Implementation of Deep Res U-Net for Camouflaged Objects

R. Linhares, J. Serdoncillo, V.G. Shenoy
EE 5561: Image Processing and Applications
University of Minnesota - Twin Cities
Minneapolis, United States
caval155@umn.edu; serdo004@umn.edu; sheno051@umn.edu

***Abstract*—This study presents a different approach to camouflaged object segmentation using a 7-level Deep Residual U-Network trained on RGB images and masks from the Camouflaged Objects dataset (CAMO). The network shows to be able to perform the segmentation technique and capture details from the hidden objects in the images as desired. An investigation of the epochs number and the learning rate used is also performed. The epoch number varied from 10 to 100, and the learning rate from 0.01 to 0.0001. Both the increase in epochs number and the decrease in the training rate showed to improve results. Mean Squared Error is used as the loss function in the training process to enhance the model's precision in identifying the camouflaged objects. The results are then compared to the performance of the Mirror Net and the Anabranch Network. Even though the Deep residual network was able to perform the segmentation task, networks that are designed specifically for camouflaged objects such as the Mirror Net, ends up presenting better results due to the implementation of specific techniques suitable for camouflaged object such as the flipping of a copied image. This work was implemented in Python with the PyTorch framework, and contributes to the exploration of camouflaged object segmentation techniques.**

***Index Terms*—ResNet, U-Net, Image Segmentation, Camouflage dataset**

I. Introduction

Image segmentation is an important task that has the goal of separating the image into distinct regions that have a certain meaning for the application. This technique finds a large field of applications, including but not limited to objects recognition, medical imaging, feature extraction and video surveillance.

A common technique used to perform segmentation tasks is called U-net. The U-net structure was first presented by Ronneberger et al. (2015) [6] and the authors used a U shaped structure (Figure 1) that consists in contracting and expanding paths. The contracting paths are responsible for capturing the features and understanding the overall structure of the given image, while the expanding paths are responsible for obtaining the precise location of these features. The extraction of the features is commonly known as encoding and the expanding as decoding. The main idea is that the attributes in the lower layers can be propagated to the higher layers showing a clear path of this specific feature. Thereby, the semantic segmentation is completed by identifying the class of each pixel of the image, which is performed with a final layer with the softmax or the sigmoid function. The choice in between these two functions for the last layer depend exclusively on the segmentation task that is being performed. If there are multiple classes available for defining a pixel the softmax is the function that needs to be used, while the sigmoid is used when we have a binary classification problem.

The U-net also takes advantage of the skip connections, where the output is directly connected to an initial layer. The main purpose of these connections are not only to preserve the spatial information, which is an essential characteristic for segmentation tasks,but also to avoid the vanishing of the gradient of that specific layer. Since then the U-net convolutional network was widely recognized in the scientific community [9], [10], [11], [12].

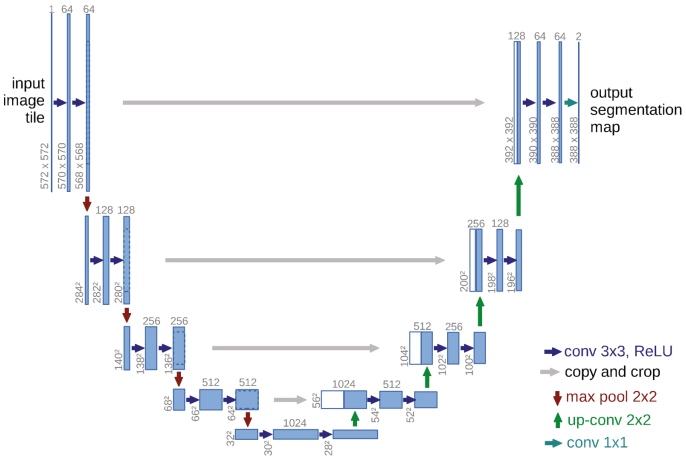

Fig. 1. Architecture of U-Net ([6])

Successfully performing a segmentation task can be the deciding factor between life or death in some applications. In 2020 Zeineldin et al. used Residual Networks (ResNet), Densely Connected Convolutional Networks (DenseNet) and Neural Architecture Search Network (NASNet), which are adaptations of the U-net architecture to perform segmentation of a common brain tumor. The segmentation process was shown to successfully detect the location and shape of the tumor and provide the results in real time during the operation procedure, demonstrating how powerful and important these techniques can be.

The variations of the U-net architecture that were implemented since its appearance exists in order to solve the unique challenges of different segmentation tasks. One of these challenges was the road detection presented by [1]. The road extraction datasets used in their work had the challenges of having a noisy and complex background. The authors implemented the deep residual U-net which contains more layers, increasing the depth, to capture more complex feature attributes. The architecture combined the advantages of both the residual network and the U-net. Just like the U-net, their code allowed the skip connections behavior to skip some layers of the network to allow information to flow better from the input and to the output, however for deep neural networks the vanishing gradient problem becomes much more severe. Deep layers architecture are known to have the problem of the vanishing gradient, where the gradients of the loss function in the initial layers become too small when they are back propagated to the output layer. The residual connections is a key idea for the success of the deep neural networks, where the input of a given layer is directly connected to its output, avoiding the problem of the gradients getting smaller and smaller after the layer transformations.

The capabilities of the deep residual U-net proved to be great for the segmentation of the roads. The network was compared to other convolutional networks as the traditional U-net [6], Saito-CNN [14], Mnih-CNN [13] and some variations of the latest. The deep residual network obtained the higher breakeven point, which is a metric that verify the accuracy of the predictions being performed. These great results for the task of road extraction have shown to be promising and worth investigating its application in other areas and different datasets.

In order to verify how precise a certain residual network is, it is interesting to use challenging datasets, and an interesting data repository was used by [15], [16]. This dataset provides 1250 images that present at least one camouflaged object in each image and the objects were masked using a custom designed interactive segmentation tool. A sample image is presented in Figure 3a, and in some cases it is possible to see even how challenging it is to perceive the differences with the naked eye. 3b shows the corresponding mask for that given image. Some CNNs that were tested in this dataset were Anabranch network [15] and Mirror net [16]. The anabranch network developed by the authors uses a different approach where they combined the segmentation task also with the classification approach. The classification showed to be great in identifying if there was an object camouflaged present in the image. Even though the Anabranch network showed an improved performance compared to existing methods at the time, the architecture based on the two branches added complexity to the network and increased the computational time. The Mirror net that was presented later was and introduced a new idea of flipping a copy of the same image from left to right and then performing the probability maps simultaneously for both of them. This technique showed to provide greater accuracy by breaking the camouflage due to the flipping part, but one of the disadvantages of the method is that as it requires the processing of not only the original image but also the flipped copy, the processing power is doubled and ends up requiring a great computational time.

Fig. 2. (**a**)

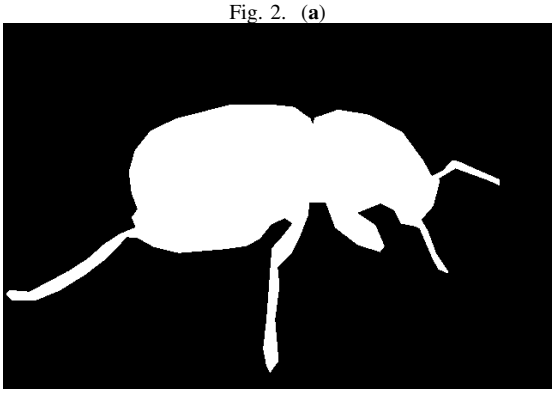

Fig. 3. (**b**)

Fig. 4. Sample image from the CAMO dataset [15], where (a) is the original image and (b) is the mask that will be used during training.

It is seen here that the U-net architecture have many variations that could be suitable for approaching the camouflaged datasets. The U-net combined with the residual networks introduced by [1] has a great potential for capturing complex features in the images and perform an accurate segmentation of the hidden objects, and therefore it is interesting to verify its capabilities in extreme challenging datasets. In this current work the deep residual network is used to perform a semantic segmentation task on a dataset with camouflaged shapes. The accuracy is then compared with other networks reported in the literature and the limitations are discussed. The goal is to successfully identify the camouflaged shapes and show the advantages and disadvantages of the deep residual network for this type of dataset.

II. Methodology

A. Building Blocks

In different image segmentation networks, unique blocks that are added are the Sequential layers as seen in deep neural nets. At the same time, with the discussion of the U-Net, concatenation is done from a layer of the encoder into a layer

in the decoder. An added feature of the proposed U-Net are the residual blocks which adds the residual layer to a layer of the original input in order to get the final output. The general form of the residual block is as shown in the figure below.

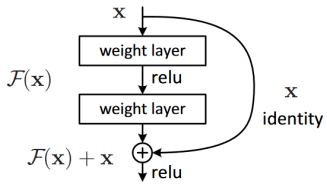

Fig. 5. General Residual Block

B. Original U-Net

The original U-net shown in Figure 1 and the step by step is described here with details about the contractions and expansions performed with this technique. The original image is used as the input and then convolutional blocks are used with ReLU activations functions that are used to extract the features from the input. An important step for the original U-net is the usage of pooling layers that are important for reducing the spatial dimensionality of the feature maps. In the original structure, the contracting path consists in four of these blocks. For the expansion part, other four stages are used to find the final output. These stages consists of convolutional block and transposed convolution operations at the beginning of each stage, which are used to umpsample the input by a factor of two. While expanding, the skip connections from the contracting layer are directly connected to the paths on the expanding process, which helps combining the overall structure of the original image with the new features identified in the contracting path. The final layer of the U-net is another convolutional block that provides a single output channel, that is key for finding the final probability map of the image, stating the likelihood of each pixel belonging to the specific segmentation. For the original U-net, the benchmark to check if the predictions are accurate is performed with a cross entropy loss function. In the current work however, the loss function used was the mean-squared error as presented in [1].

C. Deep Residual U-Net

The deep residual U-Net combined the idea of the skip connections presented by the U-net structure with the idea of the residual connections, to address issues related to vanishing gradients during deep network training as shown in Figure 6.

Similar to the standard U-Net, the Deep Residual U-Net comprises an encoder and a decoder. The encoder captures contextual information through down-sampling layers, consisting of convolutional and pooling operations, while the decoder restores spatial resolution through up-sampling layers. The skip connections between encoder and decoder facilitate the fusion of high-level and low-level features.

The key innovation in the Deep Residual U-Net is the incorporation of residual connections within each double convolutional block. A residual connection is a shortcut that bypasses one or more layers, allowing the gradient to flow more easily during backpropagation. This helps mitigate the vanishing gradient problem, enabling the training of deeper networks. In each double convolutional block, the output of the second convolution is added to the input, creating a residual connection.

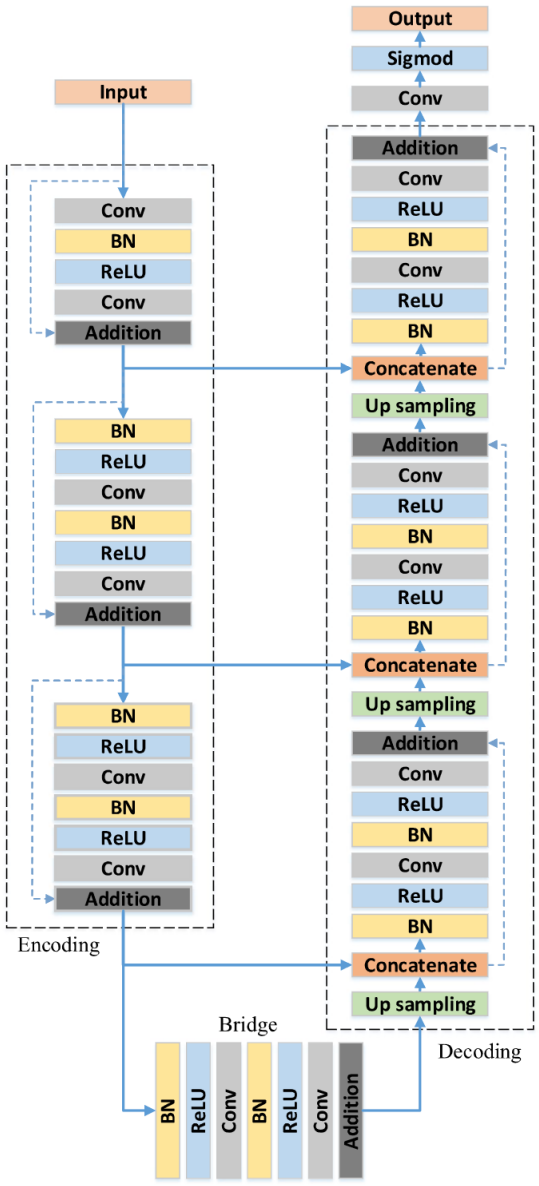

Fig. 6. Architecture of Deep Residual U-Network to implement

The core building block of the Deep Residual U-Net is the double convolutional block, similar to the standard U-Net. Each block consists of two 3x3 convolutions followed by batch normalization and ReLU activation functions. The integration of residual connections within these blocks enhances the flow of information and facilitates the learning of intricate features.

Similar to U-Net, the final layer of the Deep Residual U-Net is a 1x1 convolutional layer responsible for generating the

segmentation map. It produces pixel-wise predictions, where each pixel represents the probability of belonging to a specific class.

The Deep Residual U-Net architecture is well-suited for training on large datasets and can be adapted for various segmentation tasks by adjusting the number of input channels and output channels according to the requirements of the specific application. The combination of U-Net's spatial feature learning with ResNet's residual connections makes this architecture powerful for semantic segmentation in computer vision.

During training, the Mean Squared Error(MSE) was used as the loss function. This loss function is used to measure the difference between the predicted values and the ground truth, guiding the optimization process. The MSE here is defined as:

$$\mathrm{L(w)} = \frac{1}{N}\sum_{i=1}^{N} ||Net(I_i; W) - s_i||^2 \tag{1}$$

Where w is the parameters of the neural network, N is the total number of training samples, Net(I_i;w) represents the output of the neural network with an input I_i and the parameter s_i is the actual ground truth. Early stopping is implemented to prevent overfitting, and the training process is completed. This ensures that if the validation performance fails to improve for a predefined number of consecutive epochs (patience), training is halted, preventing the model from learning noise in the training data and becoming overly specialized to it. Early stopping aims to find the optimal balance between training a model long enough to learn useful patterns and stopping before it starts memorizing noise or idiosyncrasies present in the training set. This technique helps ensure that the final model exhibits robust performance on unseen data, making it a valuable tool for training deep learning models.

D. Mirror Net

Another network derived from the U-net is the Mirror Net. This network was designed specifically for the challenge of the camouflaged images and it will be used as comparison for our results with the deep residual network. The Mirror net introduced a new idea where the input image is mirrored in order to disrupt the camouflage of the object in the image. The flipping is able to is able to reveal subtle features like shape, texture and shades. This aditional perspective is then compared to the likelihood of having an objet also obtained by the original input image, and therefore, with an extra new set of information the results looked promising when presented in [16] However, due to this mirroring, the processing of the images are now doubled, and for that reason the computational cost of the mirror net is high.

E. Evaluation Metrics

In order to evaluate the performance of the image segmentation, precision and recall were used. In the original paper, relaxed precision and recall scores were used. This won't be implemented in our case due to the difference in data set and the mask does not contain any narrow structures. Precision is also called the positive predictive value while recall is also called as the sensitivity. These are calculated by finding the true and false, positives and negatives. Precision determines the portion of the predictions that were correct so the following equation is:

$$\mathrm{Precision} = \frac{\mathrm{TruePositive}}{\mathrm{TruePositive} \ + \ \mathrm{FlasePositive}} \tag{2}$$

A good indicator that a model is accurate is it has a precision of 1 which means that all of the predictions were correct. However, this metric is not enough because it might only be focusing on some features but are not enough to represent the whole dataset it was trained on. Which is where the concept of recall is needed which shows the proportion of actual positives that were identified correctly and will be as seen in the equation below:

$$\mathrm{Recall} = \frac{\mathrm{TruePositive}}{\mathrm{TruePositive} \ + \ \mathrm{FlaseNegative}} \tag{3}$$

It is to be noted that a model without any false negatives will have a recall of 1. At the same time, only having recall is not enough so the combination of the two provides a good evaluation for the performance of our neural network to predict binary masks.

Additionally, another metric can be used which was used in the CAMO dataset paper which is the F-measure [15]. This additional metric, is a balanced measurement between the precision and recall as seen in the equation below.

$$F_\beta = \frac{(1+\beta^2)\ \mathrm{Precision} \ + \ \mathrm{Recall}}{\beta^2\ \mathrm{Precision} \ + \ \mathit{Recall}} \tag{4}$$

In this paper, a value of β^2 of 0.3 will be used which will put an emphasis on the precision as discussed by the work of Achanta [17]. There are other evaluation metrics that were in the CAMO paper such as the Intersection Over Union (IOU) by Long [18] and Mean Absolute Error (MAE).

III. Results

A. Model

In the original paper, the proposed Deep Residual U-Net was applied for road extraction from aerial images. This produced a binary mask of the roads and it was shown that it outperforms other methods. The network was created following the figure of the architecture which created the code as shown in the appendices. The overall architecture composed of 3 encoding layers, a bridge layer and 3 decoding layers.

The camouflage dataset is composed of a total of 1250 images and labels. The original dataset has colored images with 3 channels, RGB, while the labels are binary masks which reveal the location of the camouflaged object. The images also have a large variation in their image size so some preprocessing was needed. Both the images and the masks were resized to have a size of 224x224 which is the same as what the original paper used. This greatly helps keep track on the shapes of each layer to be compared with the table from the original paper and also reduce the computational time, as

most of the images have dimensions greater than 1000 pixels. The neural network will be trained using the 1000 images with the following hyperparameters: epochs = 30, learning rate = 0.001, training-to-validation ratio = 8:2, and a batch size = 50. loss-function: Mean-Squared Error.

In terms of the model, it self the following sizes, with the convolution parameters are as shown below. It can be noted that this table does not exactly match with what the paper shown nor is it the same as the github repository that it came with. This is due to the fact that the figure seems to be incomplete while the github code didn't even follow explicitly what was proposed in the paper while also using tensorflow which is a library that is slowly depreciating. This means that in order to create a working model where the sizes match properly, some modifications were made with extra convolutional blocks in order to match the right number of trainable parameters.

TABLE I
RESIDUAL U-NET MODEL SUMMARY

Parameter	**Value**
Model Architecture	Residual U-Net
Input Channels	3
Output Channels	1
Optimizer	Adam
Learning Rate	1×10^{-4}
Loss Function	BCEWithLogitsLoss (Binary Cross-Entropy)
Batch Size	2
Number of Epochs	100
Early Stopping Patience	10
Image Size	256×256
Convolutional Layers	Input Conv2d(3, 64, kernel_size=3, padding=1) Output Conv2d(64, 1, kernel_size=1)
Residual Layers	Residual Block: Conv2d(64, 64, kernel_size=3, padding=1) Residual Block: Conv2d(64, 64, kernel_size=3, padding=1)

B. Output

The trained network was tested on 250 sets of images and labels. In the figure below, sample images of the output by using the neural network is as shown. It can be seen that the produced mask is fairly close

There were two different output types that we tried to produced. One produces a non-binary mask while the other produces a binary mask.

From all the test images and their corresponding outputs, a value of precision and recall is noted. All of the values are plotted in a scatter plot to produce the figure below.

The Residual U-Net model is instantiated with specified input and output channels (ie) (3,1). The training process involves using the Adam optimizer with a learning rate of 1×10^{-4} and binary cross-entropy loss for binary segmentation. The dataset is split into training, validation, and test sets, each managed by data loaders. The training loop, running for a designated number of epochs (50,100), incorporates early stopping(only for 100 epochs). Early stopping is configured with a patience of 10 epochs, where training halts if the validation loss does not improve for 10 consecutive epochs. The visualizations for five examples from the test set, including input images, ground truth masks, predicted masks, and overlays highlighting detected camouflage regions, are displayed.

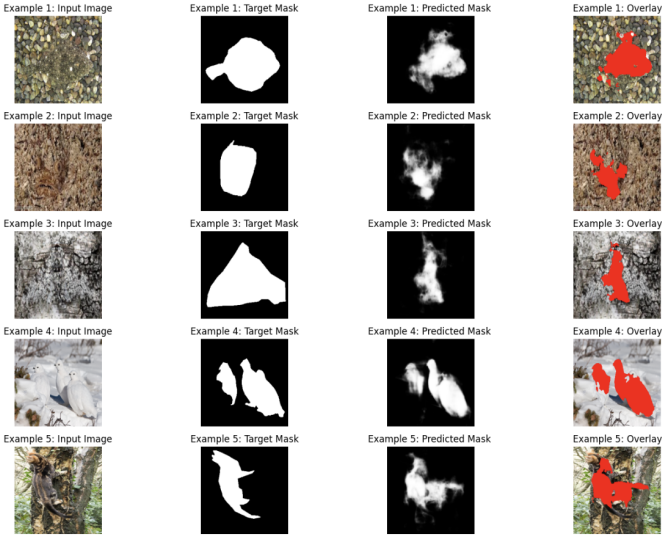

Fig. 7. Results with Non-Binary Mask Output using 50 epochs of Residual U-Net.

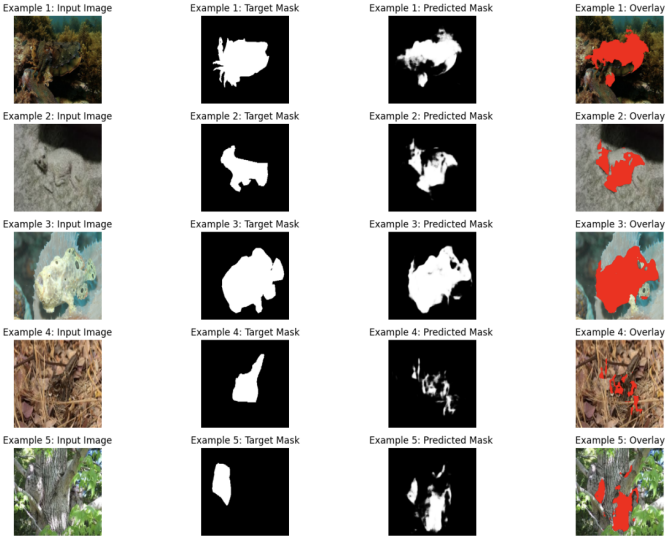

Fig. 8. Results with Non-Binary Mask Output using 100 epochs of Residual U-Net.

IV. DISCUSSION

With the results that we have, we compare the effectiveness of using the Deep Res U-Net with the neural network that was used in the first paper analyzing the CAMO dataset [15]. In that dataset, the Anabranch Network (ANet) was used which composed of 8 layers namely in the correct order: fully connected, reLU, dropout 50%, fully connected, reLU, dropout 50%, fully connected and a soft-max layer. The Anabranch

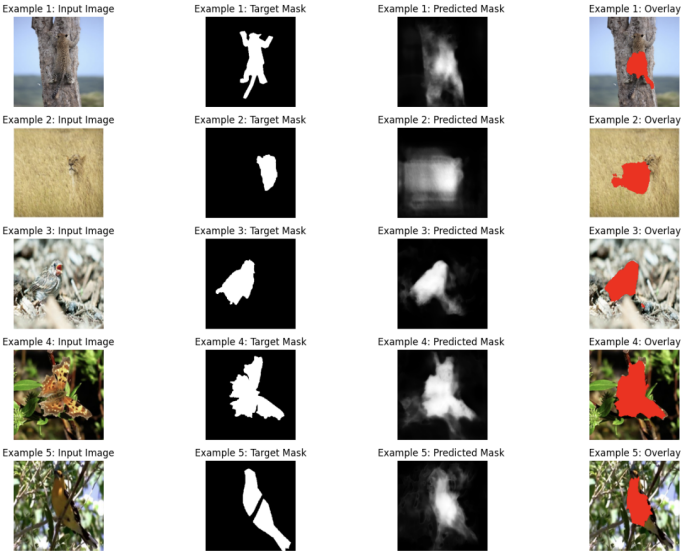

Fig. 9. Results with Non-Binary Mask Output using 100 epochs of Residual U-Net with early-stopping.

Network was compared with various different salient image segmentation networks such as the SVM-BoW, AlexNet, and VGG-16. But for intensive purposes will not be looked at. In terms of the F_β score, the ANet was able to achieve up to 90% and the Deep Residual U-Net had a value that was around that value. It is to be noted though that the ANet was tested also for non-camouflaged objects which would make it unfair to compare with ours. In the training for only the CAMO dataset, the ANet was only able to have an F_β score of 65% on average which is lower than the F_β score using the Deep Res U-Net. The ANet was trained using 10 epochs while the Deep Res U-Net was trained using 50 epochs which would be slightly unfair but that is because the ANet reached convergence much earlier.

In terms of the output of our network which was a single channel image, it is to be noted that the original output has a datatype of a float value continuous between 0 and 1. This means that to create a binary mask a thresholding must be done at a specified value. In the code that was developed in the original Deep Res U-Net paper that was found on github, a value of 0.4 was used. The reason for choosing such a value was not explained and seemed illogical because it would be more intuitive to choose 0.5 instead. Choosing this thresholding value will affect what the binary mask looks like especially near the borders of the original target mask. Because the binary mask is affected, then it's performance and the evaluation metrics will change as well. However, during some of our experiments with changing this thresholding value we were surprised to find that it barely affected the performance of the network. This might be due to large gradient in the borders of the output image, implying that small deviations of thresholding value will be negligible unless it has a difference of around 0.3.

Last thing to be noted was the proposed architecture in general. Similar to the U-Net which provided the general framework but not the specifics of the number of layers, the Deep Res U-Net can be applied to an arbitrary amount of layers as well in the encoder and decoder. The figure above which was the proposed architecture shows 3 layers each for the encoder and decoder but in the github they provided, they actually used 4 layers each. This of course will affect the performance of the network but it is interesting to take note of.

A big factor to be noted with neural networks is the computational time. This is because the complexity of the necessary network depends on the complexity of the problem. Which means that more parameters that are needed to be learned and at the same time, more epochs and training to be done in order to properly represent the entire training dataset. In our trials, various major hyperparameters were chosen which were tweaked. The list below shows the different hyperparameters chosen the following values it took.

- Epoch: 10, 20, 30, 50, 100
- Training Data: 250, 500, 1000, 1250
- Batch Size: 4, 32, 40, 50,
- Learning Rate: 0.01, 0.001, 0.0001

A first analysis that can be performed in the results shown from Figure 7-9 is regarding the effect of the epochs. It can be observed in Figure 7 that in general the results are far from satisfactory. As the network does not have enough time to learn the underlying patterns, the features that hard to capture during camouflaged images are not identified. However, it is possible to see that even with a small number of epochs, for images that the object camouflage is more soft, such as the example 4 in Figure 7, the animals can be pretty well captured, showing the good potential of the network. As we increase the number of epochs, it is possible to see the the masks being output by the deep residual network start getting better and better. In Figure 9, examples three and four completely capture the correct mask of the object, and for the fifth example it is possible to see that the mask being performed do capture the entire shape, however a small part of it still do not have incorporated a high probability on the feature map.

However, as good as it seems to always increase the number of times the entire dataset is passed forward and backwards through the network, there are some disadvantages that limits this increase. One of the most obvious reasons is that the computational time can increase significantly as we increase the epochs, specially if an order of ten is needed for capturing good improvements as seen from Figure 7 to Figure 9. Another common problem as the number of epochs is increased is the overfitting of the data, where the model not only learns the important underlying patterns in the camouflaged images, but also starts incorporating the noise and other undesired characteristics of the images in the training. As the dataset passes many times thorugh the network, the network mmight memorize these undesired patterns found in the images.

A. Limitations

It could be seen that the deep residual network was able to somewhat capture the features and shapes of the hidden objects, however its performance seem to be limited to high camouflaged objects as observed in some of the examples. One of the reasons could be that, even though the residual connections help with reducing the vanishing gradient problem, because the high detailed features are ingrained in the image, the information might be lost during the expanding path.

B. Future Work

It was seen in this work that the Mirror Net was superior in identifying the camouflaged images when compared to the deep residual network, and the reason for that is believed to be due to the sophisticated flipping technique discussed previously in the Mirror Net. An interesting future work would be to introduce into the training datasets not only the normal images with its masks as done in this work, but also introduce the flipped images and its corresponding masks in order to verify how much improvement the deep residual network would get. Even though this would increase the computational time of the network, it would be interesting to perform a comparison with the Mirror Net and see if the extra computational time in the sophisticated implementation of the Mirror Net is produce much better results that are worth spending the extra time.

V. Conclusion

In conclusion, the Deep Res U-Net proved to be feasible to be used for image segmentation aside from just road detection. Despite the non-trivial task of figuring out the location of a camouflaged object, it was able to give out substantial results. The usage of a deep residual network was shown to be able to get details of the image, but the increase in depth of the network architecture led to an increase in the computational time. Even though the deep residual network had some success in identifying the objects shape, networks such as Mirror net that are designed specifically for camouflaged datasets outperform the deep residual network results. This paper focused not just on the implementation of the network but also looking at the effects of adjusting some simple hyperparameters such as epoch, batch size, learning rate and optimizer and also some complex hyperparameters such as adding and removing layers that affected both the computational time and the accuracy. There is much more left for future work as the field is ever growing with various amounts of different architectures and datasets popping up. Hopefully in the future, we are able to work on more complex and interesting ones like this camouflaged dataset.

Acknowledgment

We would like to thank our beloved teacher, Professor Mehmet Akcakaya for instilling in us the wonders of image procession and the magic that makes it all work together. We admit that we haven't been the best listeners in class and sometimes dozes off but this is not due to the lack of enthusiasm of the professor nor is it the dullness of the topic. This was due to our own personal physical struggles which is most of the time our fault. So we would like to thank him also for his patience and understanding. We would also like to thank our unbeatable jolly teaching assistant, Merve for always providing her assistance whenever we have problems with our homeworks, projects or even just basic computer stuff. Without her, the class would have shattered our hearts due to the difficulty of the work that needs to be submitted. Lastly, we would like to thank God for giving us strength, our family for supporting us, and our friends for keeping us sane through tough times.

References

[1] Z. Zhang, Q. Liu and Y. Wang, "Road Extraction by Deep Residual U-Net," in IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 5, pp. 749-753, May 2018, doi: 10.1109/LGRS.2018.2802944.

[2] Saito, S., Yamashita, T., Aoki, Y. (2016). Multiple object extraction from aerial imagery with convolutional neural networks. Electronic Imaging, 2016(10), 1-9.

[3] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[4] He, K., Zhang, X., Ren, S., Sun, J. (2016). Identity mappings in deep residual networks. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14 (pp. 630-645). Springer International Publishing.

[5] Long, J., Shelhamer, E., Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[6] Ronneberger, O., Fischer, P., Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 (pp. 234-241). Springer International Publishing.

[7] Chollet, F. (2021). Deep learning with Python. Simon and Schuster.

[8] Le, T. N., Nguyen, T. V., Nie, Z., Tran, M. T., Sugimoto, A. (2019). Anabranch network for camouflaged object segmentation. Computer vision and image understanding, 184, 45-56.

[9] Li, F., Liu, Z., Chen, H., Jiang, M., Zhang, X., Wu, Z. (2019). Automatic detection of diabetic retinopathy in retinal fundus photographs based on deep learning algorithm. Translational vision science technology, 8(6), 4-4.

[10] Zeineldin, R. A., Karar, M. E., Coburger, J., Wirtz, C. R., Burgert, O. (2020). DeepSeg: deep neural network framework for automatic brain tumor segmentation using magnetic resonance FLAIR images. International journal of computer assisted radiology and surgery, 15, 909-920.

[11] Liu, Z., Cao, Y., Wang, Y., Wang, W. (2019). Computer vision-based concrete crack detection using U-net fully convolutional networks. Automation in Construction, 104, 129-139.

[12] Brand, A. K., Manandhar, A. (2021). Semantic segmentation of burned areas in satellite images using a U-net-based convolutional neural network. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, 47-53.

[13] Mnih, V., Hinton, G. E. (2010). Learning to detect roads in high-resolution aerial images. In Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI 11 (pp. 210-223). Springer Berlin Heidelberg.

[14] Saito, S., Yamashita, T., Aoki, Y. (2016). Multiple object extraction from aerial imagery with convolutional neural networks. Electronic Imaging, 2016(10), 1-9.

[15] Le, T. N., Nguyen, T. V., Nie, Z., Tran, M. T., Sugimoto, A. (2019). Anabranch network for camouflaged object segmentation. Computer vision and image understanding, 184, 45-56.

[16] Yan, J., Le, T. N., Nguyen, K. D., Tran, M. T., Do, T. T., Nguyen, T. V. (2021). Mirrornet: Bio-inspired camouflaged object segmentation. IEEE Access, 9, 43290-43300.

[17] Achanta, R., Hemami, S., Estrada, F., Susstrunk, S. (2009, June). Frequency-tuned salient region detection. In 2009 IEEE conference on computer vision and pattern recognition (pp. 1597-1604). IEEE.
[18] Long, J., Shelhamer, E., Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).