U-Net Convolutional Neural Network For Camouflaged Object Detection

Karl B. King
Department of Computer Science
University of South Carolina Upstate
Spartanburg, South Carolina, USA
kbking@email.uscupstate.edu

ABSTRACT

In this paper we will be exploring the effectiveness of U-Net Convolutional Neural Networks. The context for this exploration is the detection of people mostly in, or attempting to simulate military camouflage. Questions we seek to solve are: what complexity of neural networks are more effective, by what margin are they more effective, if not superior in all ways, which ways are the neural networks complexities superior to one another, and why are each of these neural network complexities more effective than the other.

KEYWORDS

AI, Artificial Intelligence, ML, Machine Learning, NN, Neural Network, CNN, Convolutional, Neural Network, U-Net, Image Segmentation, Camouflage, Environmental Mimicry, Color Replication, Blending.

1. INTRODUCTION

I believe that computer vision is going to be a cornerstone of the next wave of major technological advancement for the human race. This is evident in the role it must play in the next stage of automation, namely in self-driving vehicles, complex problems of diagnosis, and increasing the robustness of security systems. Adjacent to the problem of security systems is the problem of target acquisition for our police and soldiers, a common trend correlating to the introduction of firearms into the arsenals of nations across the world is the reduction of personal armor systems among infantry troops. This reduction in personal armor was eventually supplemented by the introduction of camouflage. The ultimate goal of this project is to create a software that nullifies the effectiveness of camouflage in a similar way.

2. LITERATURE REVIEW

My approach to this project to defeat camouflage was to first look into how our military evaluated camouflage in the past, for this I looked at a declassified report from the year 2000 “Assessing Camouflage Methods Using Textural Features”. Within this work the authors attempt to define metrics and features to look for when determining the strengths and weaknesses of a camouflage method; they landed on features of sharp edges and reflective surfaces being primary weaknesses of camouflage designs[1].

I knew I wanted to tackle this as an image segmentation problem and began my research investigating the ins and outs of segmentation and popular methods for achieving my goal. This search led to two papers, the first was “Image Segmentation Techniques Overview” by Yuheng Song and Hao Yan, the second was “U-Net: Convolutional Networks for Biomedical Image Segmentation”.

The first of these I found very valuable as a quick examination of multiple techniques including threshold segmentation, regional growth segmentation, edge detection through Sobel’s and the Laplacian Operator, clustering and finally weakly-supervised learning in CNN their strengths and weaknesses[2]. The information here led me to the structure I ended up using, the U-Net, and the paper explaining it “U-Net: Convolutional Networks for Biomedical Image Segmentation". The U-Net is the foundation of my project and its structure will be more thoroughly detailed in a later section. However, the main draw of this structure comes from its speed and how well it works with a smaller dataset[3], as the totality of my chosen dataset was just over 1000 images.

After searching through the history of each part of the problem I am attempting to solve, I decided to look to see if and how other people have attempted such a daunting task. Through this, I discovered the work “Camouflaged Object

Detection” and came up with a plan to use their more robust data set, the COD10k[4], for training and the much smaller dataset I was using previously for validation and testing to hopefully harness the precision granted by the larger training set of more complex images to establish a better baseline for finding humans attempting to hide.

Further research led to an expansion of the goals of the above mentioned project by comparing and contrasting data from salient, or obvious, objects to enhance the predictions of the camouflaged objects[5]. While I was ultimately unable to harness specific structures from these two projects due to time constraints, it serves as a good basis to learn from as I continue to develop this project on my own time.

3. METHODOLOGY

This experiment began as a comparison between the structures of the standard convolutional neural network and the standard recurrent neural network for camouflaged object detection. I quickly found that this goal was unattainable at my skill level in the given time frame as I failed to realize the depth of complexity that this problem presented. Where normal image segmentation methods rely on differences in the target segment and its surroundings, a solution to this problem instead must be able to converge on a segment that is trying to blend into its background, robbing the method of that vital information. After I realized that my original goal was unattainable, I shifted my focus towards more complex and unconventional models and eventually landed on the U-Net architecture.

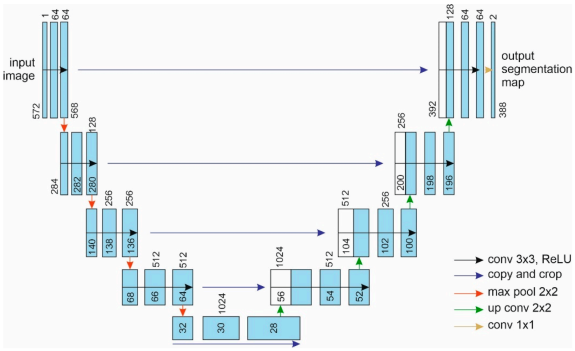

Figure 1. The U-NET Architecture

The U-NET architecture works by encoding through blocks of subsequent convolutional layers and then downsampling the image while increasing the number of observed filters between each block, down to a bridge or base layer with the lowest resolution and the highest numbers of filters. Following the base layer, the decode blocks are made of layers that upsample the image again and decrease the number of filters while convoluting and concatenating the result of the previous block of the same level. This concatenation is called skip tracing and is the heart of the architecture. In more practical terms, the process of training the U-NET is as follows: the image is input alongside a segmentation mask, also referred to as a ground truth(GT). As the image is downsampled, the ground truth’s important area loses detail and is widened to allow it to learn a simpler area before the decode layers are applied to teach more constricted and precise areas. The concatenations facilitated by skiptracing allow the model to compare its new assumptions to its original ones. The pros of this model are its speed and ability to learn off of relatively small datasets, both being major concerns for my goals in the context of supplementary optics for militaries - as every second counts when lives could be on the line. The cons are how particular it is when it comes to input shapes and how, because of the high number of parameters, it can be prone to overfitting. The latter was a large problem for me as I found there is only a very small window where my implementation of this architecture became viable.

Due to the complexity of the problem and the time it took to understand this architecture, I made the decision to shift from comparing this architecture to an alternative one, to instead getting as familiar as possible with this single architecture and determine the optimum way to interact with the model structure. This Includes but is not limited to different depths of blocks, altering the resolution of input images and more that will be discussed in the experiment setup section.

4. IMPLEMENTATION

The language used for this project is Python. The models will be developed in the web based development environment Google Colaboratory. I chose this environment because it boasts a multitude of important libraries for the development of machine learning models (Tensorflow, Keras, and NumPy), computer vision and image processing (CV2), and displaying results (Matplotlib). These libraries being integrated with the environment minimizes errors that may arise either from user error in installation or low portability that is intrinsic with any code dependent on specific versions of libraries, interpreters, et cetera. Another draw to Google Colaboratory is that it offers cloud based hardware, namely large stores of RAM and Tensor Processor Units - both of which are integral to managing both the training and execution time of models with large numbers of parameters.

5. EXPERIMENT SETUP

5.1 DATASET

The primary dataset that I used was the Military Camouflage Soldiers Dataset (MCS1K). This set consists of 1078 image-GT pairs pre-split into 748 training pairs and 330 testing/validation pairs each of varying dimensions. I used this over the more robust COD10K dataset mentioned above, as I received better and more consistent results with the smaller dataset. With this, however, I learned some limitations of the training process and with this dataset: it struggles to serialize sets of images of 512 X 512 resolution or greater in numbers larger than 1000. This was a deciding factor when choosing the MCS1K dataset over the COD10K. For future development of the project, I may work to better accommodate this larger, more robust dataset but with the timeframe I am working with, it is not feasible to do so.

5.2 EVALUATION METRICS

The primary metric for evaluating this project is the intersection over union score also known as the IoU score, this metric is calculated by dividing the area of intersection between the predicted segmentation mask and the ground truth by the total area of the union of the predicted mask and the ground truth.

Time is also an important metric, as stated above in scenarios where lives may be on the line and having a fast acting target acquisition assistant could be the deciding factor. Both training and validation loss were used to measure and determine the optimum training time.

5.3 PROCEDURES

The experiment starts with preprocessing the data, which includes resizing both the image and ground truth to a 512x 512x3 or 256x256x 3 shape. After resizing the ground truth, we normalize it with a binary threshold - setting all non 0 values to 1. This ensures proper calculations for our loss function and speeds up the load times for the GTs. These are then loaded into NumPy arrays for use in the model. Afterwards, we perform a sanity check that outputs the largest element present in our GT to verify that it is indeed 1, then outputting an image-GT pair to verify that the data has been pulled into the arrays in the correct order. Once the sanity check is performed, the images are passed into the model for training where we measure time and accuracy. After the training, we pass in a few lone images to test the effectiveness of the model and gather the IoU score. All results are output in figures using Matplotlib.

6. RESULTS ANALYSIS

The first set of results are from training model 1 a U-Net with 3 encode layers, a base layer followed by 3 decode layers - for a total of 7,697,475 parameters. The model was trained on 640 image-GT pairs with a validation set of 330 pairs.

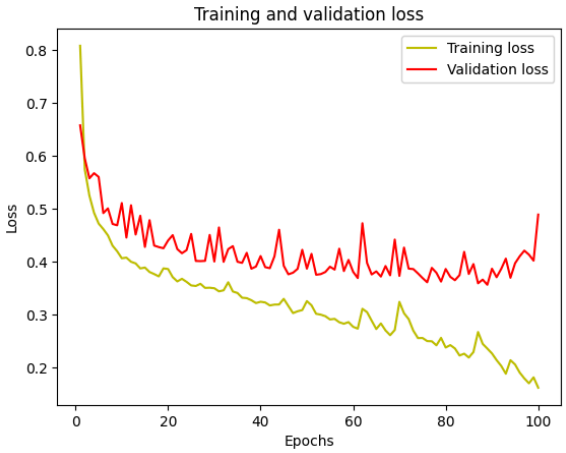

Figure 2. Model 1 Training and Validation loss

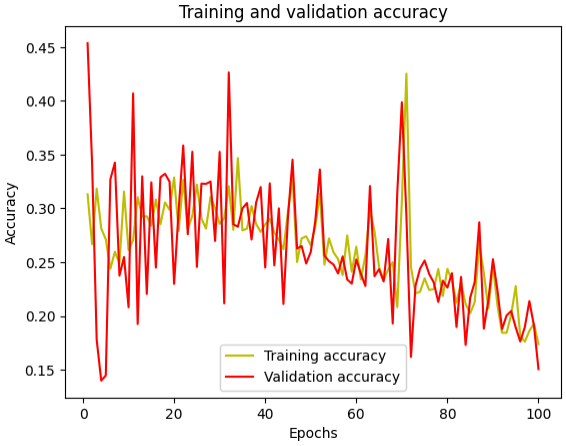

Figure 3. Model 1 Training and Validation Inverse Accuracy

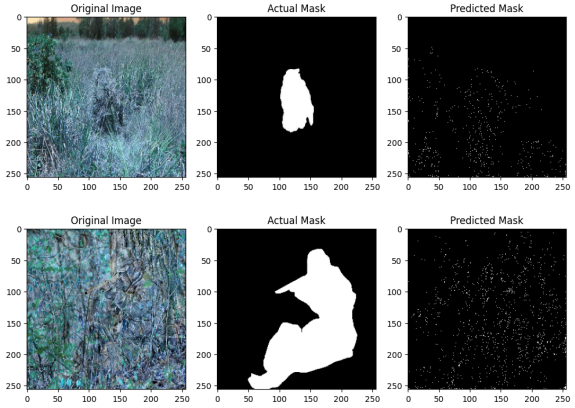

Figure 4. Sample: Upper: 9% | Lower: 4%

The following figures are from model 2 and tested identically to model 1. This model follows the same U-Net Architecture as model 1, however, model 2 adds a single encode block and its matching decode block. This 4x1x4 model has a total of 31,031,875 parameters. Through various attempts I determined the optimum number of training epochs to be 100 and the batch size to be 64

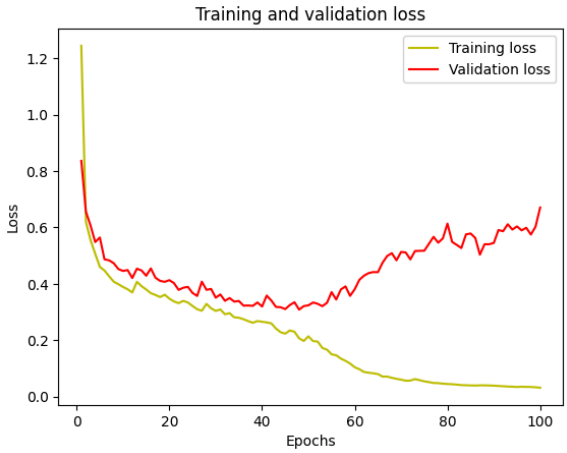

Figure 5. Model 2 Training and Validation loss

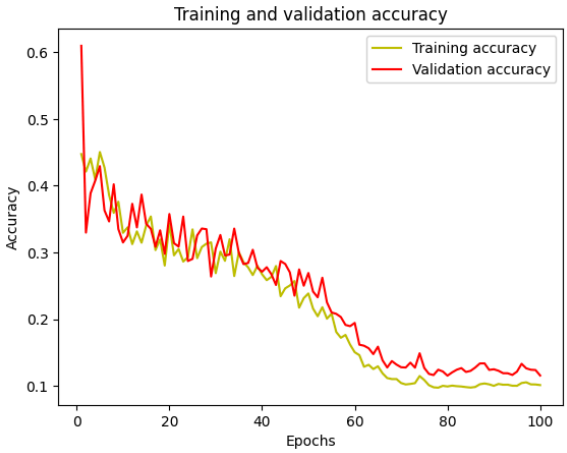

Figure 6. Model 2 Training and Validation Inverse Accuracy

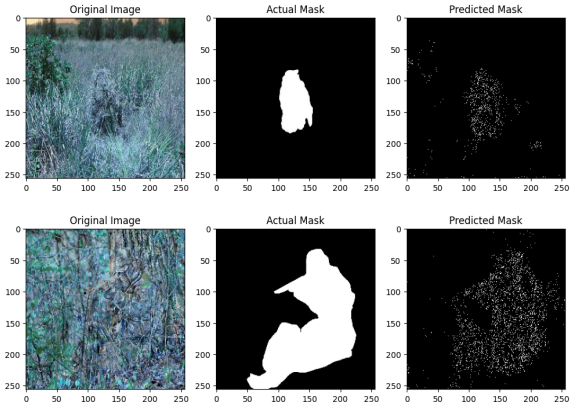

Figure 7. Sample: Upper IoU: 14% | Lower IoU: 12%

7. CONCLUSION

One can clearly see that model 2 is the superior model. While far from perfect, in many cases it serves as a good proof of concept for the use of a U-Net as a means for camouflaged object detection. Going forward, I will attempt to determine the extent to which additional levels will aid the predictions of the model while still maintaining a feasible execution time. In addition to these layer changes, I intend to make some additions inspired by the works of Fan Den-Ping and Aixuan Li. [4&5]

My hypothesis for the wide gap in effectiveness between model 1 and model 2 is two fold. first: that model 1's shallow depth, compared to model 2, prevented it from reducing the image's resolution to a shape simple enough to learn from. Second: that the model's simplicity further thwarted its capabilities. My justification for this is that model 2 was a more complex model that reduced the image resolution to an even simpler shape. This simultaneously made the training easier to learn from and the added parameters of the model extended its learning limit.

Of my metrics, one eludes me. The 'accuracy' in testing seems to measure backwards or to be more precise the lower the model's displayed accuracy, the closer the model is to actually generating an acceptable solution. At a very high accuracy, the model always outputs either a totally black or totally white image. With middling accuracy, it nearly always output a totally black image. At numbers of training epochs 95 to 105, the model would output the speckled images seen in the figure 7 - varying in concentrations, sometimes thinner and sometimes thicker. The images provided are representative of the average of these ranges. 100 epochs was the most consistent but still exhibited variance on retraining.

WORKS CITED

[1] Nyburg, Sten; et al, *Assessing Camouflage Methods Using Textural Features,* Report date 2000-03-01 ; Accession Number: ADP010540

[2] Y. Song and H. Yan, "Image Segmentation Techniques Overview," *2017 Asia Modelling Symposium (AMS)*, Kota Kinabalu, Malaysia, 2017, pp. 103-107, doi: 10.1109/AMS.2017.24.

[3] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

[4] Fan, Den-Ping, et al, *Camouflaged Object Detection,* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020

[5] Li, Aixuan, et al, *Uncertainty-Aware Joint Salient Object and Camouflaged Object Detection,* Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021