NUMFOCUS
OPEN CODE = BETTER SCIENCE

Data Retriever

Google Summer of Code Project Proposal

Developing an Advanced Image Recognition Model for Bird Nest Detection in UAV Imagery

Contact Information:

Full Name: **Om Doiphode**

University: **Veermata Jijabai Technological Institute (VJTI), Mumbai**

Course: **Bachelor of Technology (BTECH)**

Branch: **Computer Engineering**

Email: **omdoiphode161@gmail.com**

oadoiphode_b21@ce.vjti.ac.in

Physical Location: **Mumbai, Maharashtra, India**

Time Zone: **Indian Standard Time (UTC+5.30)**

GitHub

LinkedIn

Resume

To give feedback, please contact the email address.

Content

1. Overview

The use of **unmanned aerial vehicle (UAV)** images for automatic object detection has important application prospects, such as the detection of birds' nests. The traditional bird's nest detection methods mainly include the study of morphological characteristics of the bird's nest. These methods have poor applicability and low accuracy.

This proposal aims to address the **key challenges in UAV-based object detection**, with a specific focus on identifying and locating birds' nests from aerial imagery. The primary objective is to develop **an advanced deep learning model** capable of accurately detecting and classifying nests in UAV imagery, overcoming the limitations of conventional approaches.

The proposed deep learning model will leverage state-of-the-art computer vision and machine learning techniques to efficiently process and analyze high-resolution UAV images. By leveraging the power of deep neural networks and large datasets, the model can learn intricate patterns and features associated with birds' nests, enabling robust and accurate detection even in challenging scenarios.

Successful implementation of this project will significantly contribute to the field of UAV-based object detection and have far-reaching implications for ecological research, conservation efforts, and environmental monitoring. By automating the process of nest detection, researchers and conservationists can gain valuable insights into bird populations, nesting patterns, and habitat dynamics, ultimately informing evidence-based decision-making and supporting sustainable ecosystem management practices.

2. Goals

- A state-of-the-art image recognition model specifically designed for nest detection in Everglades UAV imagery.
- Improved accuracy and consistency, validated through between-week accuracy assessments.
- Identification of egg detections in raw imagery, contributing to the enhancement of the model and expanding the dataset for future research.

3. In-Depth Description

3.1 Difference between UAV Object Detection and Common Object Detection

In normal view, the datasets used for object detection algorithms are mostly taken by handheld cameras or fixed positions, so most of the images are side views. However, UAV aerial images have different characteristics from ordinary view images because they are taken from a top-down view. This means that the object detection algorithms in normal view cannot be directly applied to UAV aerial view. Challenges arise when applying object detection algorithms designed for normal view images to UAV aerial images.

Image Quality Issues: UAV aerial images often suffer from various quality degradations such as jitter, blur, low resolution, lighting changes, and distortion. These issues necessitate preprocessing steps to enhance video quality, thereby improving the effectiveness of detection methods.

Object Scale and Density Disparity: Objects in UAV aerial views appear smaller and exhibit varying densities compared to their counterparts in normal view images. For instance, pedestrians and cars may occupy fewer pixels and are irregularly distributed, leading to challenges in multi-object detection and requiring specialized network modules for feature extraction.

Differences in Occlusion Patterns: Occlusions in aerial views differ from those in normal views. While normal view occlusions typically involve objects obstructing each other, aerial view occlusions often stem from environmental elements like buildings and trees. This variance necessitates tailored algorithms to address occlusion challenges specific to UAV aerial images.

3.2 Challenges in UAV Object Detection

1. **Increase in Small Objects:** UAV remote sensing images encompass a wide scale range of objects, from buildings to pedestrians, animals, and mountains. Small objects, which constitute a tiny proportion of the image, pose a significant detection challenge. Bird nests often appear as small objects in UAV remote

sensing images, making them challenging to detect accurately. These tiny structures constitute a minuscule portion of the overall image, posing significant difficulties for object detection algorithms.

2. **Complex Backgrounds:** Dense object areas within UAV images often contain numerous identical items, leading to a higher likelihood of false detections. Additionally, background noise in UAV images can obscure or weaken the object, hindering continuous and complete detection. UAV images of natural environments, such as forests or woodlands, where bird nests are typically found, can have highly complex backgrounds. Dense vegetation, foliage, and background noise can obscure or camouflage nests, hindering their continuous and complete detection.

3. **Category Imbalance:** UAV-captured images may exhibit class imbalance, with a disproportionately large number of non-nest objects compared to actual bird nests. This imbalance can bias the detector towards predicting more common objects, making it challenging to identify the relatively scarce nest instances accurately.

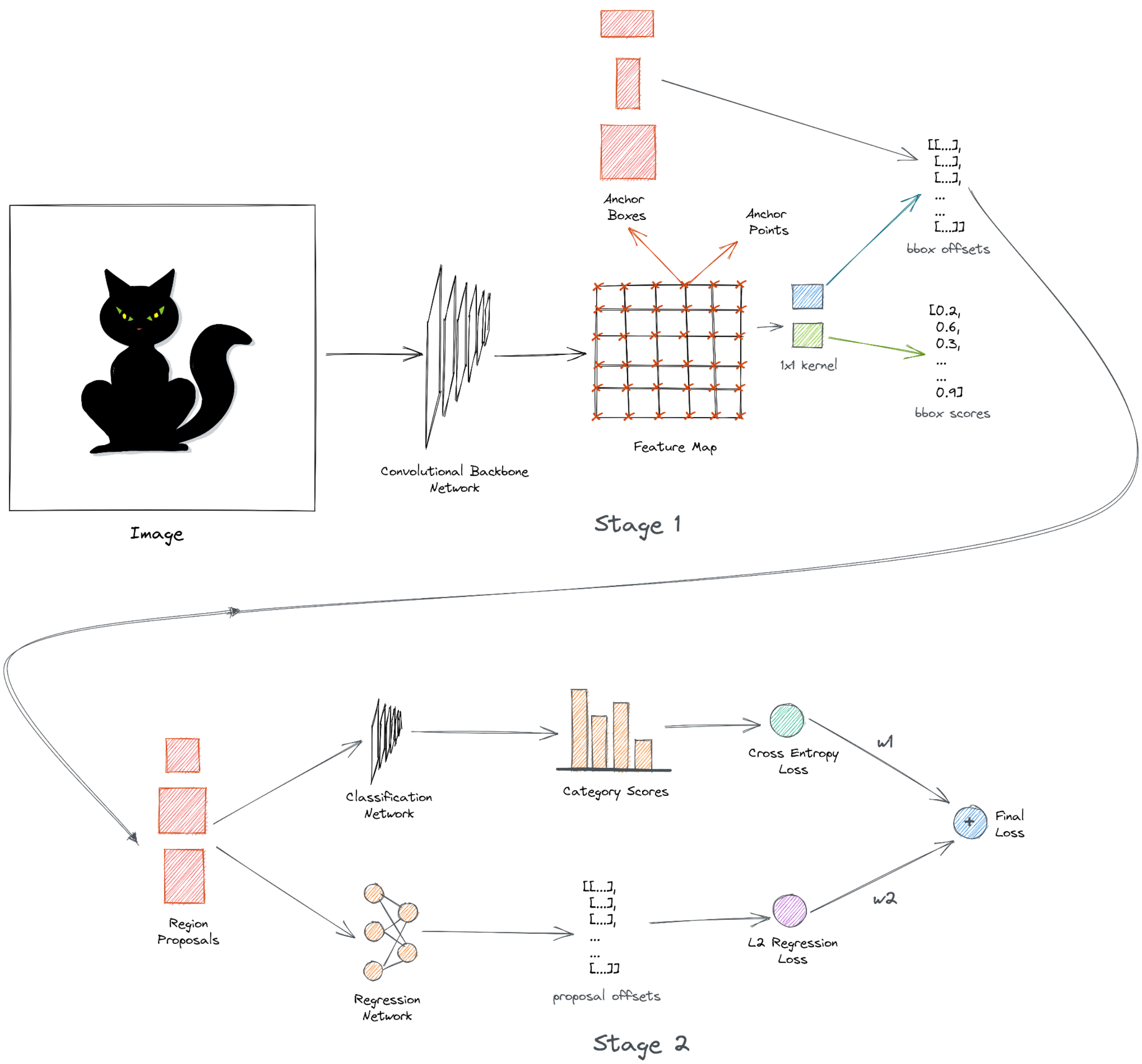
4. **Object Rotation:** Objects in UAV images can manifest in various positions and orientations, unlike traditional object detection assumptions that consider objects to be predominantly horizontal. This variability introduces challenges for algorithms reliant on object shape and appearance. Moreover, rotating objects may alter their shape and appearance in the image, further complicating accurate detection.

3.3 Approach

For detecting nests, I will be using Faster RCNN with ROI mining which will help in solving the problem of class imbalance and detection of small objects in UAV images.

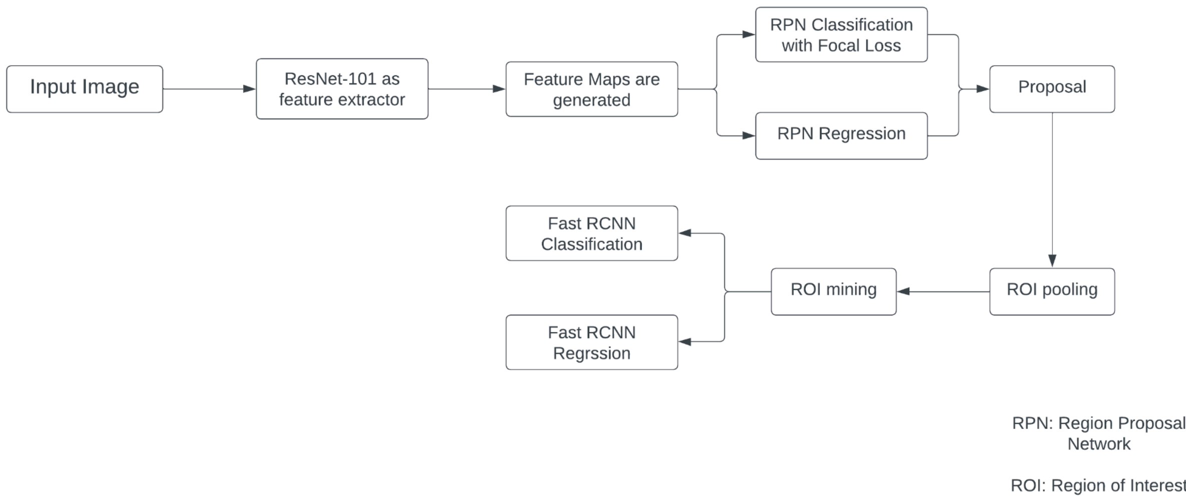
Faster RCNN consists of three parts:

1. Feature Extraction Network
2. Region Proposal Network (RPN)
3. Fast RCNN



Faster RCNN Overall Architecture

3.4 Bird's Nest Detection Network Framework



ROI mining faster region-based convolutional neural networks (RCNN) flow chart

During training, the number of negative samples can vastly outnumber positive ones, often by tens or hundreds. Easy-to-classify samples dominate the reduction in classification loss, leading to a final detector that struggles to identify small birds' nests in complex scenes. In response to this problem, ROI mining will be used.

Here's the detailed process of the ROI Mining method:

1. **Calculate Median Area:**
   a. Before training, calculate the area of all labeled boxes in the bird's nest dataset.
   b. Sort the annotation boxes in descending order based on their areas.
   c. Compute the median area *S* using the areas of the boxes.

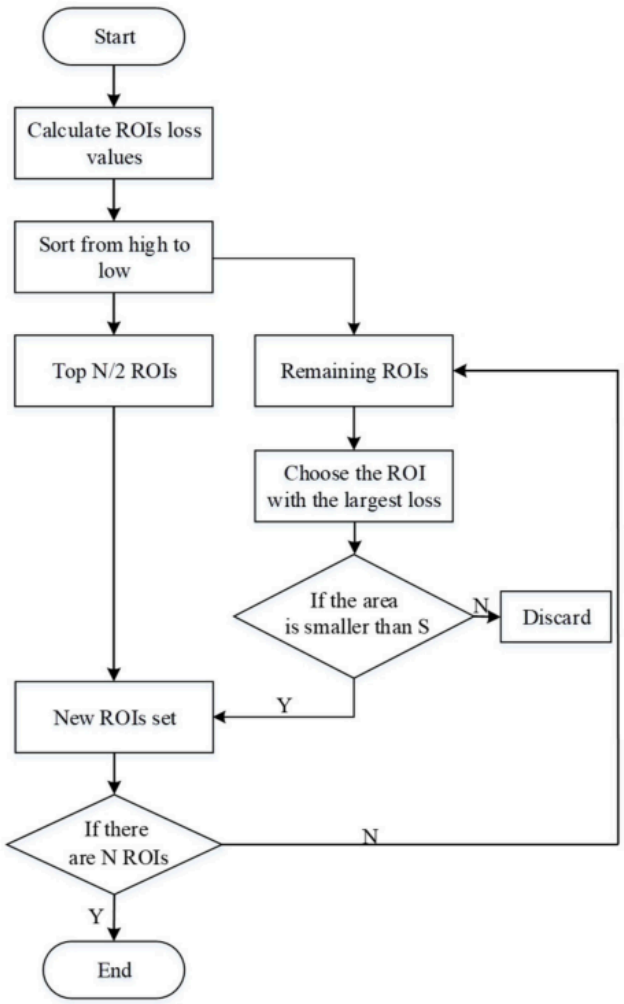
2. **Obtain Candidate ROIs:**
   a. During training, collect all candidate ROIs for input to Fast R-CNN classification and regression loss calculations.
   b. Calculate the total loss value for each ROI and sort them in descending order based on loss value.

3. **Select ROIs for Loss Calculation:**
   a. Choose *N* ROIs for loss calculation.
   b. Place *N/2* ROIs with the highest total loss values at the top of the set.

c. Select the remaining *N/2* ROIs with high loss values and areas smaller than *S*.

4. **Use Selected ROIs:**

   Replace the original *N* ROIs with the selected ones for input to Fast R-CNN classification and regression loss calculations.



Region of interest (ROI) mining flowchart.

Now the Faster RCNN is improved by combining the original Faster RCNN model with the RPN mining as shown in the flowchart. The ROI mining faster RCNN is based on ResNet-101 and is used for extracting features from images. This is done to ensure the accuracy of object detection in real-time applications and considering the detection problem in image samples obtained using an unmanned aerial vehicle (UAV). To improve the accuracy of bounding box coordinates generated by the algorithm, I will use k-means clustering for extracting anchor boxes. In addition, I will use the focal loss function in the RPN stage to balance the number of foreground and background samples.

3.5 Evaluation metrics

I will utilize four evaluation metrics to assess the proposed approach: precision, recall, F1 score, and mean average precision (mAP). In the object detection task, we gauge the accuracy of the final detection results by calculating the ratio of the intersection area between the detected boxes and the annotated boxes to the union area of the two. Detected objects are classified into positive and negative categories, with each category having four possible outcomes: true positive (TP) denotes correctly identified positive samples, false positive (FP) represents incorrectly labeled positive samples, false negative (FN) indicates misclassified negative examples, and true negative (TN) signifies correctly identified negative examples.

Based on this information, we define precision as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

The recall is defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

The F1 score is defined as

$$\text{F1} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Since the detection object in this project is only the bird’s nest category, the AP value is the mAP value. It is expressed as:

$$AP = \int_0^1 p(r)dr$$

where, p and r present the precision and recall, respectively.

3.5 Implementation Details

The implementation details can be found here: Implementation

4. Timeline and Research Plan

Period	Work Expected
4.1 Community Bonding Period (1st May - 26th May)(Get Set)	
Week 1 (4th May - 28th May) : Planning and Bonding	• Set up a Gsoc Blog regarding how I get into this organization and the tasks to be done in the future. • Getting feedback from the mentors about the submitted proposal. • Discuss with mentors the high-level details of the project (goals, priority and deliverables, meeting schedule). • After this, create a general structure of the workflow of the project and adjust the timeline and research plan accordingly. • Research how to implement ROI mining and integrate it with the existing Faster RCNN model.
4.2 Coding (28th May- 26th August)(GO)	
Week 1-2 (28th May- 11th June): Project setup and research	• Set up the development environment with the necessary tools and libraries. • Conduct an in-depth literature review of the paper and related works. • Familiarize myself with the concepts of Faster R-CNN, ROI pooling, and ROI mining.
Week 3-4(12th June - 26th June): Dataset Collection and model training	• Gather UAV imagery datasets relevant to bird nest detection in the Everglades ecosystem. • Preprocess the data, including resizing images, labeling nests, and splitting into

	training and validation sets. • Implement the Faster R-CNN model with ROI pooling and ROI mining. • Train the model on the initial dataset and validate its performance.
Week 5(27th June- 4th July): Mid-term evaluation preparation	• Evaluate the model's accuracy in detecting nests for the initial week. • Record the performance metrics and establish a baseline for comparison. • Prepare documentation for the mid-term evaluations. • Evaluate the performance of the adapted model and the clustering algorithm.
Week 6 (5th July - 12th July): Mid-term evaluation	• Participate in the mid-term evaluation and receive feedback. • Address any issues or suggestions raised during the evaluation.
Week 7-8 (13th July- 27th July): Between week accuracy	• Fine-tune the model based on initial accuracy results. • Measure accuracy, precision, recall, and mAP for each week. • Analyze the consistency of the model's performance over time.
Week 9-10 (28th July - 11th August)	• Conduct thorough testing and debugging to ensure the reliability of the adapted model. • Optimize the model parameters for improved performance.
Week 11-12 (12th August - 26th August): Finalization and documentation	• Finalize the project implementation and results. • Prepare comprehensive documentation, including a detailed report. • Use this time for any final adjustments, optimizations, or additional tasks.

	• Ensure all project deliverables are complete and ready for submission by the end of the coding period.
4.3 Students Submit Code and Final Evaluations (26th Aug- 2nd Sept)	• Create the final evaluation and submit it to mentors.

5. About Me

I am currently a third-year undergraduate student at Veermata Jijabai Institute of Technology, Mumbai pursuing a Bachelor of Technology in Computer Engineering and will graduate in July 2025. Currently, I am an ML consultant at an AI startup in Sunnyvale, California. I have experience in Python, C, C++, React, and Node. I have also done projects in Computer Vision and Deep Learning.

I am a great enthusiast of open source, I have working experience on projects and collaborating with others for developing software solutions, and have made open source contributions. I know the importance of documentation when implementing the code. I commit to adhere strictly to the deadlines mentioned which are planned with considerations of my capabilities. I will be in communication with the mentor regularly discussing ideas, plans, and issues I may face. I have interest in Deep Learning, Machine Learning and wish to pursue the field as a long-term career. This project will be a great opportunity for me to further my skills.

To the concluding part, I think my open source contributions, Projects, and Competitions efficiently signify the set of skill set I have which is a perfect match for all the requirements and the prerequisites of the project, So I am surely confident that I will be able to manage and successfully complete this project within the proposed timeline.

6. Open Source Contributions

1. **DeepForest**
   - **Fixed deprecation warning when running main.deepforest()**
     https://github.com/weecology/DeepForest/pull/375
   - **Fixed Syntax Warning**
     https://github.com/weecology/DeepForest/pull/565
   - **Added doc string for save_dir in evaluate function**
     https://github.com/weecology/DeepForest/pull/596
   - **Fixed error related to tile assignment**
     https://github.com/weecology/DeepForest/pull/599
   - **Updated Configuration.md file**
     https://github.com/weecology/DeepForest/pull/623
   - **Updated environment.yml file**
     https://github.com/weecology/DeepForest/pull/624
   - **Fixed deprecation warning**
     https://github.com/weecology/DeepForest/pull/629
   - **Gradio application for Getting Started Tutorial**
     https://github.com/weecology/DeepForest/pull/637

2. **Ivy**
   - **Add logical_and method to PyTorch Frontend**
     https://github.com/unifyai/ivy/pull/13114 (merged)
   - **Add bitwise_left_shift method to PyTorch Frontend**
     https://github.com/unifyai/ivy/pull/13120 (merged)

3. **MindsDB**
   Implemented insert, update, and delete queries for the carrier service table
   https://github.com/mindsdb/mindsdb/pull/8028

4. **Openvino ToolKit (NNCF)**
   Replaced Runtime Error
   https://github.com/openvinotoolkit/nncf/pull/2412

After contributing to open source for quite a long time I have a good understanding of **git**, **GitHub workflows,** and the version control system.

7. Other Commitments

I have my summer holidays from June to July and won't have any major commitments. I will devote **6** hours daily from **June to July** which I would divide into two parts: 7-8 hours for the actual project and 1-2 hours(learning time) and **3-4** hours daily from **August to September** considering my college hours.

However, I will do my best not to limit myself to working only for a few hours per day but to continue working as long as I can and get the best out of this opportunity.

Note: I have considered extended time durations daily because there can be unprecedented errors and system issues that may slow down the project work. The learning time mentioned is considered because I want this project to be a learning experience rather than just completing it as a task.

8. References

1. K Means clustering algorithm for anchor box generation
2. Bird Nest Detection paper
3. Faster RCNN review
4. A Survey of Object Detection for UAVs Based on Deep Learning