Transformer based Object detection on Images collected using Airbone Optical Sectioning for Forest Search and Rescue

Undergraduate Thesis

Submitted in partial fulfillment of the requirements of
BITS F421T Thesis

By

Abhishek Iyer
ID No. 2018A3TS1105P

Under the supervision of:

Dr. Oliver Bimber
&
Dr. Shishir Maheshwari

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS
December 2021

Declaration of Authorship

I, Abhishek Iyer, declare that this Undergraduate Thesis titled, 'Transformer based Object detection on Images collected using Airbone Optical Sectioning for Forest Search and Rescue' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: 

Date: 26/12/2021

Certificate

This is to certify that the thesis entitled, "*Transformer based Object detection on Images collected using Airbone Optical Sectioning for Forest Search and Rescue*" and submitted by Abhishek Iyer ID No. 2018A3TS1105P in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Co-Supervisor
Dr. Shishir Maheshwari
Asst. Professor,
BITS-Pilani Pilani Campus
Date: December 27, 2021

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

Abstract

Bachelor of Engineering, Electrical and Electronics (Hons.)

Transformer based Object detection on Images collected using Airbone Optical Sectioning for Forest Search and Rescue

by Abhishek Iyer

Recently, a vigorous effort is being put into researching different kinds of drones for various functionalities. The development over time of the technology has allowed us to use drones in multiple industrial sectors. An array of applications that drones are used for as of today are warehouse management, delivery systems, telecommunications repair, national defense, automated agriculture, and more.

Our work is focused on making an autonomous drone which can scan various types of dense forests and make use of airborne optical sectioning(AOS) to collect real time images in the colour and thermal spectrum. A deep learning based classifier can be used in tandem with image processing to robustly classify a human from these images. This allows the drone to be an invaluable tool for forest search and rescue teams. The entire process is autonomous and the drone pings the team when a human is classified.

This process has multiple novel techniques and areas that are being researched further. The main contribution of this thesis is to provide a robust deep learning based object detection classifier.

Acknowledgements

I would like to acknowledge my supervisor, Dr. Oliver Bimber, for providing guidance and his time to mentor me in my thesis.

I would also like to thank Dr. Shishir Maheshwari, my co-supervisor, for giving me constructive feedback throughout the duration of the thesis and showing great enthusiasm about the topic.

Finally, I would like to thank BITS Pilani, Pilani Campus for providing me with the resources and the environment to perform research which has been an immense learning experience for me.

Contents

Chapter 1

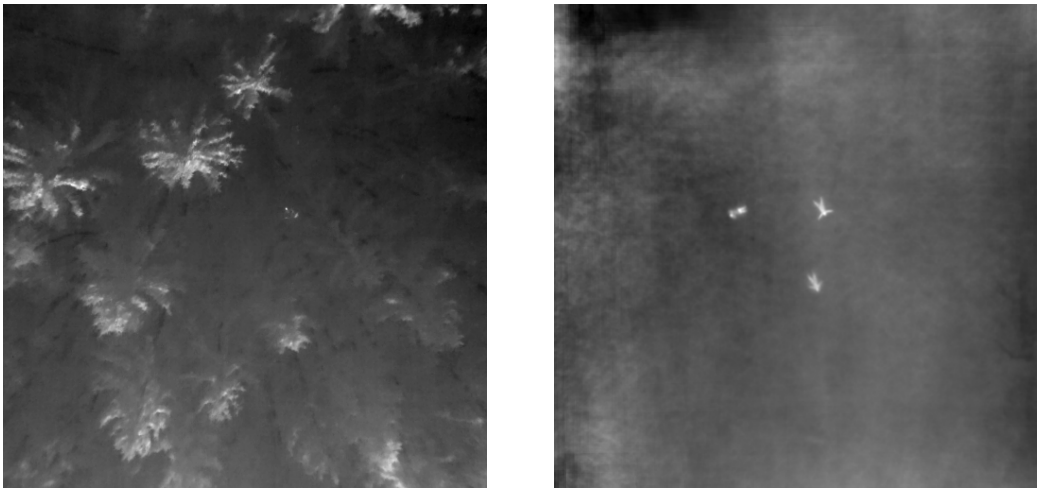
Introduction

1.1 Background

The use of Unmanned Aerial Vehicles (UAVs) is rapidly increasing in various domains. The public sector employs UAVs to perform tasks such as deliveries[7], automatic identification and repair of telecommunication wires[6], to automate an ecosystem like warehouses[57]. They are also being used in a plethora of civil applications such as searching for mines[32] for defence purposes, better situational awareness in policing[44] and disaster management[45] situations, and Search and Rescue(SAR)[46].

The reasons for rapid integration of UAVs in our day to day life are many. To begin with, drones are cheaper to acquire, maintain and operate compared to other aerial manned vehicles. In addition to this, drones can be used flexibly in a multitude of situations such as delivering a payload[52], automatic navigation of environment[50], automating repetitive tasks[16], providing surveillance[35], and even adjusting their size according to the environments[12]. Finally, they also have the benefit of keeping the pilots our of danger as any controlling is done remotely. Hence, it is not surprising to see a lot of effort from all around the world being pumped into researching ways to make drones more effective and completely autonomous. We elaborate on these technical challenges and previously demonstrated methods in Section 1.2.1

SAR operations are performed all around the world and can rack up quite a lot of operational cost as they require a large amount of manpower, expensive equipment and are usually located at terrains that are notoriously challenging to navigate. In 2018, 2723 SAR incidents were reported in the national parks of the United States[47]. The operational costs for these missions summed up to $4.5 Million out of which 36% were from air operations. Similarly, in the same year, the UK carried out 2597 SAR missions and Austria carried out 2292 alpine SAR operations.



(A) Single image captured using a thermal camera

(B) Same image without the occlusion from the forest

FIGURE 1.1: Heavy occlusion from the forest can make SAR very challenging

SAR is a particularly challenging task because of multiple technical reasons. The missions often involve searching a vast area of land which is densely forested. Most of the sunlight is blocked out by the canopy trees and other vegetation. This results in the vision of any aerial searchers always being obstructed as we can see in Fig 1.3. A popular way to combat this challenge is by using thermal cameras. They effectively point out the temperature difference between humans and the forest floor. Autonomous drones could be used in place of manned helicopters to significantly reduce costs and manpower while increasing flexibility. Our work in this thesis pertains to the application of drones to the domain of aerial SAR.

Earlier, it was mentioned that thermal cameras provide us with an effective alternative for the proposed challenge of aerial SAR. There are a few more technical challenges that go along with this such as highly occluded images where only a handful of pixels of the human body would be visible which is very hard for humans and classifiers alike to recognize [43]. An innovative solution to the occlusion problem has been presented by synthetic aperture techniques. This technique emulates a wide aperture sensors by combining the signals from either multiple or a single moving narrow aperture sensor. Airborne Optical Sectioning(AOS) is an effective synthetic aperture technique which generates images with an extremely shallow depth of field allowing us to see past occluding layers such as the canopy of a forest. Further technical details on AOS are mentioned in Section 1.2.2

Object detection and classification are two very popular problems in the domain of computer vision. The approaches to these problems are mainly categorized into traditional approaches and Deep Learning (DL) based approaches. Some popular traditional approaches are Scale Invariant Feature Transform (SIFT)[28], and Speeded Up Robust Features(SURF)[3] algorithm which offer some advantages such as efficiency in problems where DL can be overkill. An added advantage is

that traditional methods are not class specific and perform similarly for any object. In contrast to this, Deep Learning approaches such as Convolutional Neural Networks(CNNs)[22], You Only Look Once(YOLO)[39] and Mask-Region based CNNs (Mask R-CNNs)[14] are heavily dependent on their training datasets. While training, they are computationally expensive but provide better results in real time when the training dataset is representative of the inputs we expect to receive in the real world. While, many problems have a benchmark of human performance which all classifiers try to achieve, aerial SAR is one of those unique problems in which human performance is not that good to begin with. A lot of post processing is required to make results interpretable to humans. This is why it is imperative to have a classifier which scans the real time images for humans and ping the rescue team when it believes it has found a human. Further details of classifiers for object detection will also be covered in Section 1.2.3

1.2 Literature Review

1.2.1 Path Planning

In order to automate drones completely, effective path planning in any environment is a huge obstacle. A variety of methods that tackle this challenge to different degrees have been presented before. They draw inspiration from distinct places. Some techniques are based on sampling such as rapidly exploring random trees[21] and artificial potential fields[54]. We also have algorithms inspired from Biology like the particle swarm optimization method[37][49]. Mathematical models are also used for traditional optimization advantages. A few examples of these methods are mixed-integer linear programming[42], control theory[17], and heuristic approaches [31]. Finally, we have smarter methods from the rapidly developing domain of machine learning. Reinforcement learning based algorithms[5][23] are a very popular approach to this problem. These and other techniques are elaborated in detail in the following reviews of path planning and optimization techniques [1][34].

1.2.2 Airborne Optical Sectioning

The technique of AOS is based on the thought that there is a high statistical likelihood that while the forest ground is heavily occluded from one perspective, it would be relatively unoccluded from multiple perspectives, as explained by the probability model in [19]. This would allow you to attain small pieces of information in multiple timesteps and then stitch them together to highlight a clear contrast and shape between the vegetation and an object on the forest floor.

As shown in Fig 1.2, AOS detects thermal radiation using a drone that samples the forest within its SA range. Knowing the key information about the camera, the drone orientation and the

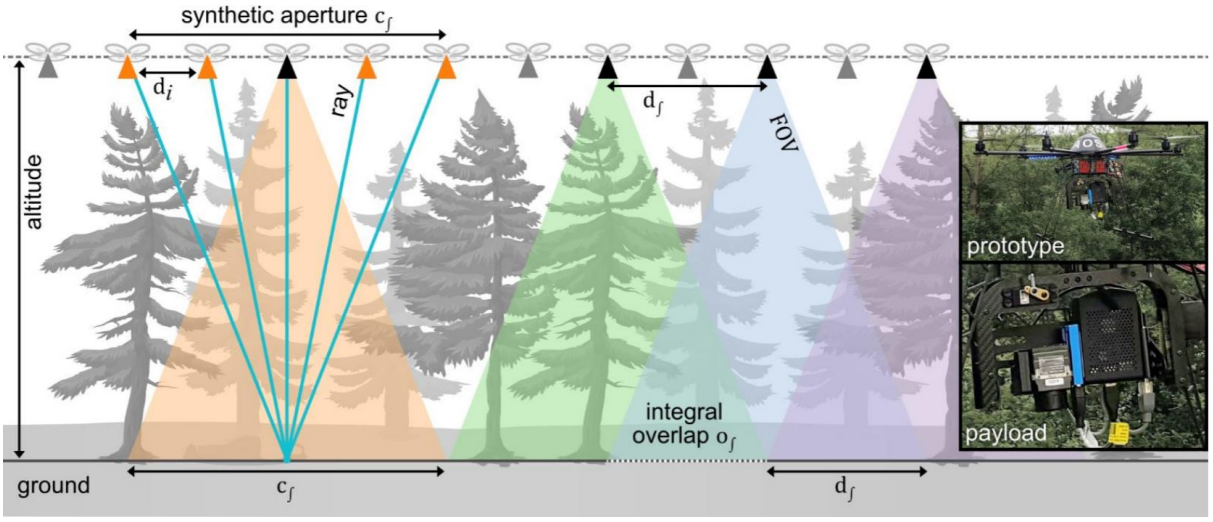


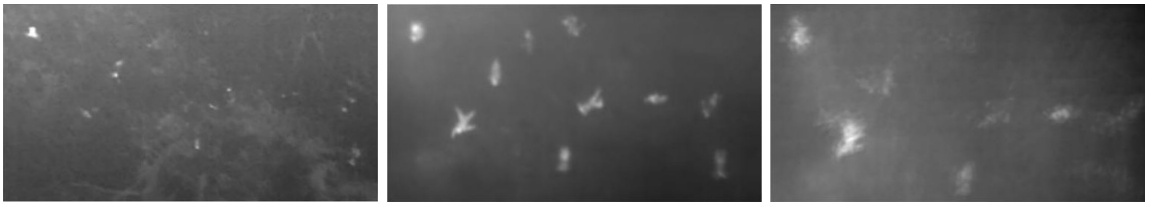
FIGURE 1.2: Synthetic aperture sampling with Airborne Optical Sectioning.

representation of the terrain, we can reconstruct each ray's origin from the ground. Averaging all the rays with the same origin results in an integral image widely free of occlusion and focus on the thermal object in question. Since the remaining occlusion only lowers the contrast of the target [19], reliable classification of the target is possible – even under strong occlusion conditions [47].

The SA was a **2D Sampling Area** in [47], and this allowed us precise pose estimation using computer vision-based reconstruction techniques. This resulted in high quality integral images, but was not usable – for two main reasons. The first one was that, using a 2D sampling area let to length flight times which was not practical in a time critical operation which SAR operations often are.Precise computer vision based pose estimation could not be applied in real time. The computations had to be carried out offline after keeping a record of all the SA waypoints.

Sampling along short and linear flight paths, **1D SAs**, significantly improved this issue [46]. It also made use of the sensors onboard for real time but imprecise data. Barometer altitude, non-differential GPS location, and compass orientation are some of the readings that were used for real-time computations directly on the drone. This led to imprecise pose estimation because of the slightly compromised sensor readings. Sampling in one lesser dimension than before did not help either. However, despite the imprecision, person classification performance was comparable to that of the 2D SA sampling with precise pose estimation approach[46].

Till now, classification with AOS had only been applied to discrete (non-overlapping) integral images, for which the recording speed was quite slow and the image transfer times were long. This process required 30 single images at an interval of 1m before they could be integrated and classified. This meant that there was no overlap between images and a person may have been present in one image but would not be in another and hence only provide a single classification opportunity after a 30s flight time. The approach described in [20] encourages overlap so that



(A) Single heavily occluded image

(B) Integral image processed with 2D SA and precise pose estimation

(C) Integral image processed with 1D SA with GPS/IMU based pose estimation

FIGURE 1.3: Difference between using 2D and 1D synthetic aperture

multiple classification opportunities present themselves. Hence shorter processing times and faster imaging speeds are beneficial.

1.2.3 Object Detection Classifiers

The problem of object detection can be broken down into two distinct challenges. These are to determine where objects are located in the image (object localization) and which category each object belongs to (object classification). So the pipeline of traditional object detection models can be mainly divided into three stages: informative region selection, feature extraction and classification.

Informative region selection poses the challenge of varying aspect ratios and scales for different objects. A naive brute force method is to scan the whole image with a multi scale sliding window. This is computationally expensive, and proposes redundant regions. However, it can also predict all possible positions of the objects in question.

Feature extraction used to be done manually when traditional computer vision algorithms were used. The issue with this approach is that it is impossible to make a feature selection process which is effective for all kinds of features. This is because of varying aspects like occlusions, brightness, and backgrounds. However, this step is left up to the model in Deep Learning approaches so it can be optimized based on the task at hand [58].

Object classification is the task of distinguishing an object from the rest of the categories. Before the advent of Deep Learning, Supported Vector Machines(SVMs)[9] were a popular classifier. However, a significant gain was seen in the accuracy with the proposal of Region based CNNs (R-CNNs)[13]. CNNs are quite different from traditional approaches, they can have very deep architectures which allows them to learn more complex features than the shallow ones. Training the model on a labelled dataset of images similar to the images it is likely to encounter in the test phase enables the model to learn informative object representations without the need for designing features manually [58].

Many models have been proposed for object detection from which the most impactful have been the YOLO, and the R-CNN. While R-CNN utilizes another network to generate region proposals, YOLO accomplishes both the tasks with fixed grid regression. Both the approaches have some advantages and drawbacks. This is discussed in further detail below.

R-CNN: The model splits the work into two phases. Generating regions and classifying each region. For the purpose of generating regions, a selective search[53] method is used. This method keeps in mind the different scales and location of objects, while also using a mix of different strategies (e.g. Colour Spaces, Similarity Measures, and Complementary Starting Regions). Based on this, bounding boxes of varying sizes, aspect ratios and in different locations are generated. Proceeding this, the classifier which uses a CNN backbone, classifies each proposal in terms of if there is an object in the region and if yes, which category does the object belong to. Even though both the bounding box generation and classification tasks are optimized jointly, it becomes a two pass process reducing the speed of the model significantly while also providing impressive accuracy. Soon, models were proposed to improve both the accuracy and the drawback of speed. Faster R-CNN[41] provided a box Average Precision (AP) of 36.8 on the Microsoft COCO test de[25] set while providing 7 Frames Per Second(FPS). Mask R-CNN[14] gained a box AP of 39.8 on the same while achieving around 5FPS.

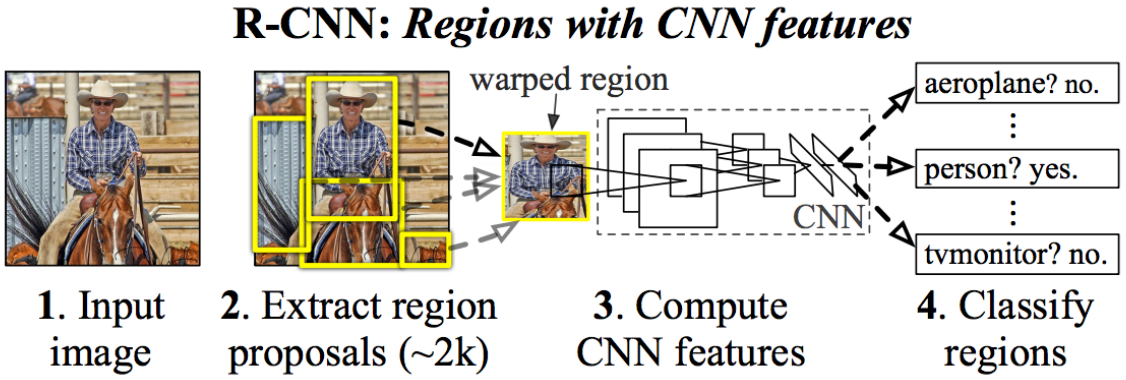


FIGURE 1.4: The 2 phases of Object classification in the R-CNN model

YOLO: This model became famous for its impressive speed of 45 FPS making it very useful for real time object detection. It uses a different approach and only requires one pass to complete both tasks of region generation and classification. The model divides the image into an $S \times S$ grid where each cell of the grid is assumed to have an object at the centre of the grid itself. This provides much fewer region proposals resulting in the speed of the model. However, the limited combinations of aspect ratio, sizing, and localization of the regions causes the model to perform poorly. Several improvements on the initial model have come out since then, like YOLOv3[38] (achieving a box AP of 33 on COCO test dev set) and YOLOv4[4] (achieving a box AP of 43.5 on COCO test dev set) while also giving 31 FPS.

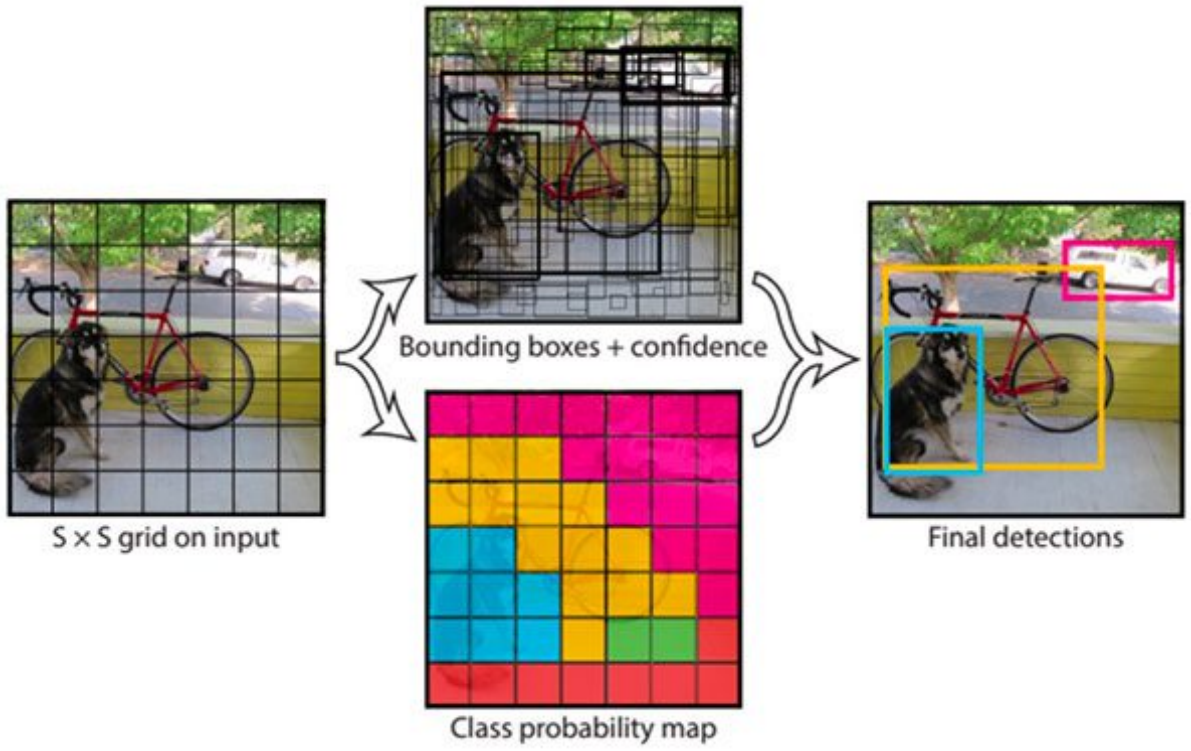


FIGURE 1.5: The single phase approach of YOLO

1.3 Scope of Research

Over the course of Section 1 we have talked about the problem of conducting aerial SAR operation using a fully autonomous drone employing the SA technique of AOS and a deep learning based classifier to perform efficient search and rescue in dense forests and hevily occluded environments. This research problem is multi pronged and requires technical innovation in several domains. The research has been carried out by Dr. Oliver Bimber[20] and his team at JKU Linz, Austria for a couple of years, steadily improving the methods and results achieved. Over this duration they have experimented with predefined path searching and adaptive searching based on potential fields [46] to help achieve complete automation. They have also successfully implemented AOS techniques to heabily suppress false detections and overcoming the problem of heavy occlusion. Their current research work implements a modification of the YOLOv4 classifier which achieves impressive results.

The aim of this thesis is to explore the implementation of Transformers [55], an architecture made popular in the domain of Natural Language Processing (NLP), in the domain of Computer Vision (CV) and specifically real time object detection. We intend to explore if transformers are suitable for this task based on their recent found success in image classification[11][26] and

object detection[10][27]. We also intend to find the strengths and the limitations of this model while also understanding the reason for the same.

Chapter 2

Experiment Setup

2.1 Dataset Collection

The dataset used for this research was collected by the team of Dr. Bimber at Johannes Kepler University, Linz. The images were recorded by the use of an octocopter drone (MikroKopter OktoXL 6S12). It was equipped with both a thermal and an RGB camera. Flir Vue Pro was used to record the thermal images with a fixed focal length lens of $9mm$ at a spectral band of $7.5\mu m$ to $13.5\mu m$. The Sony ALpha 6000 was used to record the RGB images with a $16mm$ to $50mm$ lens at infinite focus.

Both cameras were fixed to a rotatable gimbal setup and were pointing downwards for the entire flightime. A $30m \times 30m$ synthetic aperture was chosen at an altitude of 30m to provide enough clearance over the treeline and to just cause overlapping between the single images taken from that field of view (FOV). The path was planned and uploaded to the drones' software in the form of $1m$ waypoints. All the images were stored on the internal memory cards.

After landing, the high definition, unprocessed, RGB images were collected ($6000px \times 4000px$). These images were used with an Structure-from-Motion (SfM) technique for reconstructing the 3D space from an collection of unordered images, COLMAP[48]. This required 24 minutes for pose estimating of 300 images in our implementations. As both of the cameras were attached to a gimbal, the pose of the thermal camera can be calculated by applying a predefined transformation matrix to the pose of the RGB camera, which is calculated using MATLAB's calibration routine. These images were further processed to generate a functional dataset.

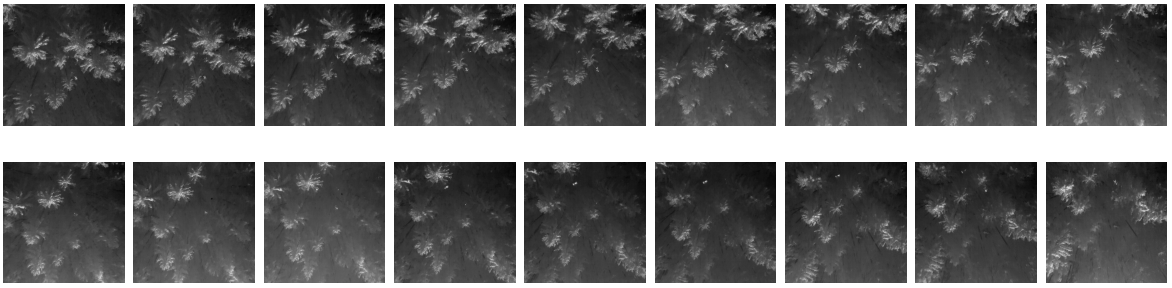
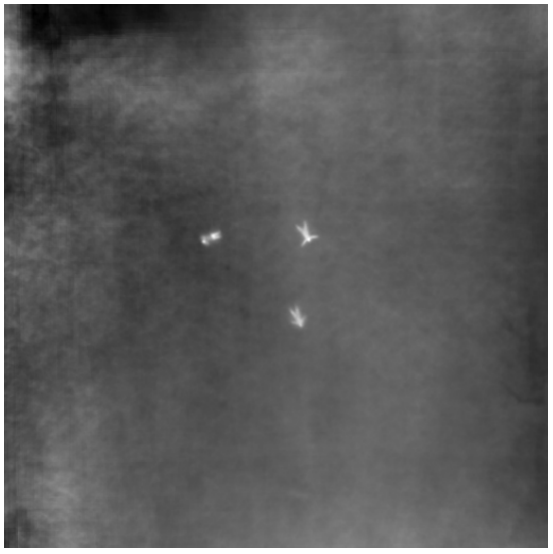
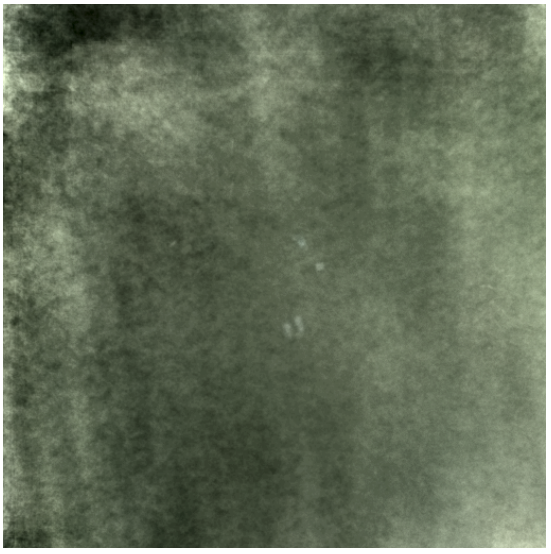


FIGURE 2.1: Series of single images recorded at intervals of 1m



(A) Thermal integral image



(B) RGB integral image

FIGURE 2.2: Integral images calculated from the series of single images above

2.2 Preprocessing

After the images are collected from the internal storage, we start by removing the lens distortions on the thermal images. This is followed by cropping the image to a 50.82°FOV and downsampling the image to a resolution of $512px \times 512px$. The integral images are then calculated using the approach described in 1.2.2 using a GPU implemented with the Nvidia CUDA toolkit. On this system, it takes about 60ms to integrate 360 single images. To visualize the integral image we place a virtual camera at the center of the SA described earlier with a FOV of 50.82°, which is also the FOV of a single image after correction. The labeling of persons was done by using the polygonal tool in MATLAB which were then axis aligned after the augmentation of images and stored in text files with their corresponding images.

The following augmentation techniques were used on the integral images for the test and validation datasets. Random rotation was added by changing the direction of the up-vector of the virtual camera in the intergral image visualization. Additionally, the focal plane parameters were altered.

The altitude of the focal plane was shifted by $\pm 25cm$ and was rotated about its vertical and horizontal axis by $\pm 2°$to provide a total of 270 augmentation combinations per scene. Optional augmentation was also added by applying Adaptive Histogram Equalization (AHE)[36]. The labels for the single images were generated by projecting the labels from the integral images using the known pose matrices. After this, all images were manually inspected and outliers were removed.

2.3 Implementation Details

We got the pretrained weights for the architectures of Swin Transformer and Mask R-CNN which were trained on the MS COCO test set. We then made modifications to the architecture to get the desired process flow. Due to hardware issues, for our proposed model has just been trained for 1 epoch on a section of the train dataset, this is soon to be improved. However, even with just one epoch of training, we get very promising results. When training on the GPU we will also change the hyperparametes to lead to the optimum result.

Chapter 3

Methodology

Write Summary

3.1 Seq2Seq Models

Sequence to sequence models are a subcategory of deep learning models which achieved a lot of success initially in tasks which have a sequence of inputs and require a sequence of outputs, such as text summarization, image captioning, and machine translation. These models follow a general encoder-decoder architecture. Some of these models are explained in detail here[51][8].

3.1.1 Encoder

The process of sequence to sequence models would begin by creating embeddings. This transforms the word into a vector which captures a lot of semantic information of the word. These embeddings would be then given as an input sequentially to an encoder structure which would consist of a Recurrent Neural Network(RNN)[15]. At every timestep, the RNN would take an input vector and the last hidden state. Once an <end> tag is received as the input vector, the current hidden state would be passed to the decoder as a context vector.

3.1.2 Decoder

The decoder would try to reverse the process to generate relevant output by taking the context vector as an input and then generate sequential outputs at every timestep such as words in a sentence. This method suffered from the bottleneck of context features. A solution was proposed by [2] and [29] which was to make use of a technique called Attention.

3.1.3 Attention

The key changes made here were that instead of passing a context vector to the decoder, the decoder received all the hidden states $(H0, H1, H2, ...)$ the encoder had generated in a concatenated format. The second, crucial change was by adding a scoring step to the decoder. In this step, all the hidden vectors would be scored by the decoder. By using the following expression, $\sum(Softmax(H0, H1, H2, ...))$, a context vector would be generated for that particular timestep. The context vector would be concatenated with the last hidden state and passed on to a Feedforward Neural Network (FNN)[18] which would produce the final output(e.g. a word) of the timestep.

3.2 Transformers

The landscape of the sequencing problems drastically changed when Vaswani et. al. introd[55]. It was a model that used attention to boost the speed with which the models could be trained. It also allowed for parallelization. Furthermore, this paper introduced a technique called Multi Headed Self Attention(MHSA) which is used in the encoder of the transformer. This method allowed the model to draw a comparison between a particular input(e.g. a word) and the rest of the inputs in the sequence (e.g. all the other words in the sentence). This provided the insight of how relevant are each of the other inputs, to this input. This approach is described in detail below.

3.2.1 Self Attention

To better understand the process of Self Attention, we can assume the input as a phrase (P_1), which consists of 2 words (I_1 and I_2). Both the inputs I_1 and I_2 are embedded into X_1 and X_2 of constant length l, and passed as an input to the Self Attention layer. Now a Query vector(Q_i), a Key vector(K_i), and a Value vector(V_i) of desired length d are created for each of the embeddings. To do this, random matrices$(W^q, W^k, andW^v)$ of shape $(l \times d)$are generated, which are trained in the training process. The equations below indicate how the necessary vectors are created.

$$Q_1 = W^Q \cdot X_1$$

$$K_1 = W^K \cdot X_1$$

$$V_1 = W^V \cdot X_1$$

Similarly, the vectors $(Q_i, K_i, and V_i$ are created for each of the embeddings. A score is then calculated between X_1 and all the other embeddings to understand the relevance of each of the inputs to X_1. The equations to calculate the score for X_1 is given below.

$$S_1, S_2 = Softmax(\frac{(Q_1 \cdot K_1), (Q_1 \cdot K_2)}{\sqrt{d}})$$

$$Z_1 = V_1 \cdot S_1$$

$$Z_2 = V_2 \cdot S_2$$

These are the outputs of the Self Attention layer. To quicken the process, the same calculations shown above are done with matrices. The inputs are represented as X. So instead of iterating over all inputs, the calculations only take 4 steps.

$$Q = W^Q \times X$$

$$K = W^K \times X$$

$$V = W^V \times X$$

$$Z = Softmax(\frac{(Q \times K^T)}{\sqrt{d}}) \cdot V$$

3.2.2 Multi-Headed Self Attention

MHSA repeats the same process described above parallely where the weights (W^Q, W^K, W^V) are different allowing different representation subspaces. Because we want the layers to be modular in the architecture, a weight (W^O) is created which projects the concatenated outputs of each head $(Z_1, Z_2, Z_3, ...)$ into a resulting matrix, Z of a lower dimension.

3.2.3 General Architecture

The input is embedded with positional encoding, to create the inputs to the encoder block. Each encoder block has the Self Attention layer first, followed by an Add and Normalize layer. The output is then given into parallel FNNs (one for each input), which is finally Added and Normalized again. Multiple encoder blocks like these are stacked on top of each other and the output vectors K and V are passed to the Decoder blocks. The decoders work in a similar way to the encoders and finally, the outputs are passed to a Softmax layer and then to a Linear layer which uses an Output Dictionary to convert the numbers into a word.

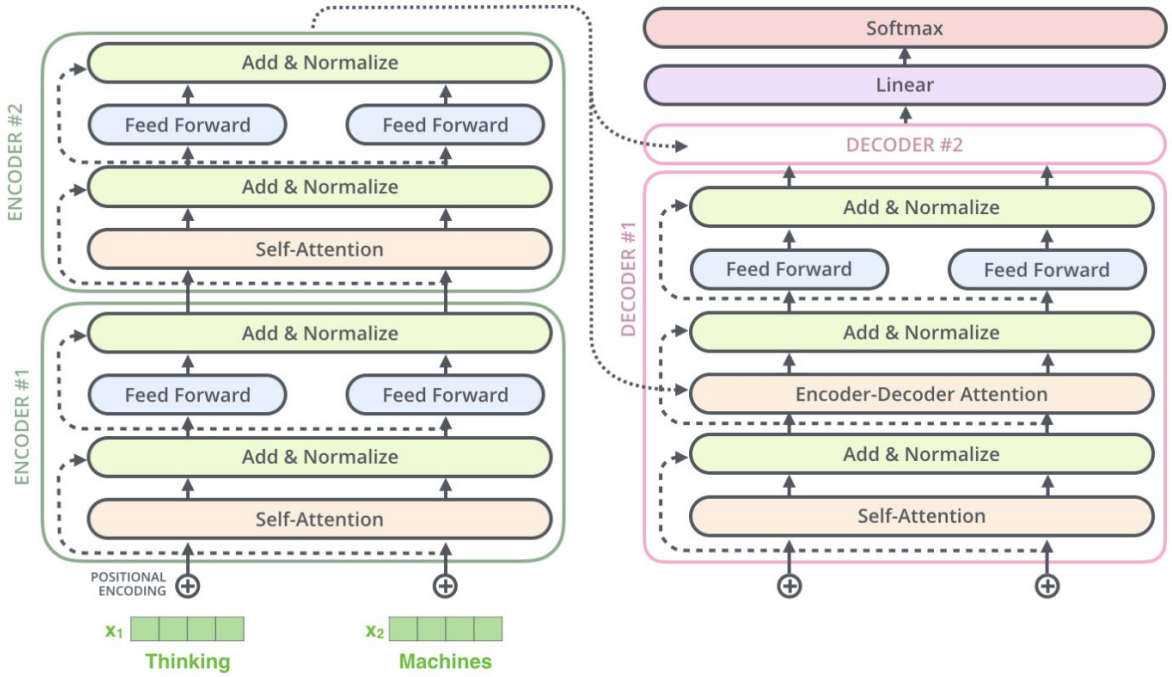


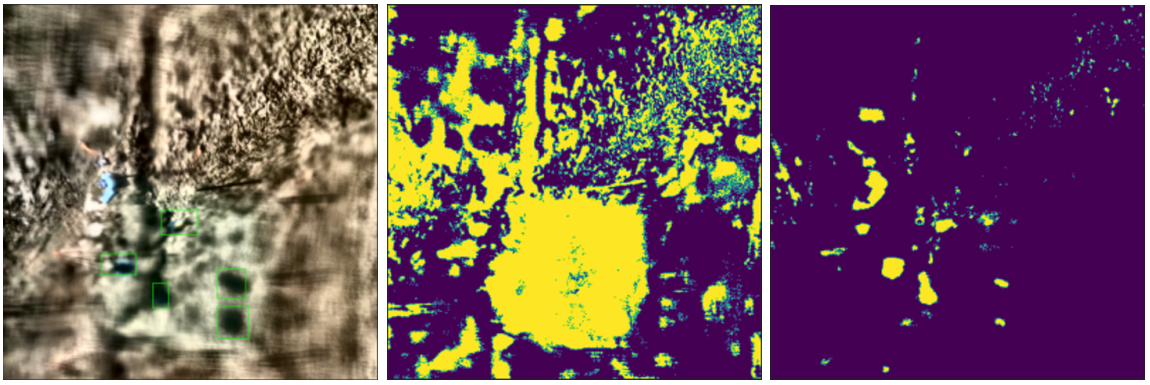
FIGURE 3.1: The transformer encoder-decoder architecture with MHSA

3.2.4 Implementation in Computer Vision

After the introduction of transformers, they were widely applied in the domain of NLP. Because the attention calculating function is quadratically related to the dimensions of the input, attention was too costly to calculate between the pixels of a high definition image. Due to this the CV domain remained undiscovered using transformers, until Vision Transformers were introduced by Google[11]. They proposed that with further research, convolutions can be completely replaced with attention and transformers could be used as the backbone of complex architectures. Since then, many models have been proposed for the different tasks in the domain of Computer Vision - Image Classification [11], Object Detection [10], Instance Segmentation [27], 3D reconstruction [56], Video Analysis [33][24], Biomedical Signal Processing [30], and research is ongoing in bringing transformers into the CV domain as a robust model. This thesis also makes use of transformers as a backbone to make a robust model at Real Time Object Detection.

3.3 Inspired R-Swin

This model takes inspiration from the R-CNN model [14] and uses Shifted Window transformer backbone [26] with further modifications to achieve the best results. Due to hardware limitations, the proposed model has only been finetuned for one epoch,leading to under-performing results. This is to be taken care of in the future. The model is described in detail below.



(A) Original Integral Image

(B) Integral image with RX detector applied (threshold = 0.5)

(C) Integral image with RX detector applied (threshold = 0.97)

FIGURE 3.2: Filtering pixels by detecting anomalies with RX detector

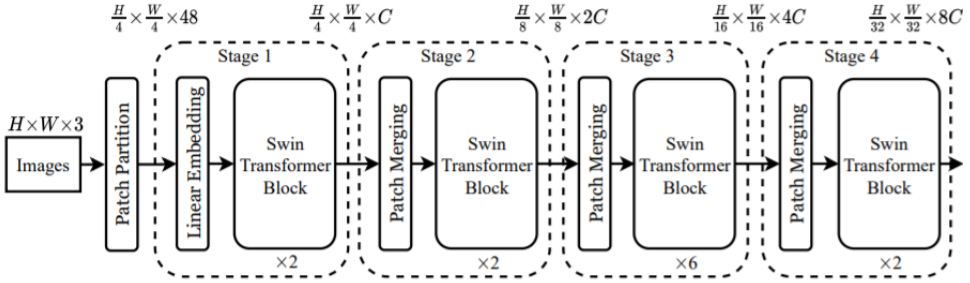
3.3.1 Filtered Anchor Box Generation

An image of shape $H \times W \times C$ is given as the input to the Reed-Xiaoli(RX) [40] algorithm. Th algorithm reads the image as a data cube of $H \times W$ pixels and C channels. The RX score for each pixel is computed by the equation below, where r is the pixel for which the RX score is being calculated, μ_c is the spectral mean and the $K_{L\times L}$ is the sample co-variance matrix.

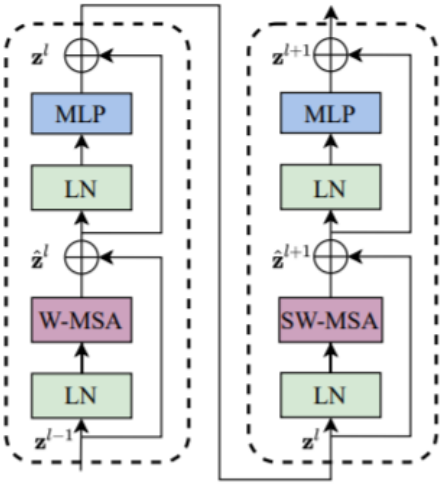
$$D_{RX} = (r - \mu_c)^T K_{L\times L}^{-1}(r - \mu_c)$$

A matrix, D_{RX}, of shape $H \times W$ is generated where each value is the RX score of each corresponding pixel. This matrix is then compared to a threshold estimated from the cumulative probability distribution of the RX scores. Finally, a binary matrix is generated for each pixel which is shown in 3.2.

As discussed previously, most object detection models generate anchors by predefining an $S \times S$ grid of locations which are treated as the centers of the anchor boxes. Around these centers, bounding boxes of a predefined combination of scales and aspect ratios are generated and proposed to the head of the model which performs the classification. Using the RX algorithm with a threshold of 0.97, it is noticed that all the human bodies in the images are always detected. Along with these, other objects are detected as well. Using the binary pixel map, we only generate centers for anchor boxes around the given pixels. Furthermore, since the altitude of the drone is constant, we use only one scale value and a few aspect ratios to account for any rotations, heavily reducing the number of generated proposals. This increases the speed of the model, compensating for the slower speed of the given architecture, allowing for real time object detection.



(A) Architecture



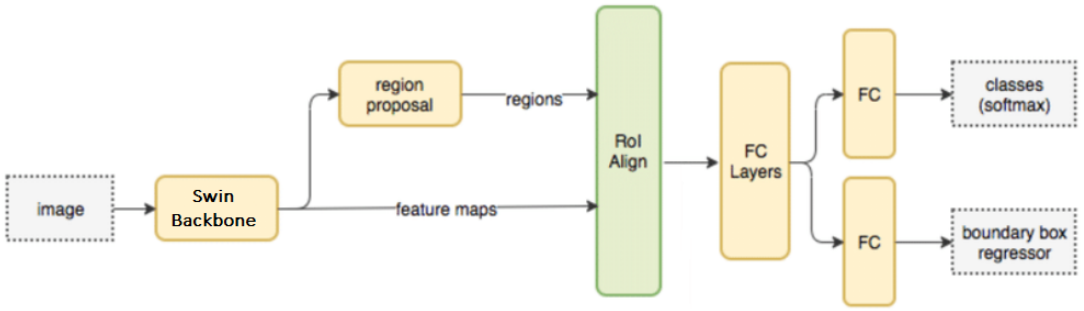
(B) Two Successive Swin Transformer Blocks)

FIGURE 3.3: The architecture of the Swin Transformer where W-MSA and SW-MSA are regular and shifted window configurations of Multi Head Attention.

3.3.2 Shifted Window backbone

As highlighted earlier, even though transformers had the capability of making long connections, the quadratic relation between image size and computation power was what stopped transformers from being utilized for CV problems. The Swin Transformer [26] successfully leverages this by computing self attention only between non overlapping, local windows. It also allows for cross window connections. Because of the hierarchical nature of this model, it can be used at varying scales. These qualities make the Swin Transformer a great general purpose backbone which can be used for any CV task.

The local windows shift after every layer, resulting in new windows and hence crossing the boundaries of the previous windows. This method is much more efficient than the sliding window technique that other transformers [11] employed. The architecture of the general purpose backbone is given in 3.3



(A) Architecture

FIGURE 3.4: Entire proposed architecture

3.3.3 General Architecture

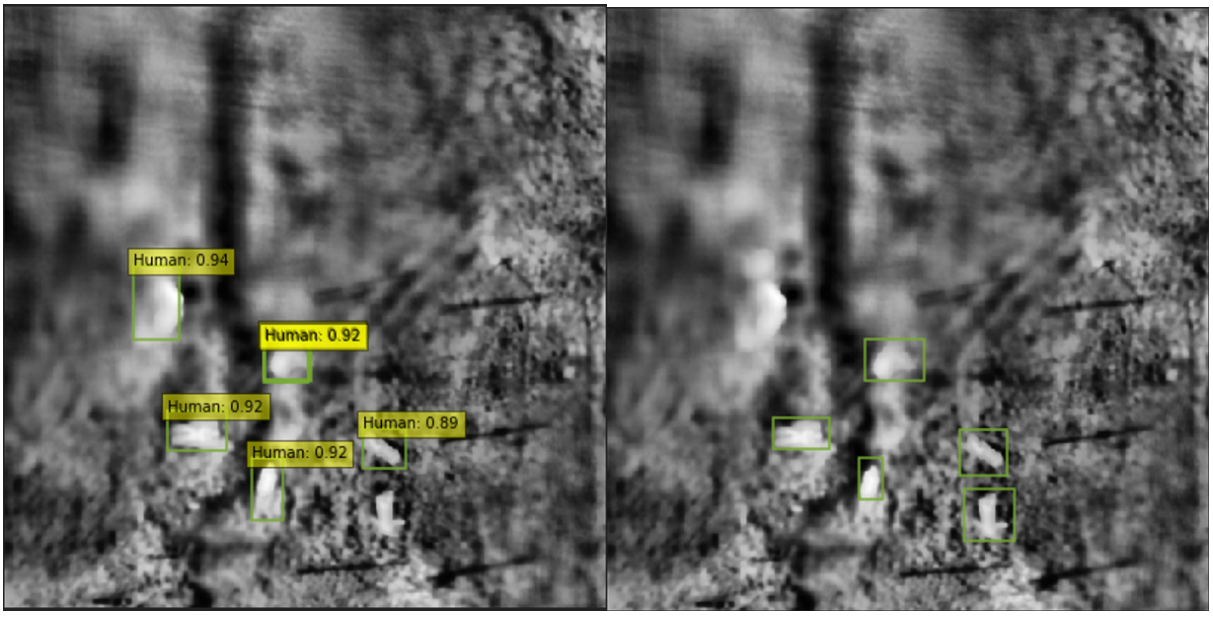
The architecture is shown in 3.4. It uses the Swin transfomer backbone to generate feature maps. These feature maps are used to generate region proposals. The region proposal step makes use of the RX detector to generate limited but useful regions. Two separate Fully Connected (FC) heads are there at the end to do the tasks of bounding box regression and the classification.

Chapter 4

Results and Discussion

4.1 Results and Discussion

As we can see in 4.1, even after finetuning the model on the train set for only 1 epoch, we see promising results. Most of the ground truths are detected with 90% confidence. We do see multiple overlapping boxes which will be removed when trained and used with the Non Max Suppression(NMS) technique. We plan to train the model using a GPU for multiple epochs and finetune the hyperparameters further accordingly. This would provide us some competitive results. The RX detection algorithm provides the initial filtering of the anchor boxes which are only generated in one scale and fixed aspect ratios. The efficiency of the Swin Backbone, due



(A) R-Swin predictions (trained for 1 epoch) (B) Ground truths for the same integral image

FIGURE 4.1: Predictions by our model vs the ground truth

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M
R-Swin(Ours)	0.565	0.900	0.630	0.421	0.615

TABLE 4.1: Results on the test sets of Modified R-CNN with pretrained weights. Note: Finetuned only for 1 epoch.

to the linear relation between image size and attention computation, is the reason for quickly generating feature maps, leading to a high FPS. Furthermore, because of the hierarchical nature of the transformer, if the scale was changing, it would have been easily dealt with. Finally, the two pass structure of the R-CNN makes it quite accurate and the drawbacks of the slow speed is dealt with the modification we made and replacing a CNN based backbone with the Swin Backbone. We expect to have even more promising results, after training the model to satisfaction on the custom dataset.

Chapter 5

Conclusion and Future Work

5.1 Conclusion and Future Work

Attention techniques are powerful and excel at finding correlations between global tokens. Overcoming the quadratic computing issue has led to a huge change in the domain of computer vision. Using shifting local windows instead of fixed patches allows a linear computing relation while also allowing relations across the boundaries of windows come through. This makes this general purpose backbone much quicker than previously proposed architectures.

The two pass method of R-CNN, which sacrifices speed for accuracy is a good compromise when used with a transformer backbone instead of the traditional CNN approach. Furthermore, while the RX detector by itself can lead to a lot of False Positives but when used as a way to initially filter out a lot of proposals, it can save a lot of computation. Lastly, transformer backed architectures are now dominating the CV domain and this trend seems to be continuing.

We get promising results with limited training as shown and we expect to outperform the previously used YOLO approaches while not trading massively on the FPS. Our future work will be to complete training and finetuning the model and see it be implemented on the drone at JKU, Linz to perform real time object detection.

Bibliography

[1] Shubhani Aggarwal and Neeraj Kumar. "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges". In: *Comput. Commun.* 149 (2020), pp. 270–299.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473 (2015).

[3] Herbert Bay et al. "Speeded-Up Robust Features (SURF)". In: *Comput. Vis. Image Underst.* 110 (2008), pp. 346–359.

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection". In: *ArXiv* abs/2004.10934 (2020).

[5] Ursula Challita, Walid Saad, and Christian Bettstetter. "Deep Reinforcement Learning for Interference-Aware Path Planning of Cellular-Connected UAVs". In: *2018 IEEE International Conference on Communications (ICC)* (2018), pp. 1–7.

[6] Vinay Chamola et al. "A Comprehensive Review of the COVID-19 Pandemic and the Role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact". In: *IEEE Access* 8 (2020), pp. 90225–90265.

[7] Wen-Chyuan Chiang et al. "Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization". In: *Applied Energy* (2019).

[8] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *EMNLP*. 2014.

[9] Nello Cristianini and John Shawe-Taylor. "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods". In: 2000.

[10] Xiyang Dai et al. "Dynamic Head: Unifying Object Detection Heads with Attentions". In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), pp. 7369–7378.

[11] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *ArXiv* abs/2010.11929 (2021).

[12] Davide Falanga et al. “The Foldable Drone: A Morphing Quadrotor That Can Squeeze and Fly”. In: *IEEE Robotics and Automation Letters* 4 (2019), pp. 209–216.

[13] Ross B. Girshick et al. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 580–587.

[14] Kaiming He et al. “Mask R-CNN”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2020), pp. 386–397.

[15] John J. Hopfield. “Neural networks and physical systems with emergent collective computational abilities.” In: *Proceedings of the National Academy of Sciences of the United States of America* 79 8 (1982), pp. 2554–8.

[16] Ron Iphofen and Mihalis Kritikos. “Regulating artificial intelligence and robotics: ethics by design in a digital society”. In: *Contemporary Social Science* 16 (2019), pp. 170 –184.

[17] Abhishek Iyer and Hari Om Bansal. “Modelling, Simulation, and Implementation of PID Controller on Quadrotors”. In: *2021 International Conference on Computer Communication and Informatics (ICCCI)* (2021), pp. 1–7.

[18] Suresh C. Kothari and Heekuck Oh. “Neural Networks for Pattern Recognition”. In: *Adv. Comput.* 37 (1993), pp. 119–166.

[19] Indrajit Kurmi, David C. Schedl, and Oliver Bimber. “A Statistical View on Synthetic Aperture Imaging for Occlusion Removal”. In: *IEEE Sensors Journal* 19 (2019), pp. 9374–9383.

[20] Indrajit Kurmi, David C. Schedl, and Oliver Bimber. “Thermal Airborne Optical Sectioning”. In: *Remote. Sens.* 11 (2019), p. 1668.

[21] Steven M. LaValle. “Rapidly-exploring random trees : a new tool for path planning”. In: *The annual research report* (1998).

[22] Yann LeCun and Yoshua Bengio. “Convolutional networks for images, speech, and time series”. In: 1998.

[23] Xiaoyun Lei, Zhian Zhang, and Peifang Dong. “Dynamic Path Planning of Unknown Environment Based on Deep Reinforcement Learning”. In: *J. Robotics* 2018 (2018), 5781591:1–5781591:10.

[24] Xinyu Li et al. “VidTr: Video Transformer Without Convolutions”. In: *ArXiv* abs/2104.11746 (2021).

[25] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *ECCV*. 2014.

[26] Ze Liu et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *ArXiv* abs/2103.14030 (2021).

[27] Ze Liu et al. “Swin Transformer V2: Scaling Up Capacity and Resolution”. In: *ArXiv* abs/2111.09883 (2021).

[28] David G. Lowe. “Object recognition from local scale-invariant features”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision* 2 (1999), 1150–1157 vol.2.

[29] Thang Luong, Hieu Pham, and Christopher D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *EMNLP*. 2015.

[30] Achleshwar Luthra et al. “Eformer: Edge Enhancement based Transformer for Medical Image Denoising”. In: *ArXiv* abs/2109.08044 (2021).

[31] Thi Thoa Mac et al. “Heuristic approaches in robot path planning: A survey”. In: *Robotics Auton. Syst.* 86 (2016), pp. 13–28.

[32] Willian F. Moreno et al. “Electromagnetic Sensor Onboard Drones for the Detectin of Land Mines”. In: 2018.

[33] Daniel Neimark et al. “Video Transformer Network”. In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (2021), pp. 3156–3165.

[34] Alena Otto et al. “Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey”. In: *Networks* 72 (2018), pp. 411–458.

[35] Javier Panadero et al. “Maximising reward from a team of surveillance drones: a simheuristic approach to the stochastic team orienteering problem”. In: *European Journal of Industrial Engineering* 14 (2020), p. 485.

[36] Stephen M. Pizer et al. “Adaptive histogram equalization and its variations”. In: *Graphical Models graphical Models and Image Processing computer Vision, Graphics, and Image Processing* 39 (1987), pp. 355–368.

[37] Riccardo Poli, James Kennedy, and Tim M. Blackwell. “Particle swarm optimization”. In: *Swarm Intelligence* 1 (2007), pp. 33–57.

[38] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *ArXiv* abs/1804.02767 (2018).

[39] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 779–788.

[40] Irving S. Reed and Xiaoli Yu. “Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution”. In: *IEEE Trans. Acoust. Speech Signal Process.* 38 (1990), pp. 1760–1770.

[41] Shaoqing Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2015), pp. 1137–1149.

[42] Arthur G. Richards et al. “COORDINATION AND CONTROL OF MULTIPLE UAVs”. In: 2002.

[43] Christopher Dahlin Rodin et al. "Object Classification in Thermal Images using Convolutional Neural Networks for Search and Rescue Missions with Unmanned Aerial Systems". In: *2018 International Joint Conference on Neural Networks (IJCNN)* (2018), pp. 1–8.

[44] Kristin Bergtora Sandvik. "The Political and Moral Economies of Dual Technology Transfers: Arming Police Drones". In: 2016.

[45] Andrey V. Savkin and Hailong Huang. "Navigation of a Network of Aerial Drones for Monitoring a Frontier of a Moving Environmental Disaster Area". In: *IEEE Systems Journal* 14 (2020), pp. 4746–4749.

[46] David C. Schedl, Indrajit Kurmi, and Oliver Bimber. "An autonomous drone for search and rescue in forests using airborne optical sectioning". In: *Science Robotics* 6 (2021).

[47] David C. Schedl, Indrajit Kurmi, and Oliver Bimber. "Search and Rescue with Airborne Optical Sectioning". In: *Nat. Mach. Intell.* 2 (2020), pp. 783–790.

[48] Johannes L. Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 4104–4113.

[49] Yuhui Shi and Russell C. Eberhart. "Empirical study of particle swarm optimization". In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)* 3 (1999), 1945–1950 Vol. 3.

[50] Amr Suleiman et al. "Navion: A 2-mW Fully Integrated Real-Time Visual-Inertial Odometry Accelerator for Autonomous Navigation of Nano Drones". In: *IEEE Journal of Solid-State Circuits* 54 (2019), pp. 1106–1119.

[51] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. "Sequence to Sequence Learning with Neural Networks". In: *NIPS*. 2014.

[52] Maryam Torabbeigi, Gino J. Lim, and Seon Jin Kim. "Drone Delivery Scheduling Optimization Considering Payload-induced Battery Consumption Rates". In: *Journal of Intelligent & Robotic Systems* 97 (2020), pp. 471–487.

[53] Jasper R. R. Uijlings et al. "Selective Search for Object Recognition". In: *International Journal of Computer Vision* 104 (2013), pp. 154–171.

[54] Prahlad Vadakkepat, Kay Chen Tan, and Wang Ming-Liang. "Evolutionary artificial potential fields and their application in real time robot path planning". In: *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)* 1 (2000), 256–263 vol.1.

[55] Ashish Vaswani et al. "Attention is All you Need". In: *ArXiv* abs/1706.03762 (2017).

[56] Dan Wang et al. "Multi-view 3D Reconstruction with Transformer". In: *ArXiv* abs/2103.12957 (2021).

[57] Lukas Wawrla, Omid Maghazei, and Torbjørn H. Netland. "Applications of drones in warehouse operations". In: 2019.

[58] Daniel Weimer et al. "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection". In: *Cirp Annals-manufacturing Technology* 65 (2016), pp. 417–420.