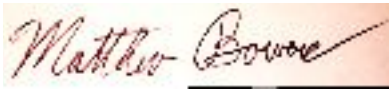
# CS/RBE 549 Computer Vision, Fall 2019

# Project Report

# **Team L. Blue Bird**

| **Member** | **Signature** | **Contribution (%)** |
|---|---|---|
| Calum Briggs |  | $33.\overline{3}$ |
| Daniel McDonough |  | $33.\overline{3}$ |
| Matthew Bowers |  | $33.\overline{3}$ |

| | | |
|---|---|---|
| Grading: | Approach | _____/20 |
| | Justification | _____/10 |
| | Analysis | _____/15 |
| | Testing & Examples | _____/15 |
| | Documentation | _____/10 |
| | Difficulty | _____/10 |
| | Presentation | _____/20 |
| | Total | _____/100 |

## Abstract

In this project, we have explored and implemented two different approaches to recognize our given object; a blue porcelain bird. Using semantic segmentation techniques such as UNet and knowledge of the object, we attempted to reliably detect the bird in any given image.


*Keywords - UNet, Region Segmentation, Computer Vision*

# Table of Contents

**Abstract** 1

**Table of Contents** 2

**List of Figures** 3

**Introduction** 4

- **Background** 6

**Methodology** 9

- **Dataset Generation** 9
- **UNet** 11
- **Manual Contour Separation** 13
- **Assessment Protocols** 16

**Results** 19

- **UNet** 19
- **Manual Contour Separation** 20

**Conclusions** 22

**References** 23

**Appendices** 26

# List of Figures

1.1 Image Recognition vs Object Detection (Fritz A.I. 2019). 5

1.2 Blue Bird. 6

3.1.1 Mask Generation. 9

3.1.2 Background Examples 10

3.1.3 CamSeq01 + Bird Dataset 11

3.2.1 UNet Architecture (O. Ronneberger 2017) 12

3.3.1 Bird in CMYK 14

3.3.2 Bird in CMYK + Thresholding 15

3.3.3 Bounding Box Generation 16

4.1.1 mAP Testing 100 Iterations 19

4.1.2 UNet Predicted Labels 20

4.2.1 Flawed Identification 21

4.2.2 Blur and Partial Occlusion 21

# 1. Introduction

Computer vision is the ability of a computer to understand the content of videos and pictures. While this is a relatively simple task for humans, it has proved to be an immense challenge to develop hardware and software that can master this task. Facial recognition, object detection, and navigation are just a few of the many important applications that drive study within the field.

Of interest to us is the ability to detect a particular object under a variety of conditions. This task provides us with both identification and location of an object or objects and can be utilized in both static images as well as in videos [1]. This is a more thorough and often more useful result than just image recognition by itself. The difference can be seen in Figure 1.1 below.

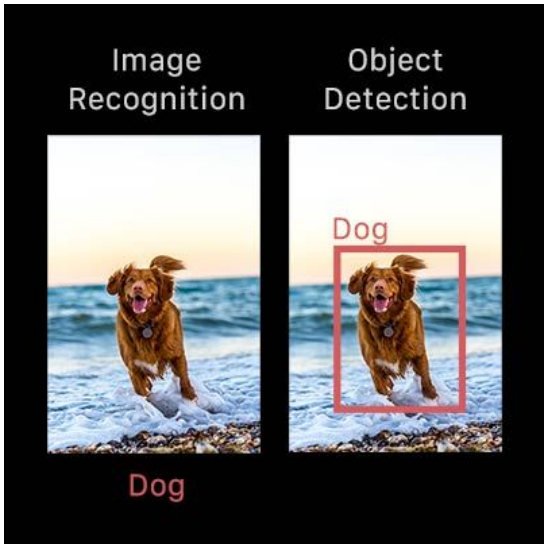



**Figure 1.1** Image Recognition vs Object Detection (Fritz A.I. 2019).

Object detection involves identifying each instance of an object in an image and drawing a bounding box around it. This provides more info than image recognition which when used to analyze an image of multiple dogs would only label the image dog, or at best, dogs, if it had been trained to classify those as separate labels. The capabilities object detection provides has a wide range of uses in situations where identification, counting, and tracking are needed. Applications such as crowd counting, self-driving cars, and video surveillance are important for accountability, safety, and security.

While as previously stated humans can easily perform these same tasks, the amount of data that is out their to be analyzed is more than what humans have time for. Additionally as the field moves forward computer will surpass, and in some cases already have, our ability to do this. Which in turn will make driving, one of the largest sources of fatalities in the world, much safer. Additionally, it will improve security as computers can watch/analyze every image constantly without distraction.

**Figure 1.2.** Blue Bird.

Our task is to detect the blue porcelain bird shown in Figure 1.2. This bird has a habit of wandering off and being in places where he doesn't belong. By utilizing our knowledge of the bird: its regions, shapes, colors, edges, and features, we can identify and locate him at all times. In the pursuit of this task we will not always have ideal conditions and would need to address the challenge of object identification in scenarios where the object is in different locations, orientations, relative sizes, situations where the image is blurred, or situations where the

bird is partially hidden, and under different lighting conditions, or even potentially if this was not a rigid object situations where regions warp and move relative to each other. In this project we have focused on the first six situations.

## 1.1. Background

Many techniques exist to tackle the computer vision goal of object detection. In general, these methods either rely on machine learning or deep learning. In machine learning approaches, machine learning algorithms define the relevant features, and then classification of the object is performed afterwards. In contrast, deep learning approaches detect the object without additional steps and does not define the features.

Machine learning approaches use features like Haar, which are used in the Viola–Jones object detection framework [2] and histogram of oriented gradients features (HOG) [3], as well as algorithms such as the scale-invariant feature transform (SIFT) [4]. While a common deep learning approach is region proposals such as regions with convolutional neural network features (R-CNN) [5-7] and its successors, which make some improvements to the speed of R-CNN.

Haar-like features are simply the difference in the sum of pixel intensities between adjacent rectangular regions. These features have an advantage over many other features because the speed of its calculations for the Integral Image image representation. In the Viola–Jones object detection framework critical features are then fed into an AdaBoost based learning algorithm. Finally increasingly complex classifiers are merged in a cascade. The advantage this provides is that the background regions

are given less focus than potential object regions, reducing the area of interest and speeding up the computations. Some major disadvantages are that it is not great with changes in lighting conditions and does not handle rotations.

Dalal describes HOG as “The proposed descriptors are reminiscent of edge orientation histograms, SIFT descriptors and shape contexts, but they are computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalizations for improved performance.” [3] HOG’s focus on edge orientation makes it robust against changes in color, lighting, and small variations in shape. This makes it good for use as a human detector. However, obviously since it is based on edge orientation it is sensitive to image rotations.

SIFT is method of feature generation which focuses on describing local features in a scale invariant manner. SIFT is great at identifying objects in cluttered images as well as when the object is partially hidden. Additionally it is invariant to changes in lighting, scale, and orientation. The main disadvantage of SIFT is that it is computationally heavy.

R-CNN is a selective search algorithm that reduces a potentially large number of regions by selectively extracting only 2000 regions. These are referred to as region proposals. Then these are combined with CNNs to achieve a dense feature representation. While less regions than previous options 2000 regions is still a lot. Additionally since it was trained on feature vectors created by Alexnet, CNN and SVM must be trained sequentially, which is computationally slow.

Our object has well defined regions and contours with large contrast in color and brightness. In addition the porcelain bird is rigid and therefore the geometric relations between features behave in a consistent and predictable manner, allowing us to define relationships between shapes and sizes between different regions of the object. We also want to be able to detect our object in a large amount of 3D orientations and even when only partially visible. Less of a concern is lighting invariance as the object features themselves are not as sensitive to changes in lighting as other types of objects. These characteristics lead us to consider region based approaches that focus on those relatively easy to detect attributes. As UNet provides semantic segmentation, we determined it would allow for easy detection of the bird's major regions of interest (the feet, head, and hair of the bird) to be detected and represented through a feature vector. Additionally, we manually method for determining feature vectors by using our knowledge of the bird's shape and colors. This was chosen because each colored region was clearly defines and monochromatic within each region.

# 2. Methodology

## 2.1. Dataset Generation

In order to test any method of detection, we produced a dataset such that we can formulate a ground truth to test against. Using a Samsung Galaxy S10 camera, we took pictures of the bird from eight of the major cardinal directions (N,NW,NE, etc.) and additionally one from the bottom. All photos were taken in a similar lighting environment, distance, and angle. Given these principle images, we then manually cropped out all but the pixels that corresponded to the bird, according to our perceived view. From these cropped images, we produced masks that we manually labeled regions of interest namely the feet, body and hair as shown in Figure 3.1.1.

**Figure 3.1.1. Mask Generation:** *These images show our cropping (left), mask generation (middle) and labeling (right) of the porcelain duck.*

From these cropped and labeled images, we were then able to produce a training set and a labeled validation set. We accomplished this by given a set of background images we overlaid the cropped image onto background with random transformations. The random transformations included, rotations from $0^0$ to $360^0$, scale from x0.5 to x2, translations depicted from the bounds of the background, and flips vertically and

horizontally. The background images used varied in lighting intensity, contours and color as shown in Figure 3.1.2.

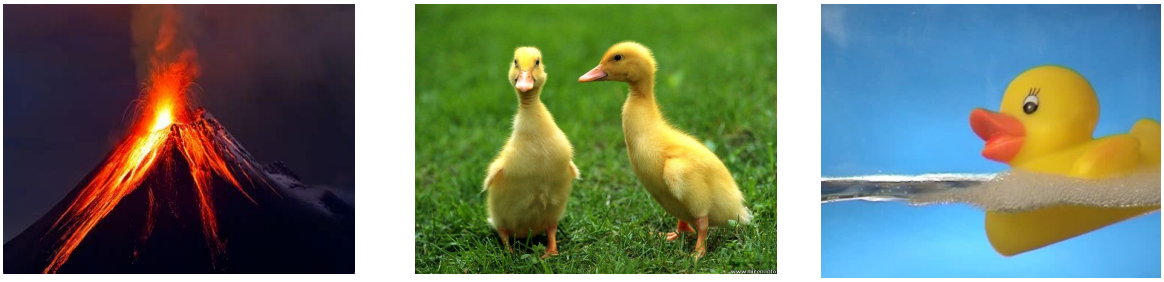


**Figure 3.1.2 Background Examples:** *These images show a sample of the backgrounds chosen for our dataset generation. The volcano image (left) was chosen as a variety in luminosity between the red lava and grey smoke fumes. The ducks image (middle) was chosen as it has a similar body shape to the porcelain duck. The rubber duck image was chosen as it had a similar background to the porcelain duck and an accent of yellow dictated by the rubber duck in the image. [8,9,10]*

In addition to random backgrounds, we incorporated the CamSeq01 dataset [11] as it was already labeled with 32 object classes. As our general approach is multi region classification, we found this dataset to be suitable for additional testing. The CamSeq01 dataset included 101, 960x720 pixel images in which each pixel was manually assigned a label. CamSeq01 dataset was produced by driving in the streets of Cambridge UK, with a Panasonic HVX200 digital camera mounted on the passenger seat in a car. The CamSeq01 dataset, both testing and labeled sets, were then overlaid by the manual duck imaged we produced and three classes were added for hair, feet and body shown

in Figure 3.1.3. For each frame in the CamSeq01 set, a random pose (N,NW,NE etc.), scale (x0.5 to x2), rotation ($0^0$ to $360^0$), translation, and flip (horizontally and vertically) were set.

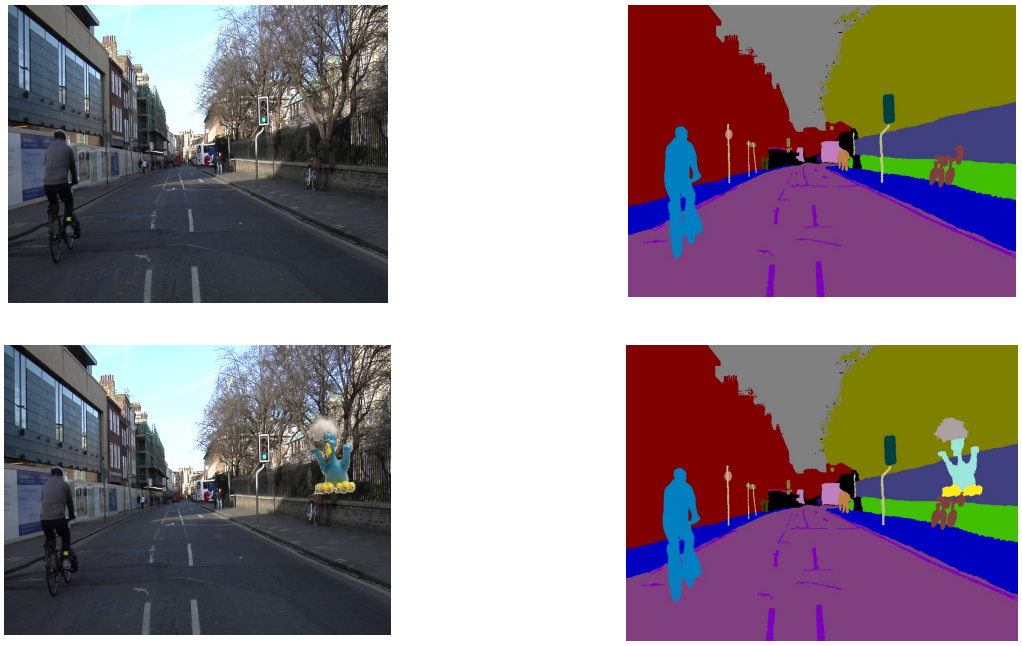


**Figure 3.1.3 CamSeq01+Bird Dataset:** *From the original CamSeq01 dataset (Top Left) and its labels (Top Right) we can superimpose the bird from our crops (Bottom Left) and also superimpose our manually labeled set onto the CamSeq01 labels (Bottom Right).*

## 2.2. UNet

UNet is a fully convolutional network (FCN) such that it only contains convolutional layers followed by rectified linear units (ReLU), and max pooling layers. UNet consists of an encoding path and a decoding path that forms a U-shape when as drawn as a diagram such as Figure 3.2.1. The encoding path, is used to obtain context in the image mainly context information. Each encoding layer includes two 3x3

convolutions, each followed by a ReLU. The resulting feature map is then pooled such that every 2x2 block of the input from the resulting encoding layer is represented by the maximum within the feature map. This allows the amount of features to be down sampled and allows the what in the image to be recognized but loses the whereabouts of the object. Thus, within the decoding path, the opposite occurs [12].

Within the decoding path, the images are up-sampled via transposed convolutional layers to obtain specific location information of objects in the image. For the network to learn and improve, skip connections concatenate the output of the transposed convolution layer from the encoding layer to the decoding layer. Using the CamSeq01 dataset we trained UNet with 70 images in 18 epochs over 100 iterations, and tested with 31 images.

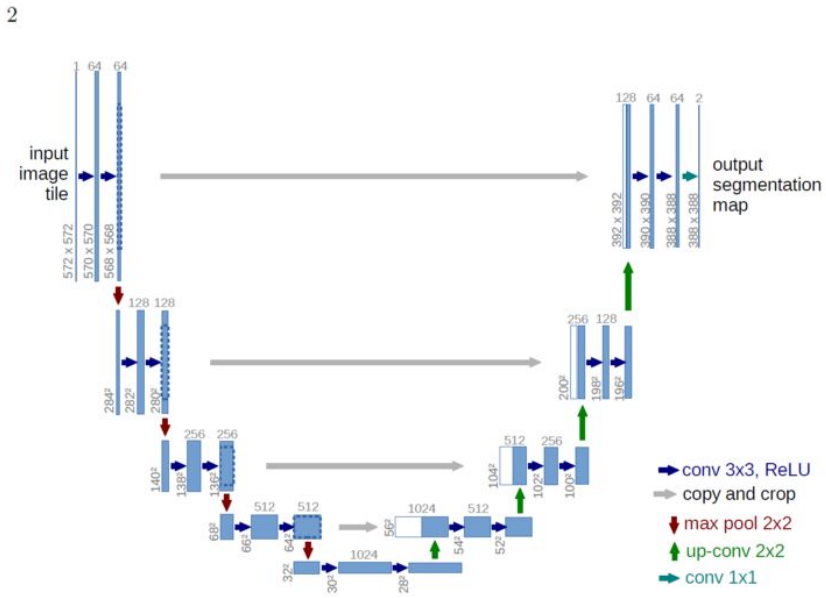



**Figure 3.2.1. UNet Architecture (O. Ronneberger. 2017):** The *UNet architecture as shown above shows the encoding layer on the left and the decoding layer on the right.*

*As the encoding layer increases the number of features to detect what an object is and then the decoding layer minimizes the number of features to represent within the segmentation map.*

## **2.3.** Manual Contour Separation

The color of each part of the bird is distinct, and the borders between the segments are very distinct. It is therefore possible to manually find the contours of different segments of the bird manually by using simple color isolation, thresholding, and contouring tools. This method was chosen due to the very distinct color region in the object and the monocolored nature of each region.

First the brightness of each color component is extracted from an image. Then the images are converted to grayscales so that there are three grey images each corresponding to the brightness of each color component. These images are then combined in such a way to maximize the combined brightness of each color component. The particular signature of each color was determined experimentally. The turquoise body of the bird is identified by adding the green intensity to two times the blue intensity, and subtracting 2.5 times the red intensity. The yellow feet and bill are identified by adding the green and red component and subtracting six times the blue. The black component is found by inverting the brightness of each image and adding them together. The white is determined by adding all of the components and then subtracting the calculated yellow value. This was done because the yellow is very bright, and due to its reflective nature, was often brighter than the white hair.

This process yields several brightness maps corresponding to the places in an image which best match the colors on the object. The following figures show how the colors are segmented.

**Figure 3.3.1: Bird in CMYK:** *Left: Original image, Right: cyan, white, yellow, and black levels in the image respectively. A binary threshold was then applied to each of these images in order to separate the regions of color more distinctly.*

**Figure 3.3.2 Bird in CMYK + Thresholding:** *Left: Original image, Right: cyan, white, yellow, and black levels after thresholding*

Next, the opencv contouring tool was used to find the contours of all of the shapes of each color. In order to determine which of these contours make up the shape, their relative sizes and locations are compared. Beginning with the largest cyan contour, the centroids of the nearby yellow regions are compared in size and position. If the size of the yellow regions is within the correct ratio tolerance, between ⅓ and 1/30 of the area of the body. The are considered to be its nose and feet. The same occurs for the white areas, if a white area is between ½ and ⅕ the area of the body, it is considered to be part of the hair. The centroids of the hair, body, and feet/nose of the bird are then connected to create the computer's understanding of the "skeleton" of the bird and a bounding box is drawn around the area which is believes the bird occupies.

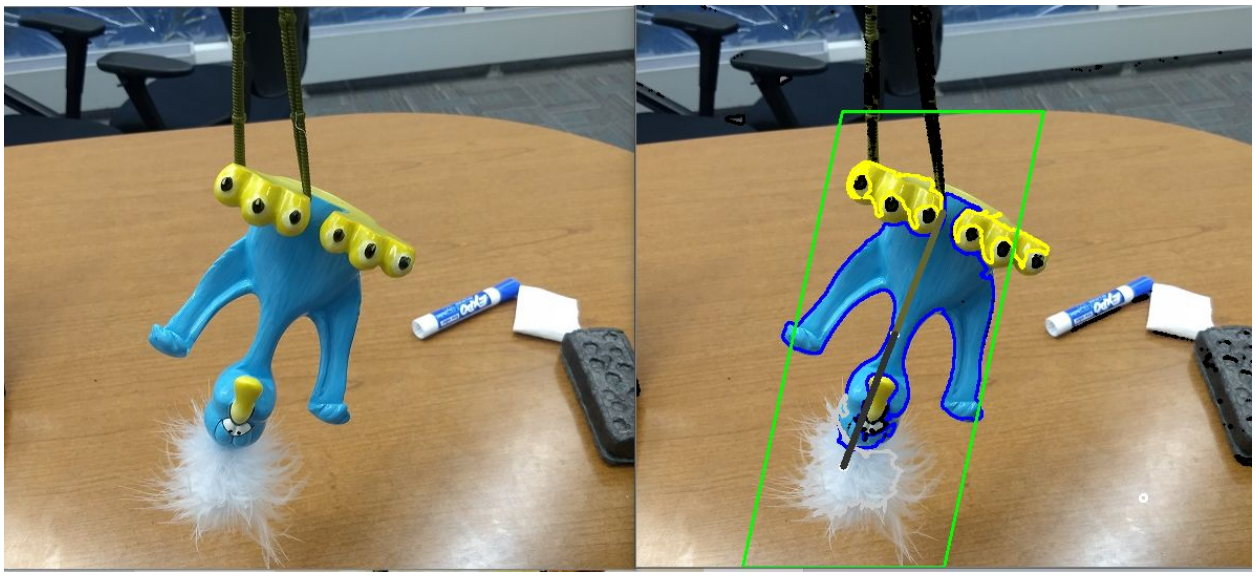


**Figure 3.3.3 Bounding Box Generation:** *Left: Original Image, Right, Image with bounding box, contours, and "skeleton"*

## 2.4. Assessment Protocols

Measuring our detection precision of the bird, we made use of the Mean Average Precision (mAP) confidence measure [21] and the 2007 PASCAL Visual Objects Classes Challenge (PVOC) [20]. This metric is represented by the following set of equations and pseudo code:

$$IOU = \frac{Intersection\ of\ prediction\ and\ ground\ truth\ masks}{Union\ of\ predicted\ and\ ground\ truth\ masks} \quad (13)$$

$$TP = True\ Positive$$

$$FP = False\ Positive$$

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

Pseudocode for mAP:

**Input:** *Set of predicted and ground truth masks*

**Output:** *Mean Average Precision of an algorithm over a whole dataset.*

*For all thresholds, 0 to 1, by 0.1 interval:*

*For each data point in the data set:*

*Compute Precision where:*

*TP = IOU ≥ threshold*

*FP = IOU < threshold*

*Average Precision = mean of Precision from all data points.*

Where mAP is the mean of Average Precision, PVOC is the Average Precision at a constant .5 threshold. mAP is a good tool for evaluation due to how it represents an intersection of a union, meaning that for our purpose [15], it would suffice for comparing predicted bird pixels to known/target bird pixels. We would take the matrices of the predictive image as well as the true image and run mAP on it to determine whether the binary masks were the same. If there was a difference, indicating a lack of precision by the classifier, that difference would be saved and at the end, all the differences would

be averaged together to create the mean of the algorithm's average precision. mAP's accuracy is a good representation when there is a normal distribution present within the sample data. If several outliers are present however, then the mean average precision will be skewed. This problem is avoided by also measuring the average normalized Euclidean distance [16] in relation to the mAP measurement.

# 3. Results

## 3.1. UNet

From testing with our UNet implementation using the CamSeq01 dataset, we found that overall it worked well with a ~90% mAP over all classes as shown in Figure 4.1.1. Looking at just the head, feet, and hair classes however, show that UNet could classify these with a ~45% mAP. This can be seen from looking at the predicted labels of the dataset in Figure 4.1.2, such that segments are disjointed, contain loose pixels, or extend the area of predicted pixels. This error produced

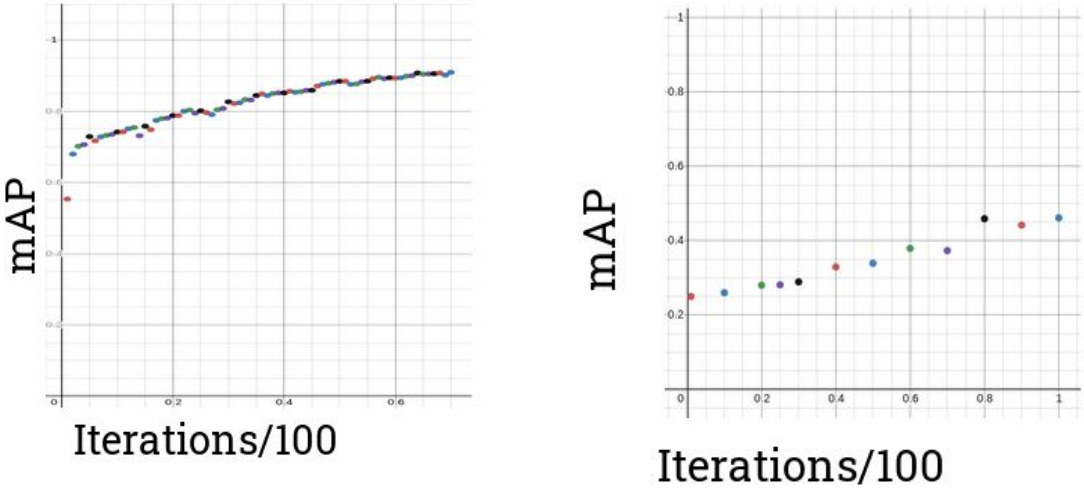



**Figure 4.1.1. mAP Testing 100 Iterations:** *The mAP when accounting for all 36 classes (left) was 90% after 100 iterations using UNet segmentation. When only accounting for the duck classes (hair,body,feet) the mAP was ~45% after 100 iterations (right).*

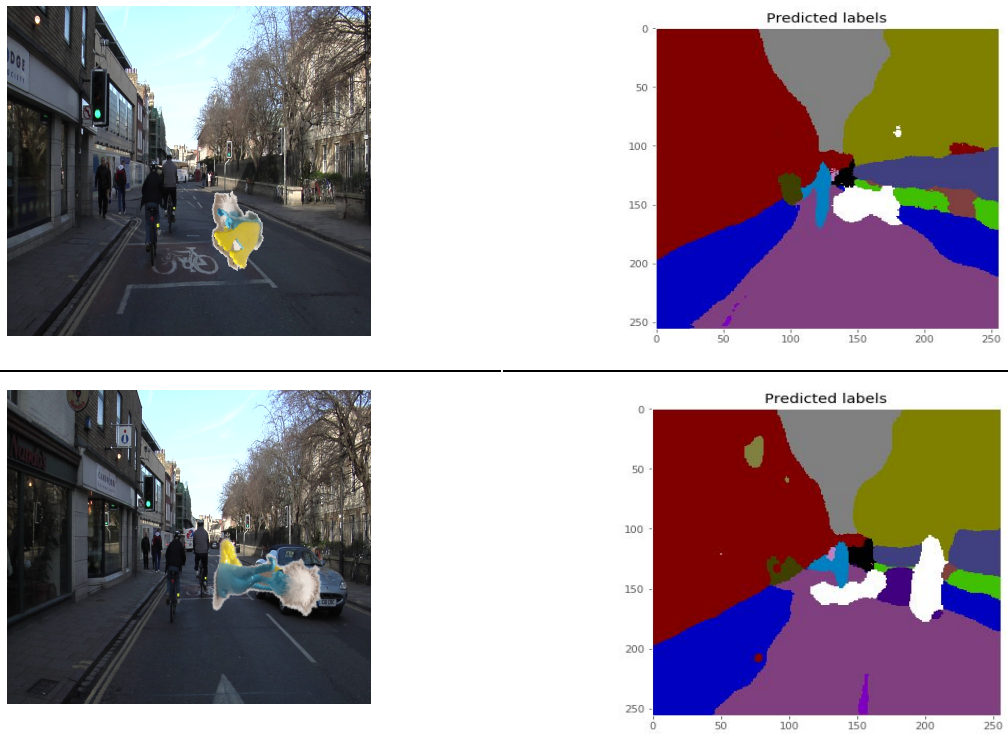



**Figure 4.1.2. UNet Predicted Labels:** *From two sets of actual image and predicted labels we can see that predicted labels may have stray pixels (Top) or segments that are elongated or disjointed (Bottom).*

## 3.2. Manual Contour Separation

The Manual Contour Separation method was very successful at identifying the position and orientation of the object give certain conditions.

- The background did not contain colors that were very close in size and color to object in the image.
- The feet, hair, and nose of the bird were visible.
- The lightning was full spectrum, and did not have any particular hue.

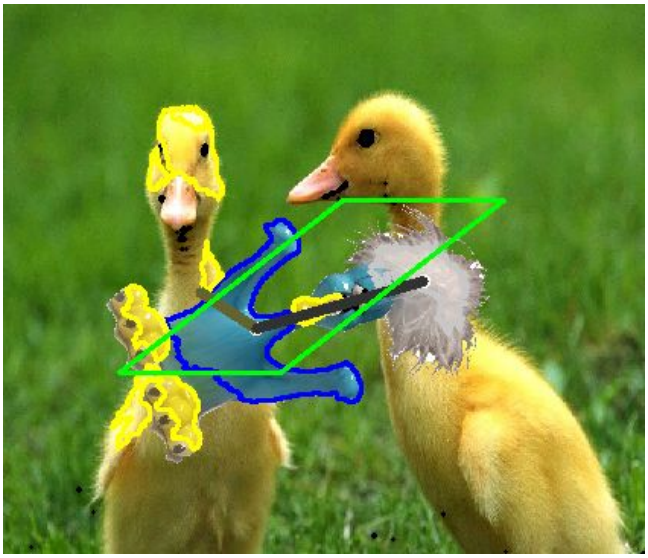


**Figure 4.2.1 Flawed Identification:***: Flawed identification due to background duckling head mistaken for foot*

If those conditions were met, the object detector functioned invariant to scale, rotation, and position. Additionally, the system could handle occlusion as long as it did not obscure entire feature, as could recognise the object even with a reasonable degree of blur.

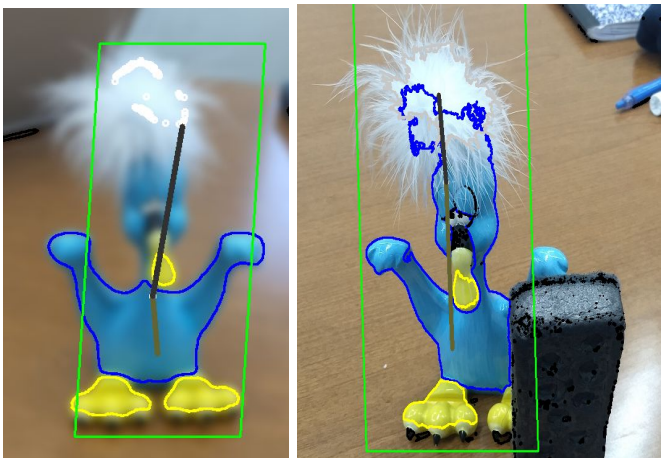


**Figure 4.2.2 Blur and Partial Occlusion**: *Blurred image (Left), Partially occluded image (Right)*

# 4. Conclusions

Overall, our idea of detecting the duck based on our knowledge of the item worked well as a basis. However, due to possible lack of training or the random nature of the dataset, the duck proved difficult for UNet to distinguish patterns. Additionally, most segments detected by UNet were false positives and have very low precision. We suspect that the low precision of the duck classes is due to the sporadic nature of the bird in the generated dataset. As all labels other than those corresponding to the bird, tend to have the same relative location, and orientation throughout the dataset, however this conclusion has yet to be tested. The UNet method did however include nine angles of the bird to test on, was scale, rotation, and flip invariant.

The manual method of detecting color regions worked well but had flaws such that it was influenced by the background, and could only detect the bird when facing forward. Even with these limitations, manual segmentation had 2D rotational, scale, and translation invariance. Due to the nature of our generative dataset, we were unable to test for variance in brightness. However, this was a deliberate choice as we feel the object is intrinsically robust to changes in brightness due to the sharp boundaries and high contrast between regions.

# References

[1] “Object Detection Guide,” *Fritz A.I*. [Online]. Available: https://www.fritz.ai/object-detection/. [Accessed: 12-Dec-2019].

[2] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.*

[3] Navneet Dalal, Bill Triggs. “Histograms of Oriented Gradients for Human Detection,” *International Conference on Computer Vision & Pattern Recognition (CVPR ’05)*, Jun 2005, San Diego, United States. pp.886–893, ff10.1109/CVPR.2005.177ff. Ffinria-00548512

[4] D. Lowe, “Object recognition from local scale-invariant features,” *Proceedings of the Seventh IEEE International Conference on Computer Vision*, 1999.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[6] R. Gandhi, “R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms,” *Medium*, 09-Jul-2018. [Online]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. [Accessed: 12-Dec-2019].

[7] S.-H. Tsang, “Review: Faster R-CNN (Object Detection),” *Medium*, 20-Mar-2019. [Online]. Available: https://towardsdatascience.com/review-faster-r-cnn-object-detection-f5685cb30202. [Accessed: 12-Dec-2019].

[8] https://www.sporcle.com/blog/2015/03/10-things-learned-sporcle-week-6/. (2019). [image].

[9] https://sites.psu.edu/emmagambino/2013/04/04/duckies/. (2019). [image].

[10] https://www.shutterstock.com/video/clip-2346854-yellow-plastic-duck-floating-swimming-pool-filmed. (2019). [image].

[11] Julien Fauqueur, Gabriel Brostow, Roberto Cipolla, Assisted Video Object Labeling By Joint Tracking of Regions and Keypoints, IEEE International Conference on Computer Vision (ICCV'2007) Interactive Computer Vision Workshop. Rio de Janeiro, Brazil, October 2007

[12] O. Ronneberger, “Invited Talk: U-Net Convolutional Networks for Biomedical Image Segmentation,” Informatik aktuell Bildverarbeitung für die Medizin 2017, pp. 3–3, 2017.

[13] A. Rosenberg and J. Hirschberg, "V-Measure: A conditional entropy-based external cluster evaluation measure," Association for Computational Linguistics, pp. 410-420, 2007.

[14] E. Yilmaz and J. A. Aslam, "Estimating average precision with incomplete and imperfect judgements," CIKM '06 Proceedings of the 15th ACM international conference on Information and knowledge management, pp. 102-111, 2006.

[15] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, “A support vector method for optimizing average precision,” in Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’07, 2007.

[16] P.-E. Forssen, “Maximally Stable Colour Regions for Recognition and Matching,” in 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007.

# Appendices

UNet Code Repository: https://github.com/Mcdonoughd/ComputerVision

Manual Contour Separation: https://github.com/briggscalum/ComputerVision.git