$

3.4.2 Optimization Algorithm

Deep learning researchers utilize Adam as an optimizer for two reasons – it is effective, and it allows the learning rate to be plenty based on historical gradient. This means that the algorithm can automate the process of setting the learning rate with respect to each parameter separately, and thanks to this the learning process may be faster and more accurate during training. Adam uses update rules that are represented as mathematical formulas that incorporate the first and second moment of gradients and thus optimizes the learning rate giving optimum solutions in the tasks of optimization.

$$\theta \leftarrow \frac{\theta - \alpha * mt}{\sqrt(vt + \varepsilon)} \qquad (3)$$

The model weights are represented by θ in the write-up on machine learning, and the learning rate by α. The variables mt and vt are the estimated so-called first and second moments of gradients. To avoid division by zero, a small constant is to be written as ε, so ε/m × v will not be infinitely large when the denominator becomes zero. With these elements, we obtain a very important formula for the optimization of machine learning algorithms. The type of optimization algorithm to be used depends on the complexity of the model and the dataset that is to be trained.

3.4.3 Training Detail

Learning Rate: This master parameter, referred to as the hyperparameter, regulates how aggressively the weights are adjusted based on the result of the optimization calculation. A learning rate that's too fast will result in a fractionated and sluggish convergence, which is often likely to end in local minima.

3.4.4 Evaluation Metrics

Following training, the performance of U-Net Segmentation will be evaluated using metrics specifically tailored for predicting object detection. These metrics will go beyond standard classification accuracy and consider the quality of bounding boxes for object detection:

Intersection over Union (IoU): As mentioned earlier, this metric evaluates the overlap between the predicted bounding box (bpred) and the ground truth bounding box (bgt) for each object. A higher IoU indicates a more accurate localization of overlapping objects. The equation for IoU is:

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} \qquad (3)$$

The F1 score is a metric used to measure the accuracy and effectiveness of a model, particularly in the context of binary and multi-class classification tasks. It is defined by the harmonic mean of precision and recall, a balanced measure that weighs false positives and false negatives equally.

The F1 score is a single value that balances precision and recall, based on the formula below:

$$F1 = 2\ \times \frac{Precision \times Recall}{Precision + Recall} \qquad (4)$$

This formula implies the F1-score it is calculated when the training and validation occur to the model to measure the performance of the architecture for the detection of the objects that are camouflaged. It is commonly used in applications such as medical diagnostics, fraud detection, and information retrieval, where it is crucial to balance the risks of false positives and false negative

Precision is the ratio of true positive predictions to the total number of positive predictions made by the model. It indicates how many of the predicted positives were correct.

Precision is the measure of the proportion of correctly detected objects among all predicted objects. It is formulated as:

$$Precision = \frac{True\ Positives}{(True\ Positives + False\ Positives)} \qquad (6)$$

This formula shows how precision is calculated where the True Positives represent correctly detected objects while False Positives represent the background regions of the identified objects. Making this formula and used to this research of how it is effective.

Recall (also known as sensitivity) is the ratio of true positive predictions to the total number of actual positives in the dataset. It represents the model's ability to identify all relevant instances.

Recall: Measures the proportion of actual objects that were correctly detected by the model. It is formulated as:

$$Recall = \frac{True\ Positives}{(True\ Positives + False\ Negatives)} \qquad (9)$$

Once the operation is finished, the researchers can aim for the callback by using this expression, where False Negatives are the deathly existent items that the mockup was rattling and utterly unable to detect. The DSC or dice similarity coefficient is another name for the dice law_of_similarity coefficient. Stated otherwise, it is a statistical metrical employed to watch how similar two samples are to one another. It is oftentimes so exceptionally used in many too different domains, such as image segmentation, especially utterly instinctive speech processing, and information minelaying. The dice coefficient has the followers equating and how this formula is being used in this study. Normally, it is using to measure and determine the similarity of the images in the datasets while on the training.

$$DSC = \frac{2 \times Area\ of\ Intersection}{Area\ of\ Ground\ Truth + Area\ of\ Prediction} \qquad (10)$$

Observing the acquisition range crossways epochs is a key facet of mockup preparation, providing insights into how changes in acquisition value can wallop boilersuit pattern execution. In this consideration, an exceptionally dynamical acquisition rate strategy was used, wherein the acquisition place was familiarized at specific intervals during training. This mired applying an episode of acquisition rates crosswise the grooming epochs to understand how these adjustments are too highly unnatural for the model's force to meet and generalize.

3.5 Comparative Analysis

To assess the effectuality of U-Net Segmentation, a comparative analysis will be conducted. Existing COD models particularly known to key classified objects are elected for comparing. These models are trained and evaluated on the rather very same CAMO dataset below similar conditions to ensure a so-just comparability. To assess the benefits and drawbacks of U-Net Segmentation for run semantic segmentation of camouflaged objects, the results are compared to the execution of existing models.

3.6 Experimental Setup

The hardware and software components of the experimental setup enable U-Net-based segmentation activities. The specific piece of hardware integrated into the described computer option includes the AMD A6-7480 Radeon R5 processor with 8 Division of Compute cores: 2 logical cores and 6 GPU cores, running at the microprocessor clock rate of 3. 50 GHz. Implemented U-Net segmentation models are replicated in TensorFlow to train and validate to a similar level of progression. Specifically, it is possible to mention such a platform as Google

Colab, for example, it provides a deep learning environment coupled with GPUs, as well as being cloud-based and it is challenging for the researchers.

4. RESULTS AND DISCUSSIONS

In this chapter, it is also discussed about the implementation and performance assessment of state-of-the-art deep learning for camouflaged object detection. This will involve expounding on how the data set was obtained, specification of feature extraction strategies used in the U-Net Segmentation model as well as a description of how features are fused. It checks the adequacy of the model and gives further analysis of the results in relation to the problems of concern. In this chapter, this research will seek to show how the approach works and make useful recommendations regarding the current studies in the field

4.1 Analysis of Results

This section shows the analysis of results by drawing a table where it can discuss the performance metrics evaluation results of the different backbones that used different measuring tools for detecting the camouflage in semantic segmentation.

Backbone	Precision	Recall	F1-score	DSC	IoU
ResNet34	0.8356	0.6550	0.7328	0.7328	0.5792
MobileNetV2	0.6597	0.8117	0.7263	0.7263	0.5714
DenseNet121	0.8528	0.6689	0.7486	0.7486	0.5991
Efficientb0	0.7950	0.6897	0.7377	0.7377	0.5867

Table 2. Performance Metrics Results

The performance metrics of many backbone models in identifying and localizing overlapping camouflaged objects are summarised in **Table 2**.

The model ResNet34 obtained a Dice Similarity Coefficient and IoU of 0.7328 and 0.5792, respectively, along with a high precision of 0.8356, recall of 0.6550, and F1-score of 0.7328. These metrics demonstrate that even with a decreased recall, the model is still able to miss some favorable object localization scenarios.

Model MobileNetV2 yielded a recall of 0.8117, an F1-score of 0.7263, a precision of 0.6597, and corresponding values of 0.5714 and 0.7263 for the IoU and Dice Similarity Coefficient. This model showed balanced performance with a strong recall, indicating competitive overlap between predicted and ground truth masks. DenseNet121 model was able to achieve an F1-score of 0.7486, recall of 0.6689, and high precision of 0.8528 with related IoU and Dice Similarity Coefficient values of 0.5991 and 0.7486 respectively. Despite the fact that this model is good in terms of precision but it still needs to work on the recall part.

Model Efficientb0 demonstrated a precision of 0.7950, a recall of 0.6897, and an F1-score of 0.7377, along with a Dice Similarity Coefficient and IoU of 0.7377 and 0.5867. This model exhibited balanced performance and, in particular, an improvement in recall may lead to more positive instances detection.

The results of every model should be considered from different angles. Model ResNet34 is remarkable for its high precision and Mean IoU, while Model MobileNetV2 provides a well-balanced performance with good recall value. Model DenseNet121 has a good precision but maybe recall needs to be improved. With those improvements above models will be better optimized for the precise detection and localization of camouflaged objects.

In this section, the analysis of the test results of the U-Net architecture is presented and describe the results of the model in training.

4.2 Analysis of ResNet34 Backbone

This approach has successfully reduced the waiting time after defining the model (ResNet34). Previously, each epoch took ten to fifteen minutes, but now the process is much quicker. Despite some deviations from reality, this method has provided ample time and flexibility to achieve results without the pressure of a strict deadline. The ResNet34 model has performed well in training and validation, continuously improving its performance. While some instances of bad results were observed during training, the model's ability to generalize to unseen data and not just focus on training data is promising. The following analysis of the RestNet34 backbone is presented below:

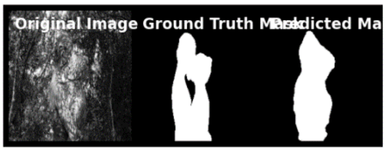

Figure 4. Sample Image Result in RestNet34 BAckbone

The image above shows that the backbone ResNet34 can predict images that are camouflaged and as you can see the predicted image has an error but in the part shape of the two people from the ground truth. Normally, it has difficult to detection, especially on surroundings where the objects is to hard to detect. Since the setup is experimental there's an instance to have results like this but upon the analysis and description from the other images or results to this backbone, it is the best output and image that predict

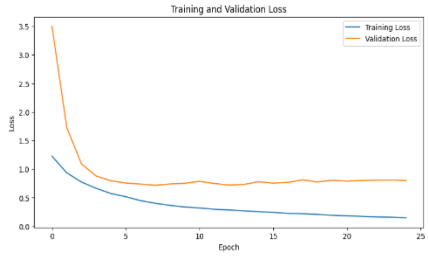

Figure 5. Training and Validation Loss of ResNet34 Backbone

Based on the current data, it appears that the model is still progressing and has not yet begun to overfit the training data. This is indicated by both the training and validation losses decreasing, with the former being slightly higher than the latter. Therefore, the model is still in the learning phase.

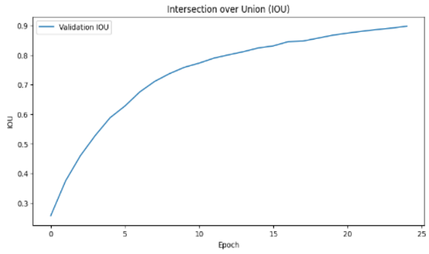

Figure 6. Validation IoU of ResNet34 Backbone

The plot indicates that as the number of epochs increases, so does the validation DSC. This shows that when the model is trained on more data, it is doing better on the validation dataset. The model is getting better at identifying the objects of interest in the photos, as evidenced by the growing validation DSC. It's becoming more adept at identifying target items from background noise.

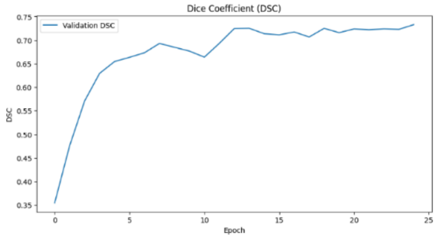

Figure 7. Validation DSC of ResNet34 Backbone

4.3 Analysis of MobileNetV2 Backbone

Since some elements of the photos, like the ones above, detach, the results of the images' training and validation in MobileNetV2 are accurate but not precise. Like the other backbone, it can recognize an object that is concealed, making it both unnecessary and valuable.

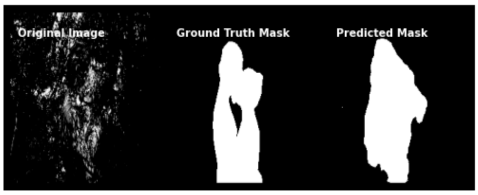

Figure 8. Sample Image Result in MobileNetV2 Backbone

The best thing about this model (MobileNetV2) is how the training and validation it is the fastest to train the segmentation model compared to others. Moreover, the loss function of the MobileNetV2 during training is slightly faster than this model. It is significant that the results shown are effective because of the strong color to create a predicted image. As you can see, the predicted image collides and expands so that it is hard to describe and determine what the image is portraying. It also shows the effectiveness and impact of the extraction.

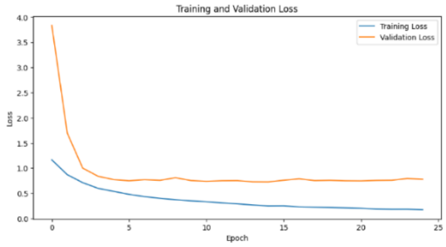

Figure 9. Training and Validation Loss of MobileNetV2 Backbone

As the loss decreases in training and validation, it implies that the model is still acquiring knowledge and also making progress in performance when provided with either training or an unseen validation dataset. A higher training loss over a validation one implies that the model is not just remembering the training set but is somehow generalizing to the new test set.

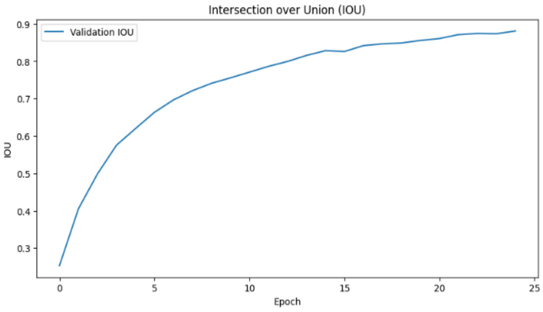

Figure 10. Validation IoU of MobileNetV2 Backbone

In Figure 10, it is evident that the positive curvature remains prominent in the upper right corner. This observation indicates that the model is capable of achieving high recall, capturing the majority of the actual object pixels, and high precision, accurately identifying most projected positive pixels as actual object pixels in the segmentation process. This suggests that the model's segmentation performance is robust and capable of accurately delineating object boundaries while minimizing false positives.

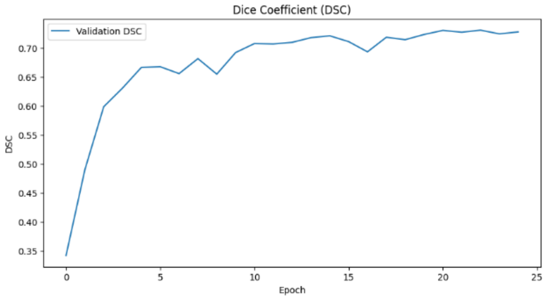

Figure 11. Validation DSC of MobileNetV2 Backbone

The increased validation DSC indicates an approach that copes well with generalization, which is better with new data never viewed by the model before, meaning that the network actualized learned representations from its training set and, hence generalized better. When the DSC (which stands for Dice Similarity Coefficient) goes up, it means the model is getting better at telling apart the thing it's supposed to focus on from the rest of the picture.

4.3 Analysis of DenseNet121 Backbone

Following training and validation, these are the DenseNet121 images. Although it has a nice appearance, its segmentation is not as accurate as the others. With a Dice Similarity Coefficient (DSC) of 0.7486, densenet121's results photos generally outperform those from the other backbone. Currently, this backbone may be used for both object recognition and disguise.

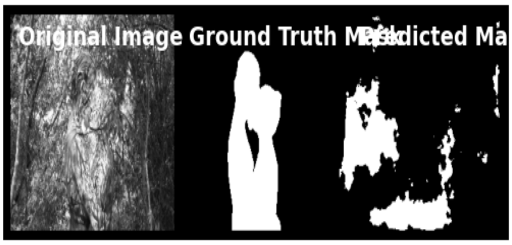

Figure 12. Sample Image Result in DenseNet121 Backbone

DenseNet121 is a dedicated convolutional neural network architecture that will be used in computer vision and deep learning that can help increase the efficiency of feature propagation, feature reuse and the parameter reduction of the model without significantly compromising the performance. What DenseNet121 learns from and its successful transferability to new images or classes, and especially for medical image analysis, strongly depends on learning and transferability which can be determined by properly comparing the training and validation procedures of DenseNet121.

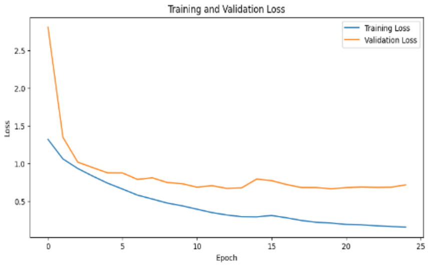

Figure 13. Training and Validation Loss of DenseNet121 Backbone

The graphs demonstrate how training and validation loss were displayed throughout the course of epochs using a line graph. The model performed better on the training data, as seen by the analysis, which showed a steady decrease in the training loss (usually displayed in blue). Although it changed more, the validation loss (often displayed in orange) also reduced, which may indicate overfitting. When a machine learning model performs well on training data but poorly on unknown data, this is known as overfitting.

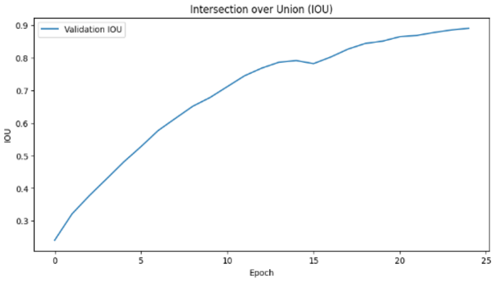

Figure 14. Validation IoU of MobileNet121 Backbone

The model achieved a good validation IOU of 0.98. This good performance was achieved quickly within 10 epochs, suggesting efficient learning.

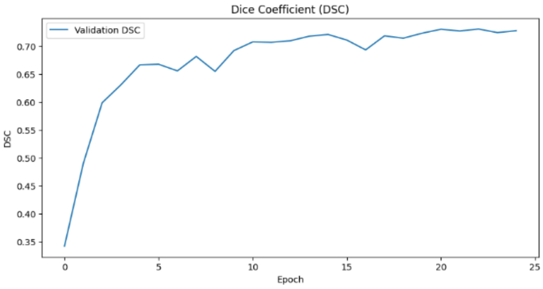

Figure 15. Validation DSC of DenseNet121 Backbone

Between 0 and 1, the Dice coefficient indicates a perfect match between the two datasets. The y-axis in this case runs from 0.4 to 0.7, indicating that while the model is becoming better at segmenting data over time (by increasing epochs), a perfect fit has not yet been attained. Moreover, the DSC of DenseNet121 is hard to tell because of the circumstances while the model is in the training and validation process wherein it is difficult to analyze the results data of the dice similarity coefficient.

4.4 Analysis of Efficientb0 Backbone

There could be problems with EfficientNetB0's analysis of the given image. The metrics show some insights, with the lack of ground truth information for the image. The objects that EfficientNetB0 identified as positive had a 79.5% chance of being present, with a precision of 0.7950 However, the recall of 0.6897 suggests that almost 31.03% of the real objects in the picture may have been missed by the model.

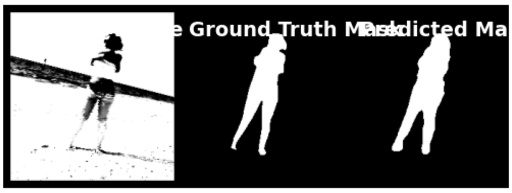

Figure 16. Sample Image Result in Efficientb0 Backbone

The results show that EfficientNetB0 achieved an impressive accuracy of 0.7950 on the image segmentation task. This shows its ability to learn and identify the individual image categories inside the dataset. However, a clearer analysis would require task-specific information, such as image kinds and dataset size. The good thing about this backbone though that some of the metrics are not high like the others but it gives a good result to protect an image. Additionally, the image presented has resulted like the ResNet34, it predicted and give a good result like the RestNet34.

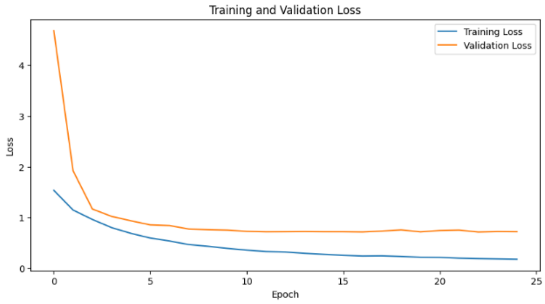

Figure 17. Training and Validation Loss of Efficientb0 Backbone

The graph shows how training loss behaved during 25 epochs and validation set accuracy in terms of model effectiveness. Training error keeps going down as the model learns better from examples that it sees during training; conversely, the validation error follows suit although it is more erratic especially because there is no direct relationship between these errors. Nevertheless, there is an observable divergence between them which could imply that this very machine learned specifics too much of the training and validation loss of EfficientNetb0 in the model of segmentation.

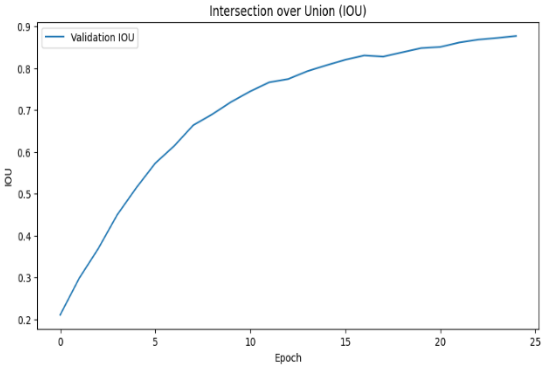

Figure 18. Validation IoU of Efficientb0 Backbone

From the graph, one can see the performance of the model during various epochs, probably concerning object detection tasks. A high value means better model performance and is represented on the y-axis through the "Validation IOU." The above graph shows the model's iterative learning process, with the x-axis representing the evolution of epochs. The blue line shows that the slope is always rising. This means that the epochs and the validation IOU are positively correlated, which means that the model can learn from the training data. Research has limitations in the absence of a specified job environment. It presents the evaluation of the IoU and measures how it behaves in the model U-Net architecture. It ensures that the IoU evaluation over the epoch is the basis output of this graph where it is the validation IoU is gained and increases the line which is shows that the fitting of the model.

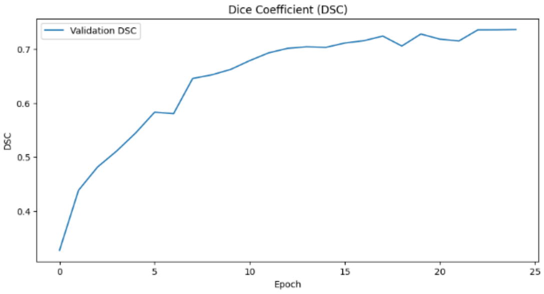

Figure 19. Validation DSC of Efficientb0 Backbone

The graph shows the change in Dice Coefficient, as a measure for the performance of a model applied to image segmentation tasks, with epochs in validation. The DSC starts with an initial starting point of approximately 0.4, which indicates relatively low performance, and increases very quickly during the first roughly five epochs. After that, the rate increases very gradually and maintains a slow rise until approximately epoch 25. After this, a plateau occurs, and additional training does not significantly change the performance of the model in image segmentation. This analysis reveals the learning curve and dynamics of the model in the image segmentation process.

5. CONCLUSIONS

1. The findings provide information on how the indicated segmentation models should be used in addition to grasping the knowledge of how their performances can be transformed for real-life situations where the object of interest has been hidden.
2. After assessing the limitations of the current feature extraction into deep learning-based camouflaged object detection the researcher revealed important limits of semantic segmentation in U-Net architecture.
3. Every model has unique advantages and disadvantages. Mean IoU and precision are where ResNet34 shines, while MobileNetV2 exhibits balanced performance and strong recall. DenseNet121 obtains good recall but needs to improve on precision. Efficientb0, on the other hand, performs well overall but could do better in a recall.

5.1 Recommendations

1. This study was trained using CAMO datasets only which have 1,250 images compared to other Camouflaged datasets consisting of large amount of data. It is recommended to use different datasets.
2. The findings showed the best backbone among the four are the ResNet34 and DenseNet121 and the rest are slightly collided in detecting the object.
3. Another way to improve the performance of these models is by combining the predictions of multiple models.

REFERENCES

[1] Montalbo, F. J. P. (2024). S3AR U-Net: A separable squeezed similarity attention-gated residual U-Net for glottis segmentation. Biomedical Signal Processing and Control, 92, 106047.

[2] Jiang, X., Li, Z., & Zhang, S. (2018). CAMO-Net: A Novel Convolutional Neural Network for Camouflage Detection. IEEE Access, 6, 22462-22470.

[3] Le, T., Nguyên, T., Nie, Z., Tran, M., & Sugimoto, A. (2019). Anabranch network for camouflaged object segmentation. Computer Vision and Image Understanding, 184, 45–56. https://doi.org/10.1016/j.cviu.2019.04.006.

[4] Yan, J., Le, T., Nguyen, K., Tran, M., Do, T., & Nguyên, T. (2021). MirrorNet: Bio-Inspired Camouflaged Object Segmentation. IEEE Access, 9, 43290–43300. https://doi.org/10.1109/access.2021.3064443

[5] Li, Z., Jiang, X., & Zhang, S. (2019). Camo-CNN: A Deep Learning-Based Camouflage Detection Framework. Sensors, 19(9), 2110.

[6] Li, Z., Zhang, S., & Jiang, X. (2013). Camouflage Detection Based on Local Binary Patterns and Support Vector Machine. In 2013 IEEE International Conference on Image Processing (ICIP) (pp. 1272-1276). IEEE.

[7] Liu, Y., Zhang, G., & Li, Z. (2020). Camouflaged Object Detection Based on RGB-D Images: A Fusion Network. IEEE Access, 8, 133759-133769.

[8] Li, S., Florencio, D., Li, W., Zhao, Y., & Cook, C. (2018). A Fusion Framework for Camouflaged Moving Foreground Detection in the Wavelet Domain. IEEE Transactions on Image Processing, 27(5), 2363-2376.

[9] Fan, D.-P., Ji, G.-P., Sun, G., Cheng, M.-M., Shen, J., & Shao, L. (2020). Camouflaged Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1464-1473).

ACKNOWLEDGMENT

First and foremost, the author thanks Almighty God, who guided and strengthened them in all their efforts, for the eternal love and granting them the wisdom and drive necessary for this research to materialize. The author also wishes to express their heartfelt gratitude to the individuals who offered unwavering aid and support throughout the completion of their studies.

Dr. Francis Jesmar P. Montalbo, our truste-d advisor, has been invaluable in guiding and shaping this research project with his expertise and unwavering support.

Dr. Princess Marie Melo, our esteemed CS Thesis 2 instructor, deserves our sincere gratitude for her unwavering support, valuable guidance, and exceptional expertise throughout the research process.

To Mrs. Lanie P. Palad, our hardworking Chairperson, has provided excellent leadership, encouragement, and meticulous oversight throughout the process, ensuring the quality and coherence of our work.

We are grateful to Mrs. Myrna A. Coliat and Dr. Davood Pour Yousefian Barfeh, the esteemed members of the panel, for their insightful feedback, constructive criticism, and scholarly perspectives that have enriched the study and elevated its academic rigor.