Camouflaged Object Detection Using Deep Learning

Khalak Bin Khair, Saqib Jahir, Mohammed Ibrahim, Fahad Bin and Debajyoti Karmaker

***Abstract*- Object detection is a computer technology that deals with finding instances of semantic items of a specific class in digital photos and videos. It is connected to computer vision and image processing. On top of object detection, we detect camouflage objects within an image using Deep Learning techniques. Deep learning is a subset of machine learning that is essentially a three-layer neural network. Over 6500 images which possess camouflage properties are gathered from various internet sources and divided into 4 categories to compare the result. Those images are labelled and then trained and tested using vgg16 architecture on the jupyter notebook using the TensorFlow platform. The architecture is further customized using Transfer Learning. Methods for transferring information from one or more of these source tasks to increase learning in a related target task are created through transfer learning. The purpose of this transfer of learning methodologies is to aid in the evolution of machine learning to the point where it is as efficient as human learning. After training the model using all the techniques and customization mentioned and described above, At last the architecture gives us outstanding accuracy.**

***Keywords*: *Deep Learning, Transfer Learning, TensorFlow, Camouflage, Object Detection, Architecture, Accuracy, Model, VGG16*.**

I. Introduction

1.1 Overview

Object detection is a computer vision approach for detecting things in photos and videos. To obtain relevant results, object detection algorithms often use machine learning or deep learning. We can recognize and locate objects of interest in photos or video in a handful of seconds when we glance at them. The purpose of object detection is to use a computer to imitate this intelligence. But what is the purpose of object detection? Object detection's main purpose is to identify and find one or more effective targets in still or video data. It covers a wide range of techniques, including image processing, pattern recognition, artificial intelligence, and machine learning. Traditionally, object detection has been accomplished by manually extracting feature models, with popular features represented by HOG (histogram of oriented gradient), SIFT (scale-invariant feature transform), Haar (Haar-like features), and other grayscale-based approaches. As you can see in this picture [Figure 1.1], with the help of object detection we are able to spot the military soldiers and differentiate them from the tanks.

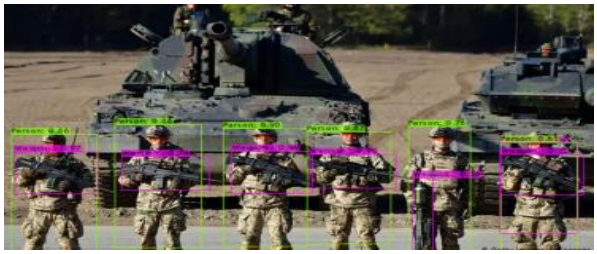

Figure 1.1: Military Object Detection

Object detection can be done in two methods, machine-learning approach or deep learning approach. We discussed in brief some of the machine-learning methods in the papers we reviewed, like Haar, HOG and so on. For the sake of this research, we'll be focusing on deep learning methods because they've become the state-of-the-art approaches to object detection. We are going to propose Camouflage Detection System using Deep Learning. Our purpose is to create a robust model that will be able to detect the camouflaged objects.

1.2 Background

Handcrafted characteristics were used to create the majority of the early object detection algorithms. People had no choice but to build complicated feature representations and a range of speed up skills to exhaust the use of limited computing resources due to the lack of effective image representation at the time.

• Viola Jones Detectors

P. Viola and M. Jones performed real-time face detection without any limitations for the first time 18 years ago [1], [2]. The detector, which ran on a 700MHz Pentium III CPU, was tens or even hundreds of times faster than existing methods at the time with comparable detection accuracy. The authors' names were given to the detection method, which was eventually known to as the "Viola-Jones (VJ) detector," in honor of their substantial contributions. The VJ detector uses a simple method of detection called sliding windows, which involves going through all potential locations and scales in a picture to see if any of the windows include a human face [Figure 1.2]. Although it appears to be a straightforward process, the

calculations required were well beyond the capabilities of the computer at the time.

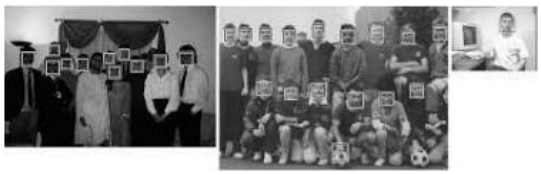

Figure 1.2: Viola Jones Detectors

• HOG Detector

N. Dalal and B. Triggs proposed the Histogram of Oriented Gradients (HOG) [3] feature descriptor in 2005. HOG is regarded as a significant advancement in the scale-invariant feature transform and shape contexts of the time. The HOG descriptor is designed to be computed on a dense grid of uniformly spaced cells and use overlapping local contrast normalization (on ”blocks”) for improving accuracy while balancing feature invariance

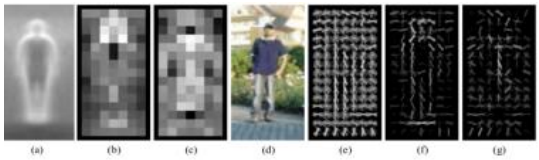

Figure 1.3: Human detector using HOG

(Including translation, scale, illumination, and so on) and nonlinearity (on discriminating different object categories) [Figure:1.3].

•Deformable Part-based Model (DPM)

DPM was the pinnacle of traditional object detection systems, having won the VOC-07, -08, and -09 detection competitions. P. Felzenszwalb [4] suggested DPM in 2008 as an extension of the HOG detector, and R. Girshick improved it significantly. The DPM uses the ”divide and conquer” detection philosophy, in which the training can simply be thought of as learning a right way to decompose an object, and the inference can be thought of as an ensemble of detection on distinct object portions.

1.3 Problem Statement

Object detection of camouflage objects can be quite complex using machine learning techniques. It can be quite difficult to spot the camouflage object. For example, the pygmy seahorse is a kind of fish which is only 1cm, and its body is natural camo for under the sea objects, so detecting them can be quite difficult. The same goes with military, most military uniforms are natural camos so they can blend with the natural surroundings at war easily. To detect them can be quite complex. Therefore, building a robust model for object detection of camouflaged objects using deep learning helps us in detecting these complex objects with a lot more ease. We are providing solutions where the users can easily understand the objects and recognize them amidst their camo background.

1.4 Research Question

• How is Deep Learning technique more effective than Machine Learning technique when it comes to Object Detection?

When the data amount is high, Deep Learning outperforms conventional techniques. Traditional Machine Learning techniques, on the other hand, are preferable when dealing with tiny amounts of data. Traditional machine learning techniques, such as linear regression or a decision tree, have a relatively straightforward structure, but deep learning is based on an artificial neural network. This multi-layered ANN is complicated and interwoven, just like a human brain.

1.5 Objective

General Objective

In this study we are going to develop a camouflage object detection system. This system is will aim to detect the object that are camouflaged with the nature. This system will help us to even detect small animals such as pygmy seahorse with ease.

Specific Objective

• To design a robust camouflage object detection system using Deep Learning

• To develop a camouflage object detection system using TensorFlow platform.

• To find out the effectiveness of this architecture for object detection

Our research is based on object detection using deep learning and our aim is to make a robust object detection so that detecting the camouflage images becomes slightly easier and make a positive impact in our human life. The proposed Object Detection system will help us yield a result with high accuracy. Working with TensorFlow helps us perform a variety of tasks related to deep neural network training and inference.

II. Literature review

This research is based on research publications from a variety of fields, we go through the literature review of papers based on object detection, deep learning, CNN architectures, Anchor-based method object detection.

What exactly do we mean by the term object detection?

Detecting instances of visual objects of a specific class (such as persons, animals, or cars) in digital photographs is an important computer vision task. Object detection's goal is to create computational models and approaches that give one of the most fundamental bits of information required by computer vision applications. Object detection is one of the most fundamental problems in computer vision, and it provides the foundation for many other computer vision tasks such as instance segmentation, image captioning, object tracking, and so on. Object detection can be divided into two research topics:" general object detection" and" detection applications," with the former aiming to investigate methods for detecting various types of objects in a unified framework to simulate human vision and cognition, and the latter referring to detection in specific application scenarios such as pedestrian detection, face detection, text detection, and so on. In recent years, the rapid development of deep learning techniques [5] has brought new blood into object detection, leading to remarkable breakthroughs and pushing it forward to research hot-spot with unprecedented attention [6]. Object detection is currently widely employed in a wide range of real-world applications, including autonomous driving, robot vision, video surveillance, and so on.

Constantine Papageorgiu et al. [7] carried out a research that built a robust trainable object detection system that is used in surveillance applications, driver assistance systems and as front ends to recognition systems. Their representation is capable of capturing the structure of the object class we want to detect while ignoring the noise in the photos. Image reconstruction techniques prompted the usage of an overcomplete dictionary; our goal is classification, and the overcomplete dictionary offers us with a richer expressive language in which we may compare complicated patterns. An example-based learning strategy is used, in which a model of an object class is inferred implicitly from a set of training instances. Specializing this generic system to a specific domain in this way entails just plugging in a new batch of training data rather than changing the basic system or creating a new model by hand. The specific learning engine used is a support vector machine (SVM) classifier. This classification technique has a number of properties that make it particularly attractive and has recently received much attention in the machine learning community [7]. The system uses 1) a series of photos of the object class that have been aligned and scaled so that they are all in roughly the same position and size, and 2) a set of patterns that are not in our object class as input in the training step. In the testing phase, detecting objects in out-of-sample images is the task. The system slides a fixed size window over an image and uses the trained classifier to decide which patterns show the objects of interest [7]. Wavelets provide a natural mathematical structure for describing the patters in a more detailed manner. The Haar wavelet is perhaps the most basic finite support feature. We convert our photographs from pixel space to a three-dimensional space wavelet coefficient, yielding an overcomplete dictionary of features that can subsequently be utilized to train a machine learning algorithm. The Haar transform creates a multiresolution image by using wavelet features at different sizes to capture varying levels of information; coarse scale wavelets encode vast regions, while fine scale wavelets describe smaller, localized areas. A wavelet's strong response shows the presence of an intensity differential, or border, at that spot in the image, whereas a wavelet's weak response suggests a uniform area. The fact that wavelets capture visually realistic characteristics of the shape and interior structure of things that are invariant to specific changes is our primary motivation for employing them. As a result, diverse sample photos from the same object class map to similar feature vectors, resulting in a compact representation. A weak coefficient in a relatively dark image may still indicate the presence of an intensity difference that is significant for classification purposes, a weak coefficient in a relatively dark image may still indicate the presence of an intensity difference that is significant for classification purposes. The authors normalized a coefficient's value against the other coefficients in the same area to lessen these effects on the features utilized for categorization. Ensemble average values more than 1 show strong intensity difference features that are consistent across all examples, values less than 1 indicate consistent uniform regions, and values near 1 indicate inconsistent features or random patterns.

The term salient means most noticeable or important. Salient object identification is the process of detecting and segmenting salient objects in natural settings. In 2019, Ali Borji et al. [8] described salient object segmentation in computer vision as 1) detecting the most salient object and 2) segmenting the accurate region of that object. [8] A model should meet at least three of the following criteria for good saliency detection: 1) accurate detection: the likelihood of missing real salient regions and incorrectly marking the background as a salient region should be low; 2) high resolution: saliency maps should have high or full resolution to accurately locate salient objects while retaining original image information; and 3) computational efficiency: these models should detect salient regions quickly as front-ends to other complex processes. In recent years, with the rise in popularity of convolutional neural networks (CNNs) and in particular with the development of fully convolutional neural networks, a third wave of interest has recently emerged. Many researchers have used CNN-based approaches because they minimize the need for hand-crafted features and reduce the reliance on center bias information. Neurons with broad receptive fields provide global information that can aid in identifying

the most salient region in an image, whereas neurons with narrow receptive fields provide local information that can be used to refine saliency maps generated by the upper layers. Using CNN based models helps in refining the boundaries thus highlighting the salient regions.

In an influential study, Achanta et al. [9] adopt a frequency-tuned approach to compute full resolution saliency maps. The saliency of pixel x is computed as $s(x) = ||I\mu - I\omega hc(x)||^2$where Iμ is the mean pixel value of the image (e.g., RGB/Lab features) and Iωhc is a Gaussian blurred version of the input image (e.g., using a 5 × 5 kernel)*. Detection of salient object based on pixels or patches like the formula above suffer a couple of shortcomings: 1) high-contrast edges usually stand out instead of the salient object, and 2) the boundary of the salient object is not preserved well. To overcome this issue, the number of regions is far smaller than the number of blocks, allowing for the development of extremely efficient and quick algorithms. This above method uses heuristics to detect salient objects. While hand-crafted features enable real-time detection, they have a number of drawbacks that limit their capacity to capture important items in difficult situations. CNNs have recently been found to be quite effective when used for salient object detection. CNNs can accurately capture the most important regions without any prior information thanks to their multilevel and multi-scale properties.

Deep learning-based salient object identification algorithms can be divided into two groups. Models that use multilayer perceptron's (MLPs) for saliency detection fall into the first category. The input image is generally over segmented into single- or multiscale tiny regions in these models. Then, using a CNN, high-level features are extracted, which are then input into an MLP to compute the saliency value of each small region. Kaiwen Duan et al. [10] carried out research called Centre-Net which identifies each item as a triplet of key points rather than a pair, improving precision and recall. In this paper, the researchers create two unique modules, cascade corner pooling and center pooling, that enrich data acquired by both the top-left and bottom-right corners while also providing more recognized data from the center regions. The anchor-based flowchart, one of the most common flowcharts today, sets a series of rectangles with predetermined sizes (anchors) on a picture then regresses the anchors to the desired location using ground- truth objects. The drawbacks that arise from anchor-based approaches are tackled by an object-detection pipeline named Corner-Net. However, CornerNet's performance is constrained by its limited capacity to refer to an object's global information. In order to address this problem, CentreNet a low-cost yet successful approach that explores the central part of a proposal, that is, the region near the geometric center of a box, with one additional keypoint is introduced. We reason that if a predicted bounding box has a high IoU with the ground-truth box, the center keypoint in the central region of the bounding box is likely to be predicted as the same class, and vice versa. Instead of using a pair, a triplet of keypoints is used to represent each object. To improve the detection of centre keypoints, 2 strategies are used, center pooling and cascade pooling. To improve the performance of centre key points and corners, 2 strategies are introduced.

First strategy is center pooling, which is used in the branch for predicting center keypoints. Center pooling helps the center keypoints obtain more recognizable visual patterns within objects, which makes it easier to perceive the central part of a proposal. The second strategy is cascade corner pooling, which equips the original corner pooling module [11] with the ability to perceive internal information. They achieve this by obtaining the maximum summed response in both the boundary and internal directions of objects on a feature map for corner prediction [10].

CenterNet, which combines center pooling and cascade corner pooling, has an AP of 47.0 percent, outperforming all existing onestage detectors by a wide amount.

CornerNet is used as a baseline in this study. CornerNet creates two heatmaps for detecting corners: one for the top-left corners and one for the bottom-right corners. The heatmaps show the locations of keypoints in many categories and give each one a confidence level. CornerNet also predicts the embedding of each corner as well as a collection of offsets. The embeddings are used to determine if two corners belong to the same item. The offsets learn to remap the heatmap corners to the supplied image. To detect the regions inside the bounding box, unlike CornetNet, the researchers use CentreNet. CentreNet method uses a triplet of keypoints rather than a pair for object detection. This method retains a one-stage detector while partially inheriting the capabilities of RoI pooling. Their method just takes into account the most important data, and it comes at a low cost. On the basis of CornerNet, they integrate a heatmap for the center keypoints and forecast the offsets of the center keypoints. The detection results are influenced by the size of the central region in the bounding box. For small bounding boxes, for example, small center regions result in a low recall rate, whereas large central regions result in low precision. As a result, a scale-aware center region that can adapt to different bounding box sizes was proposed. For a small bounding box, the scaleaware central region tends to generate a relatively large central region, and for a large bounding box, it tends to generate a relatively tiny central region.

Miang Liang et al. [12] conducted research on multi-task muti-sensor fusion for 3D Object Detection. Most self-driving cars rely on three-dimensional perception

because it allows for interpretable motion planning in a bird's eye view. In this study the authors claim that by solving numerous perception tasks at the same time, we can develop richer feature representations and hence improve detection performance. A multi-sensor detector to achieve this goal is built, which considers 2D and 3D object identification, ground estimation, and depth completion. Importantly, our model can be taught from start to finish and accomplishes all of these functions at the same time. On the KITTI object detection benchmark [13] as well as the more difficult TOR4D object detection benchmark [14], the usefulness of the approach is demonstrated. The authors exhibit considerable performance improvements over earlier state-of-the-art approaches in 2D, 3D, and BEV detection tasks on the KITTI benchmark. Meanwhile, the proposed detector operates at a rate of over 10 frames per second, making it a viable real-time option. To combine several sensors, F-PointNet [15] employs a cascade technique. Specifically, 2D object detection is performed on images initially, followed by the generation of 3D frustums by projecting 2D detections to 3D, and the use of PointNet [16] [17] to regress the 3D location and shape of the bounding box. Each stage, which is still using a single sensor, is constrained by the overall performance in this framework. Perceiving the scene in real time is one of the most important jobs in autonomous driving.

A multi-task multi-sensor fusion model for 3D object detection is proposed in this paper. The following are some of the key aspects of our strategy. First, they create a multisensor architecture that incorporates feature fusion on a point-by-point and ROI-by-ROI basis. Second, the geometry of the road is considered by our integrated ground estimating module. Third, we use the depth completion task to improve multi-sensor feature learning and achieve dense point-wise feature fusion. A LiDAR point cloud and an RGB picture are fed into our multi-sensor detector. The backbone network is divided into two streams, one extracting image feature maps and the other extracting LiDAR BEV feature maps. To fuse multiscale picture features to the BEV stream, point-wise feature fusion is used. The final BEV feature map uses 2D convolution to anticipate dense 3D detections per BEV voxel. Network architecture includes Point-wise fusion, ROI-wise feature fusion. Point-wise feature fusion: The authors combine the convolutional feature maps of LiDAR and picture streams using point-wise feature fusion. To supplement BEV features with the information richness of image features, the fusion is routed from image steam to LiDAR steam. With a feature pyramid network, they collect multi-scale features from all four blocks in the picture backbone network. The multi-scale image feature map that results is then fused to each LiDAR BEV backbone network block.

ROI-wise feature fusion: The goal of ROI-wise feature fusion is to improve the precision of high-quality detections' localisation in 2D and 3D spaces, respectively. To do this, the ROI feature extraction must be precise in order to accurately anticipate relative box refinement. The authors get an axis-aligned image ROI and an orientated BEV ROI by projecting a 3D detection onto the image and BEV feature maps. To extract features from an axis-aligned image ROI, they use ROIAlign [18] Tao Kong et al. [19] proposed a method FoveaBox, an object detection framework that is accurate, adaptable, and fully anchor-free. While practically all modern object detectors use predetermined anchors to list possible positions, sizes, and aspect ratios for object search, their performance and generalization capabilities are also constrained by anchor design. Instead of using an anchor reference, FoveaBox learns the object's existing potential and bounding box coordinates immediately. This is accomplished by (a) predicting category-sensitive semantic maps for the item's existing possibility, and (b) generating a category-agnostic bounding box for each position that might contain an object. To improve the model's accuracy, an instance is assigned to adjacent feature levels in FoveaBox. On standard benchmarks, we demonstrate its efficacy and present thorough experimental analyses. FoveaBox achieves state-of-the-art single model performance on the standard COCO and Pascal VOC object detection benchmarks without any bells and whistles. More crucially, FoveaBox eliminates all anchor box computation and hyper-parameters, which are often sensitive to final detection performance.

In object identification frameworks, anchors must be constructed and handled with care. (a) The density with which anchors cover the instance location space is one of the most critical factors in anchor design. Anchors are meticulously designed depending on statistics computed from the training/validation set in order to attain a high recall rate. (b) A design decision based on a specific dataset may or may not be applicable to other applications, reducing the generality. (c) Anchor-methods use intersection-overunion (IoU) to determine positive/negative samples during the training phase, which adds extra computation and hyper-parameters to an object detection system.

FoveaBox is a straightforward concept: it consists of a backbone network and a fovea head network. The backbone, which is an off-the-shelf convolutional network, is responsible for calculating a convolutional feature map over a full input image. The fovea head is divided into two sections: the first performs per-pixel classification on the backbone's output, and the second performs box prediction for each place that may be covered by an object.

For each position potential contained by an instance, FoveaBox forecasts the object existence possibility and the accompanying boundary.

$$x_1^{'} = \frac{x_1}{s_1}, \quad y_1^{'} = \frac{y_1}{s_1}, \quad x_2^{'} = \frac{x_2}{s_1}, \quad y_2^{'} = \frac{y_2}{s_1},$$
$$c_x^{'} = 0.5(x_2^{'} + x_1^{'}), \quad c_y^{'} = 0.5(y_2^{'} + y_1^{'}),$$
$$\omega^{'} = x_2^{'} - x_1^{'}, \quad h^{'} = y_2^{'} - y_1^{'}$$

We'll go over the major components one by one in this section. 1) Object Occurrence Possibility: Given a valid ground-truth box denoted as (x1, y1, x2, y2). We first map the box into the target feature pyramid P1 where sl is the down-sample factor. The positive area Rpos on the score map is designed to be roughly a shrunk version of the original one. At training phase, each cell inside the positive area is annotated with the corresponding target class label. The negative area is the whole feature map excluding area in Rpos [19].

2) Scale Assignment: based on the number of feature pyramidal levels, divide the scales of objects into several bins. On pyramid levels P3 through P7, each pyramid has a basic scale rl that ranges from 32 to 512.

3) Box Prediction: Each ground-truth bounding box is stated as G = (x1, y1, x2, y2).

$$t_{x_1} = log\frac{s_1(x+0.5) - x_1}{r_1},$$
$$t_{y_1} = log\frac{s_1(y+0.5) - y_1}{r_1},$$
$$t_{x_2} = log\frac{x_2 - s_1(x+0.5)}{r_1},$$
$$t_{y_2} = log\frac{y_2 - s_1(x+0.5)}{r_1},$$

FoveaBox computes the normalized offset between (x, y) and four boundaries straight from a positive point (x, y) in Rpos.

Yanghao Li et al. [20] Proposed a research paper to challenge the scale variation in object detection, called TridenNet. Trident Network (TridentNet) is a project aimed at creating scale-specific feature maps with consistent representational power. The authors create a parallel multi-branch architecture in which each branch is independent of the others, has the same transformation parameters as the other, but has a different receptive field. The handling of size variation is a central challenge for both systems. Detectors, especially those that are extremely small or very huge, are hampered by the scales of object instances, which can range from small to very enormous. An intuitive option to address big scale variance is to use multi-scale image pyramids, which are common in both hand-crafted feature-based methods and contemporary deep CNN-based methods. To solve the problem of scale variation in object detection. TridentNet was able to build scale-specific feature maps with a uniform representational power thanks to its multibranch structure and scale-aware training. Through their weight-sharing trident-block design, they offer TridentNet Fast, a fast approximation with only one major branch, introducing no new parameters or computational cost during inference.

With comprehensive ablation studies, they were able to validate the effectiveness of our technique on the standard COCO benchmark. Using a single model with a ResNet-101 backbone, the suggested method obtains a mAP of 48.4 when compared to state-of-theart methods.

Although it has been widely assumed for years that modeling relationships between objects will aid object recognition, there has been little evidence that this is the case in the deep learning era. To tackle this challenge, Han Hu et al. [21] Put forward a proposal that detect systems individually without exploiting ther relations during learning. This paper proposes a module for object relationships. It processes a group of objects at the same time by interacting with their appearance features and geometry, allowing for the modeling of their relationships. It's light and stays in place. It doesn't need any extra supervision and is simple to integrate into existing networks. In the contemporary object detection pipeline, it has been demonstrated to improve object recognition and duplicate removal processes. It proves that modeling object relations in CNN-based detection is effective. It is the first object detector that is truly end-to-end. There has been no substantial development in utilizing object relations for detection learning during the deep learning period. The majority of approaches still focus on recognizing items separately. In the contemporary object detection pipeline, it has been demonstrated to improve object recognition and duplicate removal processes. It proves that modeling object relations in CNN-based detection is effective. It is the first object detector that is truly end-to-end. There has been no substantial development in utilizing object relations for detection learning during the deep learning period. The majority of approaches still focus on recognizing items separately

Relation for Duplicate Removal:

The process of removing duplicates inherently necessitates the use of object relationships. A basic example of the heuristic NMS approach is that the object with the greatest score will eliminate its neighboring neighbors with lower scores.

End-to-End Object Detection:

The area proposal loss, instance recognition loss, and duplication classification loss are all combined with equal weights in the end-to-end training.

Mingxing Tan et al. [22] extensively examine neural network architecture design choices for object detection and offer some major enhancements to increase efficiency. First, they propose a weighted bi-directional feature pyramid network (BiFPN), which enables simple and quick multi-scale feature fusion; second, propose a compound scaling method that scales the resolution, depth, and width of all backbone, feature network, and box/class prediction networks at the same time. They built a new family of object detectors called EfficientDet based on these optimizations and EfficientNet backbones, which consistently achieve considerably superior efficiency than prior art across a wide range of resource limitations. EfficientDet's overall architecture, which is mostly based on the one-stage detector paradigm. As the backbone network, the authors use ImageNettrained EfficientNets. The feature network in our proposal is the BiFPN, which takes

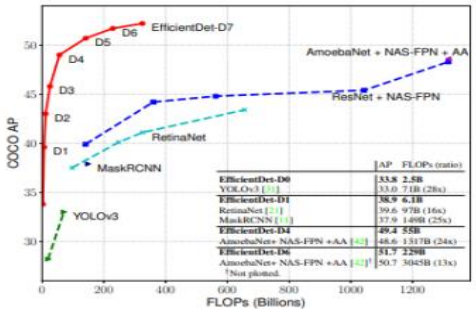

Figure 2.1: The EfficientDet achieves new state-of-the-art 52.2% COCO AP with much fewer parameters and FLOPs than previous detectors. More studies on different backbones and FPN/NAS-FPN/BiFPN.

level 3-7 features from the backbone network and applies top-down and bottom-up bidirectional feature fusion periodically. To provide object class and bounding box predictions, these fused characteristics are fed into a class and box network.

Compound Scaling:

The authors of this paper created a family of models that can fulfill a wide range of resource limitations, with the goal of optimizing both accuracy and efficiency. The ability to scale up a baseline EfficientDet model is a crucial hurdle here. They offered a new compound object recognition approach that leverages a simple compound coefficient to scale up all dimensions of the backbone network, BiFPN network, class/box network, and resolution at the same time.

Input Image Resolution:

Because BiFPN uses feature levels 3–7, the input resolution must be 2 time 7. As a result, the equation to linearly increase resolutions. Rinput = 512 + φ · 128 The EfficientDet is evaluated on COCO2017 detection datasets with 118k training images, the model is further optimized using an SGD optimizer.

Xuebin Qin et al. [23] proposed a boundary-aware salient object detection, BASNet, The architecture is made up of a highly supervised Encoder-Decoder network and a residual refinement module, which are in charge of saliency map refining and prediction, respectively. By combining Binary Cross Entropy (BCE), Structural SIMilarity (SSIM), and Intersectionover-Union (IoU) losses, the hybrid loss guides the network to learn the transformation between the input image and the ground truth in a three-level hierarchy – pixel, patch, and map level. The suggested predict-refine architecture is able to efficiently segment the salient object regions and reliably forecast the fine structures with unambiguous bounds thanks to the hybrid loss. It delivers high-quality bounds and accurate salient object segmentation (i) A new predict-refine network is presented to capture both global (coarse) and local (fine) circumstances. It combines an unique residual refinement module with a UNet-like deeply supervised Encoder-Decoder network. The Encoder-Decoder network converts the input image to a probability map, and the refinement module fine-tunes the predicted map by learning the residuals between the coarse saliency map and the ground truth. (ii) To obtain a high-confidence saliency map with a clear boundary, we suggest a hybrid loss that incorporates Binary Cross Entropy, Structural SIMilarity (SSIM), and IoU losses, all of which are expected to learn from ground truth information at the pixel, patch, and map levels, respectively.

Refine Model:

Refinement Module (RM) [24] [25] is commonly constructed as a residual block that refines the coarse saliency maps that have been forecasted. Sref ined = Scoarse + Sresidual. To define the term "coarse" before the authors talk about their refinement module. The term "coarse" has two meanings in this context. The first is the hazy and noisy limits. The other is the regional probabilities that are unevenly forecasted. SOD, ECSSD, DUT-OMRON, PASCAL-S, HKU-IS, and DUTS are six commonly used benchmark datasets on which the authors tested their technique. SOD includes 300 photos that were created with image segmentation in mind. These photographs are difficult to work with because most of them have many important items that are either low contrast or overlap with the image boundaries.he DUTS-TR dataset, which contains 10553 pictures, was used to train the network. The dataset is augmented by horizontal flipping to 21106 photos before training. Each image is scaled to 256256 pixels and then randomly cropped to 224224 pixels during training. The ResNet-34 model [18] is used to initialize some of the encoder parameters. Xavier [26] initializes the other convolutional layers. To train their network, they used the Adam optimizer, with the hyper parameters set to

default.Alex Bochkoskiy et al. [27] proposed a research YOLOv4 for optimal speed and accuracy of object detection. Rather than the theoretical indicator of low computing volume, the major purpose of this work is to create a high operating speed of an object detector in production systems and optimization for parallel computations (BFLOP). The authors created a product that will be simple to train and operate. Anyone who trains and tests with a traditional GPU, for example, may get real-time, high-quality, and convincing object detection results, as the YOLOv4 does.

1. A model for detecting objects that is both efficient and powerful. It allows anyone to train a highly fast and accurate object detector using a 1080 Ti or 2080 Ti GPU.

2. During the detector training, the influence of state-of-the-art Bag-of-Freebies and Bag-of-Specials item detection approaches.

3. They modify state-of-the-art methods and make them more efficient and suitable for single GPU training, including CBN [28], PAN [29] etc.

Bag of freebies A traditional object detector is usually trained offline. As a result, researchers are continually looking for ways to improve training methods so that the object detector can receive improved accuracy without increasing the inference cost. We refer to these strategies as ”bag of freebies” because they just vary the training plan or enhance the training expense. Data augmentation is a technique used by object detection techniques that fits the concept of a ”bag of freebies.” The goal of data augmentation is to increase the heterogeneity of the input photos so that the built object detection model can handle photographs from a variety of contexts. Photometric distortions and geometric distortions, for example, are two prominent data augmentation methods that clearly improve the object detection process. We alter the brightness, contrast, color, saturation, and noise of an image when dealing with photometric distortion. We use random scaling, cropping, flipping, and rotating to create geometric distortion. Bag of specials

“Bag of specialties” refers to plugin modules and post-processing procedures that only increase the inference cost by a little amount while greatly improving object detection accuracy. Of general, these plugin modules are used to improve particular qualities in a model, such as increasing the receptive field, adding an attention mechanism, or improving feature integration capability, while post-processing is a way for screening model prediction outcomes.

Rather than the theoretical signal of low computing volume, the primary goal of this architecture is rapid neural network operating speed in production systems and optimization for parallel computations (BFLOP). They provide two real-time neural network options:

• In convolutional layers, they use a modest number of groups (1-8) for GPU: CSPDarknet53 / CSPResNeXt50

• They employ grouped-convolution for VPU, but we don’t use Squeeze-and-excitement (SE) blocks - precisely, this applies to the following models:

MixNet [22] / GhostNet [30] / MobileNetV3 / EfficientNet-lite.

For improving the object detection training, a CNN usually uses the following:

• Activations: ReLU, leaky-ReLU, parametric-ReLU, ReLU6, SELU, Swish, or Mish

• Bounding box regression loss: MSE, IoU, GIoU, CIoU, DIoU

• Data augmentation: CutOut, MixUp, CutMix

• Regularization method: DropOut, DropPath , Spatial DropOut , or DropBlock

• Normalization of the network activations by their mean and variance: Batch Normalization (BN), Cross-GPU Batch Normalization (CGBN or SyncBN) , Filter Response Normalization (FRN), or Cross-Iteration Batch Normalization (CBN)

• Skip-connections: Residual connections, Weighted residual connections, Multi-input weighted residual connections, or Cross stage partial connections (CSP) [31]

Jia-Xing Zhao et al. [32] proposed an edge guidance network for salient object detection EGNet, The goal of salient object detection (SOD) is to find the things in a scene that are the most easily recognized. It’s been widely employed in vision and image processing applications such as content-aware picture editing, object recognition, photosynth, non-photo-realist rendering, poorly supervised semantic segmentation, and image retrieval. There have also been a number of studies published on video salient object detection and RGB-D salient object detection. The bulk of SOD techniques based on CNN architecture that take picture patches as input use multi-scale or multi-context information to generate the final saliency map. To preserve the salient object boundaries, the authors propose an EGNet that explicitly models complimentary prominent object information and salient edge information within the network. The conspicuous edge traits, on the other hand, are also useful for localisation. Their model optimizes these two complementing tasks in tandem by allowing them to mutually assist one another, resulting in much improved predicted saliency maps. On six frequently used datasets, we compare the suggested methods to 15 state-of-the-art methodologies. Our strategy outperforms the competition in three evaluation metrics even without the bells and whistles. When compared to region-based algorithms, pixel-wise

salient object recognition methods have demonstrated to be superior. However, they overlooked the images' spatial coherence, resulting in unsatisfactory salient object borders. In both segmentation and localization, a good salient edge detection result can aid the salient object detection task, and vice versa. The researchers developed an EGNet to describe and fuse the complementary salient edge and salient object information within a single network in an end-to-end way based on this concept.

In their strategy, they will still have five side paths: S (2), S(3), S(4), S(5), and S(6). These five characteristics could be denoted by a backbone features set C: C = C (2), C(3), C(4), C(5), C(6).

where C (2) denotes the Conv2-2 features and so on. Conv2- 2 preserves better edge information [33]. Thus we leverage the S (2) to extract the edge features and other side paths to extract the salient object features [32].

In short this paper, extracts U-Net-based multi-resolution salient object characteristics The researchers next present a non-local salient edge features extraction module that combines local edge information with global location information to extract salient edge features.

The use of salient edge features improves the localisation and limits of salient objects. On six frequently used datasets, their model outperforms state-of-the-art approaches without any pre-processing or post-processing. We also provide evaluations on the EGNet's effectiveness.

Xingyi Zhou et al. [34] proposes ExtremeNet, ExtremeNet is a bottom-up object de- tection framework that recognizes an object's four extreme points (top-most, left-most, bottom-most, and right-most). To locate extreme points, the researchers employ a cuttingedge keypoint estimation framework that predicts four multi-peak heatmaps for each item category. In addition, they forecast the object center using one heatmap per category, which is the average of two bounding box edges in both the x and y dimensions. This architecture was introduced to tackle the limitations of Top-down aprroaches, which have dominated object detection for years. ExtremeNet uses keypoint prediction which is similar to CornerNet, which we have discussed earlier.

Two essential components of their technique are different: keypoint definition and grouping. A corner is a type of bounding box that has many of the same problems as top-down detection. A corner is generally seen outside of an object, with few distinguishing features. Extreme points, on the other hand, are found on objects, can be seen, and have a consistent local look. The topmost point of a human, for example, is usually the head, while the bottommost point of a car or airplane is usually a wheel. This simplifies the detection of extreme points. The geometric grouping is the second difference from CornerNet. Our detection system is entirely solely on appearance, with no implicit feature learning. The appearance-based grouping performs substantially better in our tests.

HoureglassNet is used by ExtremeNet to detect five keypoints which includes four extreme points and one center. The offset prediction is category-agnostic, but it is specialized to extreme points. For the center map, no offset is predicted. As a result, their network generates 5 C heatmaps and 4 2 offset maps, where C is the number of classes. Extreme points are located on opposite sides of an object. This makes grouping more difficult. An associative embedding, for example, might not have a broad enough view to group these keypoints. They use a different technique here, utilizing the dispersed nature of extreme points. Five heatmaps per class are used as input to our grouping algorithm: one center heatmap and four extreme heatmaps. They extract the corresponding keypoints from a heatmap by detecting all peaks. t, b, r, and l are four extreme points taken from heatmaps. Their geometric center is calculated as c = (lx+tx /2 , ty+by/ 2) For three equally spaced colinear objects of the same size, center grouping may result in a high-confidence falsepositive detection. These false-positive detections are referred to as "ghost" boxes. The researchers show how to remove ghost boxes with a simple postprocessing step. A ghost box, by definition, contains a large number of tiny detections. They apply a type of soft non-maxima suppression to prevent ghost boxes. If the sum of the scores of all boxes contained in a bounding box exceeds three times its own score, we divide it by two.

Extremes do not necessarily have a distinct definition. If an object's vertical or horizontal edges form extreme points, any point along those edges could be called an extreme point. As a result, instead of a single powerful peak response, their network provides a modest reaction along any aligned edges of the object. To solve this problem, the researchers employ edge aggregation. They aggregate the score of each extreme point, retrieved as a local maximum, in either the vertical (left and right extreme points) or horizontal (top and bottom keypoints) direction.

With at least twice as many annotated values as a basic bounding box, extreme points carry a lot more information about an object (8 vs 4). By building an octagon whose edges are centered on the extreme points, they present a simple way to approximate the object mask using extreme points. They extend an extreme point in both directions on its corresponding edge to a segment that is 1/4 of the edge length. When the section reaches a corner, it is shortened. The four segments' end points are then connected to make the octagon.

Their model gets an AP of 43.7 after multi-scale testing, exceeding all published onestage object detectors and matching popular two-stage detectors. It outperforms CornerNet by 1.6 percent, demonstrating the benefit of detecting extreme and center points over detecting corners with associative characteristics.

Wanli Ouyang et al. [35] proposed a deformable deep convolutional neural network for object detection called DeepID-Net. A new deformation constrained pooling (defpooling) layer in the proposed new deep architecture mimics the deformation of object pieces with geometric constraint and penalty. To learn feature representations that are more suitable for the object identification task and have strong generalization capabilities, a new pre-training technique is proposed. A set of models with great variety is created by modifying the net architectures, training procedures, and adding and removing some critical components in the detection pipeline, which significantly increases the effectiveness of model averaging.

Object detection with a new deep learning framework. Feature representation learning, part deformation learning, context modeling, model averaging, and bounding box location refining are all effectively integrated into the detection system. Extensive experimental examination provides a detailed component-by-component analysis. This is also the first study to look into the impact of CNN structures on large-scale item detection in the same context. Multiple detectors with substantial variability are formed by modifying the setup of this framework, resulting in more effective model averaging.A new pretraining technique for the deep CNN model.

Instead of using picture-level annotations, which are typically utilized in present deep learning object detection, we propose pretraining the deep model on the ImageNet image classification and localization dataset with 1000-class object-level annotations. The deep model is then fine-tuned on ImageNet/PASCAL-VOC object detection, which are the two datasets' targeting object classes.A novel deformation constrained pooling (defpooling) layer extends the deep model by learning the deformation of object pieces at different levels of information abstraction. The def-pooling layer can be used to learn the deformation properties of parts and replace the max-pooling layer.Candidate bound- ing boxes are proposed using a selective search method. In their experiment, an existing detector is employed to reject bounding boxes that are most likely to be background. To acquire 200 detection scores, an image region within a bounding box is clipped and fed into the DeepID-Net. Each detection score represents the degree of certainty in a cropped image comprising a single object class. The 1000-class whole-image classification scores of a deep model are used as contextual information to refine the detection scores of each candidate bounding box. To improve detection accuracy, the average of numerous deep model outputs is used. To mitigate localization mistakes, the RCNN proposes bounding box regression.

III. Methodology

3.1 Overview

This chapter comprises of the dataset used, its preparation, preprocessing and the architecture implemented to find the result. It also describes the process of how we trained our model by dividing the dataset into training data and testing data with validation.

3.2 Research Methodology

The main goal is to develop a robust camouflage object detection system using Transfer Learning on VGG16 architecture. We show the optimal accuracy we achieved using our customized training model on VGG16 architecture and the benefits we can obtain like detecting soldiers behind enemy lines in war also we can discover the life cycle of so many species which are endangered on deep sea.

3.3 Data Description

The dataset we gathered are from various internet sources. We collected a grand total of approximate 6500 pictures and divided them into 4 categories, Army, Pygmy seahorse, chameleon and octopus. Army consists of approximate 1300 pictures, pygmy and chameleon of 2000 each and octopus of over a 1000. However, we try to seek among most effective camouflage picture from the internet.

3.4 Data Preparation and Data preprocessing

After gathering the datasets from various sources and dividing them into 4 categories, we label the dataset using software. The link for this graphical image annotation tool is:https://github.com/tzutalin/labelImg

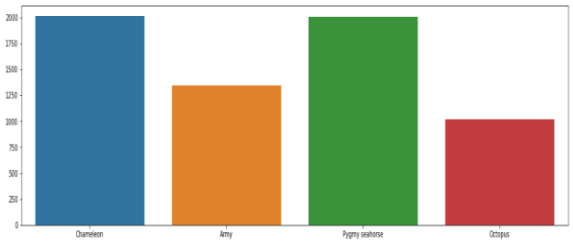

Figure 3-1: Chameleon (blue) , Army (Orange),Pygmy seahorse(Green),Octopus(Red)

From this bar chart we can see the images being divided into 4 categories. Then we label this image using labeling tools. The process of preprocessing labelling is we select that particular image, then we write the name

of the image so that the tool that detect the exact image more accurately that we are referring to.

3.5 Model Training

At first, we import OS because OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory. Then, we import cv2, it is the module name for opencv-python. After that we import glob, pyplot , sns and imageDataGenerator respectively. For the training we used seed 1000, image size 100 and batch size 128. In training generator, we used ImageDatagenerator

where, rotation range 30 width_shift_range=0.2, height_shift_range=0.2, zoom range = 0.3, horizontal_flip=True,validation_split=0.2.subset='training', class mode = 'sparse' has been used for train batch also in valid batch it remains same class mode but subset used as 'validation'. In train batch we found 5011 images belonging to 4 classes where as in validation batch it was 1252 images belonging to 4 classes. For the base model we used 'imagenet' as weights for our training model.

In briefly, The ImageNet project is a large visual database designed for use in visual object recognition software research. More than 14 million images have been hand-annotated by the project to indicate what objects are pictured and in at least one million of the images, bounding boxes are also provided. ImageNet contains more than 20,000 categories with a typical category, such as "balloon" or "strawberry", consisting of several hundred images.

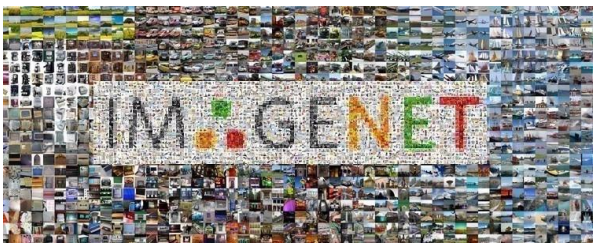

Figure 3-2:ImageNet

3.6 Architecture Selection

The preprocess data was used to train Transfer Learning on VGG16.During the architecture selection, We used ResNet50, Alexnet, SSD Mobile net and VGG16 architecture to test for accuracy. Among them VGG16 gives us highest amount of accuracy consider to all other architecture.

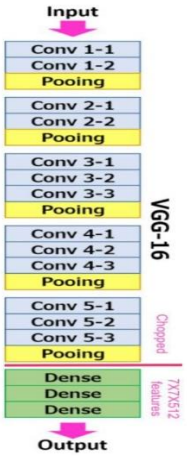

Figure 3.3 : VGG16 Architecture Model

About VGG16, On the ImageNet dataset, it was shown to be the highest performing model out of all the setups. A size 224 by 224 image with three channels – R, G, and B – is regarded the input to any of the network configurations. The only pre-processing done is to normalize each pixel's RGB values. Every pixel is subtracted from the mean value to achieve this. Following ReLU activations, the image is sent through a first stack of two convolution layers with a very small receptive area of 3 × 3. There are 64 filters in each of these two layers. The padding is 1 pixel, while the convolution stride is fixed at 1 pixel. The spatial resolution is preserved in this arrangement, and the output activation map is the same size as the input image dimensions. The activation maps are then run via spatial max pooling with a stride of 2 pixels over a 2 x 2-pixel window. The size of the activations is reduced by half. As a result, the activations at the bottom of the first stack are 112 × 112 x 64 in size.

The activations are then passed through a second stack, this time with 128 filters instead of 64 in the first. As a result, after the second layer, the size is 56 x 56 x 128. The third stack consists of three convolutional layers and a max pool layer. The number of filters used here is 256, resulting in a stack output size of 28 x 28 x 256. After that, there are two stacks of three convolutional layers, each with 512 filters. At the end of both stacks, the result will be 7 × 7 x 512.

Three fully connected layers follow the stacks of convolutional layers, with a flattening layer in between. The first two layers each feature 4,096 neurons, while the final fully connected layer serves as the output layer, with 1,000 neurons corresponding to the ImageNet dataset's 1,000 potential classifications. Following the output layer comes the Softmax activation layer, which is utilized for categorical categorization. [36].

In short about Transfer Learning, user can take the learning from previously trained network then customize that model according to their need. The benefit of transfer learning is if user has really small dataset, then if they try to build a network or train a network from the very scratch it will perform outstanding way. There is a misnomer that traditional machine learning is performing better compare to deep learning if the dataset size is very small. Now that is no longer true because using transfer learning technique in pre train network even if user has very small datasets, they can build a wonderful network which will outperform any machine learning or traditional machine learning approaches.

3.7 Summary

Overall, in this chapter we discussed about efficiency of VGG16 and its architecture. what is transfer learning and how it works. We also talked about ImageNet and out training model, and how we customized the architecture of our own model.

IV. Result Analysis

4.1 Overview

In this chapter we describe the result that we have gathered from our architecture. It mainly focuses on our architecture and how we trained our model in transfer learning on VGG16. Here, all the working procedures of our architecture has been discussed. We also explain about our train, validation accuracy and also train, validation loss. Finally, we discussed about the efficiency of our developed model. Our main goal is to find the highest accuracy which will work more effectively and robustly for detecting camouflage object.

4.2 Result Description

As previously mentioned, we used Transfer Learning on VGG16 architecture. We customize our own pre-train model. In our base model input shape is 100 and weights given 'Imagenet'. As we know that VGG16 was shown to be the highest performing model out of all setups on Imagenet. In this model we used train batch and 50 epochs.

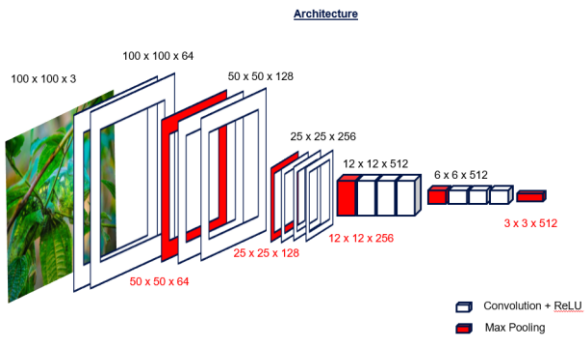

Figure 4.1: VGG16 Pre-train layer

During Implementations we made our datasets image shape is 100 x 100 with 3 filters. After that, it's gradually narrow down and remain 3 x 3 with 512 filters. Where activation function performs ReLu and 5 max polling or flattening layer remained. About flattening, it is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.

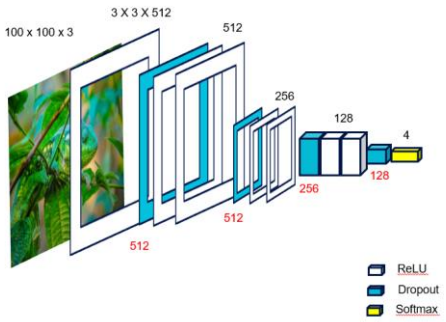

Figure 4.2: Extended layer of Architecture.

This is our customize layer (figure 4.2) where the number of filters getting reduced from 512 to 4 neurons. At the end it remains 4 neurons and they are Army, Octopus, Pygmy seahorses and Chameleon. There are two activation function has been used one is ReLu and another one is softmax.

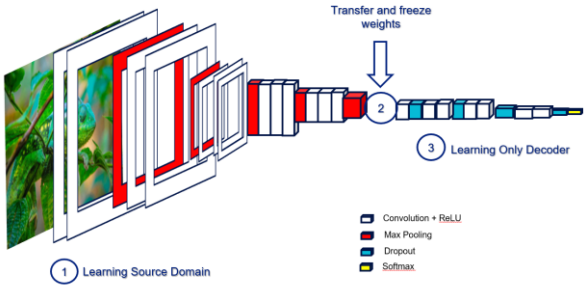

Figure 4.3: Cam_VGG16 full architecture

In figure 4.3 we draw a visual architecture of our network where all the layer has been discussed.

Train Accuracy	Validation Accuracy	Train Loss	Validation Loss
95%	91%	0.1%	0.29%

Table 4-A: It represents predicted result from the graph.

Considering Train accuracy, there are so many ups and downs trend until 50 epochs. Where it gives us almost 95% accuracy. On the other way, there was huge

fluctuations on validation accuracy and it's shows nearly 91% accuracy until 50 epochs run.

Moving to the Loss part both train loss and validation loss has downwards trend, compared to validation loss train loss gives more better result.

For presenting, accuracy graph (15,5) figure size has been made for plotting figure where subplot size was (1,2,1). There are two different labels on graph one is train accuracy another one is validation accuracy. On the other hand, for showing Loss we used subplot (1,2,2) and two different labels where train loss and validation loss exist.

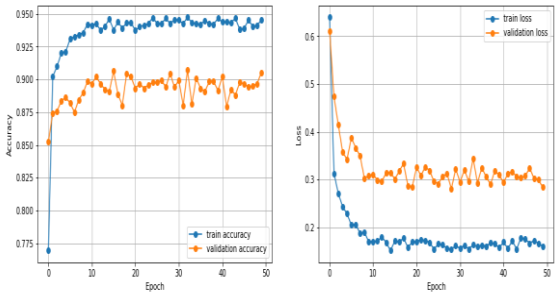

Figure 4.4: Result of this Model

As it shown from the picture Epoch are depicted horizontal line where Accuracy and Loss in Vertical line.

4.3 Summary

To conclude this chapter, we mainly focus on the making of the architecture, how we added our own customized layer and overall layer description. After that, we added the accuracy and loss from the result section and discussed about it.

V. Conclusion

5.1 Overview

This chapter demonstrates the summary of the research. The widespread objective of this research is to detect camouflage objects using Deep Learning techniques. We designed a vgg16 architecture, customized it by implementing Transfer Learning on it. The accuracy obtained from training and testing our model proves it be to robust and efficient.

5.2 Framework of this Model

The proposed framework of this research was based on Transfer Learning. In Transfer Learning, the user can take the learning from the previously trained network and customize the model according to the user needs. The most significant advantage of transfer learning are resource savings and increased efficiency while training new models.

5.3 Implementation of Methodology and Objective

We implement object detection system using Deep Learning techniques and improve the vgg16 model with Transfer Learning. Here, we develop our model in jupyter notebook using TensorFlow platform comparing several types of architecture seeking for an optimal accuracy which can detect the camo objects in a more robust and effective manner.

5.4 Limitation of the Study

We faced several drawbacks while conducting this research.

1) In our data collection step, that is collecting thousands of images from various internet sources, we had to make sure all the images are .jpeg or .png, any other formats will result in Unknown image file format.

Figure5.1: Mismatch Of File Format

2) We need a powerful gpu, the gpu we trained our model in is GeForce 2060. An outdated graphics card will not work.

3) We need to label images using labelme, while labelling we have to make sure to include weight as image net.

4) While collecting the dataset, we needed to find the right kind of images, that is, images that is properly camouflaged.

5.5 Recommendation for Future Work

Object detection technology's future is still being proven, it has the potential to release people from tiresome work that computers can accomplish more efficiently and effectively. It will also open up new research and operational possibilities, which will yield more benefits in the future. As a result, these problems avoid the requirement for extensive training that requires a large number of datasets in order to perform more sophisticated jobs. With continuing evolution, as well as the devices and methodologies that enable it, it could soon become the next big thing in the future.

Most present algorithms only address a tiny subset of the various tasks required for image comprehension and are expensive. To replicate a fraction of the normal person's capacity to recognize objects, one would need to merge multiple distinct algorithms into a single system that runs in real time, which would be a huge problem with today's hardware. Detecting objects is a crucial task for most computer vision systems.

5.6 Summary of The Research

We have developed a robust object detection system using deep learning technique. The architecture we worked on is vgg16 and customized it using Transfer Learning. We worked on the jupyter notebook using the TensorFlow platform. We obtained an accuracy of approximate 95% when we trained our model. The validation accuracy we achieved is approx. 91%. Train Loss is 0.1% and validation loss is 0.29%. In training accuracy there are many ups and downs with trend until 50 epochs. When the dataset size is small, there is a misconception that classical machine learning outperforms deep learning. That is no longer the case since, by employing the transfer learning technique in the pre-train network, users may design a fantastic network that outperforms any machine learning or traditional machine learning approaches, even if they have extremely small dataset.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," pp. 91-110, 2004.

[2] J. M. a. J. P. Serge Belongie, "Shape matching and object," 2002.

[3] N. D. a. B. Triggs, " Histograms of oriented gradients for human detection," pp. 886-893, 2005.

[4] D. M. a. D. R. Pedro Felzenszwalb, "A discriminatively trained, multiscale, deformable part model," pp. 1-8, 2008.

[5] Y. B. G. H. e. a. Yann LeCun, " Deep Learning," pp. 436-444, 2015.

[6] Z. S. Y. G. J. Y. Zhengxia Zou, "Object detection in 20 years," 2019.

[7] T. P. Constantine Papageorgiou, " A trainable system for object detection," pp. 15-33, 2000.

[8] M.-M. C. Q. H. H. J. a. J. L. Ali Borji, "Salient Object Detection," p. 2019, 117-150.

[9] S. H. F. E. a. S. S. Radhakrishna Achanta, "Frequency-tuned salient region detection," pp. 1597-1604, 2009.

[10] S. B. L. X. H. Q. Q. H. a. Q. T. Kaiwen Duan, "Centernet: Keypoint triplets for object detection," pp. 6579-6578, 2019.

[11] J. D. Hei Law, " Cornernet: Detecting objects as paired keypoints," pp. 734-750, 2018.

[12] B. Y. Y. C. R. H. a. R. U. Ming Liang, " Multi-task multi-sensor fusion for 3d object detection," pp. 7345-7353, 2019.

[13] P. L. a. R. U. Andreas Geiger, "Are we ready for autonomous driving? the kitti vision benchmark suite," pp. 3354-3361, 2012.

[14] W. L. R. U. Bin Yang, " Pixor: Real-time 3d object detection from point clouds," pp. 7652-7660, 2018.

[15] W. L. C. W. H. S. L. J. G. Charles R Qi, "Frustum pointnets for 3d object detection from rgb-d data," pp. 918-927, 2018.

[16] H. S. K. M. a. L. J. G. Charles R Qi, "Pointnet: Deep learning on point sets for 3d classification and segmentation," pp. 652-660, 2017.

[17] L. Y. H. S. L. J. G. Charles Ruizhongtai Qi, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," pp. 30-37, 2017.

[18] G. G. P. D. a. R. G. Kaiming He, " Mask R-CNN," pp. 2961-2969, 2017.

[19] F. S. H. L. Y. J. L. L. a. J. S. Tao Kong, "Foveabox: Beyound anchor-based object detection," pp. 7389-7398, 2020.

[20] Y. C. N. W. a. Z. Z. Yanghao Li, "Scale-aware trident networks for object detection," pp. 6054-6063, 2019.

[21] J. G. Z. Z. J. D. a. Y. W. Han Hu, "Relation networks for object detection," pp. 3588-3597, 2018.

[22] R. P. Q. V. L. Mingxing Tan, "Efficientdet: Scalable and efficient object detection," pp. 10781-10790, 2020.

[23] Z. Z. C. H. C. G. M. D. M. J. Xuebin Qin, " Basnet: Boundary-aware salient object detection," pp. 7479-7489, 2019.

[24] M. K. M. R. N. D. B. a. Y. W. Md Amirul Islam, " Salient object detection using a context-aware refinement network," , 2017.

[25] X. H. L. Z. X. X. J. Q. G. H. P.-A. H. Zijun Deng, "R3net: Recurrent residual refinement network for saliency detection," 2018.

[26] X. G. a. Y. Bengio, " Understanding the difficulty of training deep feedforward neural networks," pp. 249-256, 2010.

[27] C.-Y. W. a. H.-Y. M. L. Alexey Bochkovskiy, "Optimal speed and accuracy of object detection," 2020.

[28] Y. C. S. Z. G. H. S. L. Zhuliang Yao, " Crossiteration batch normalization," pp. 12331-12340, 2021.
[29] L. Q. H. Q. J. S. J. J. Shu Liu, "Path aggregation network for instance segmentation," pp. 8759-8768, 2018.
[30] Y. W. Q. T. J. G. C. X. a. C. X. Kai Han, "Ghostnet: More features from cheap operations," pp. 1580-1589, 2020.
[31] X. G. a. Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," pp. 249-256, 2010.
[32] J.-J. L. D.-P. F. Y. C. J. Y. M.-M. C. Jia-Xing Zhao, "Egnet: Edge guidance network for salient object detection," pp. 8779-8788, 2019.
[33] D. W. H. L. H. W. X. R. Pingping Zhang, "Amulet: Aggregating multi-level convolutional features for salient object detection," pp. 202-211, 2017.
[34] J. Z. P. K. Xingyi Zhou, "Bottom-up object detection by grouping extreme and center points," pp. 850-859, 2019.
[35] X. W. X. Z. S. Q. P. L. Y. T. H. L. S. Y. Z. W. C.-C. L. e. a. Wanli Ouyang, "Deepid-net: Deformable deep convolutional neural networks for object detection," pp. 2403-2412, 2015.
[36] K. S. a. A. Zisserman., "Very deep convolutional networks for large-scale image recognition," 2015.

Abbreviations:

VGG	Visual Geometry Group
CNN	Convolutional Neural Network
HOG	Histogram of Oriented Gradients
SVM	Support Vector Machine
NLP	Natural Language Processing
MLP	Multi-Layer Perceptron
IOU	Intersection over Union
BIFPN	Bi-directional Feature Pyramid Network
RM	Refinement Module
etc.	etc.

Table A1:

Symbols	Name:
Φ	Phi
etc.	etc.

Khalak Bin Khair was born in Feni. His B.Sc. Degree on going in Computer Science and Engineering from American International University-Bangladesh (AIUB) in 2022. He is looking forward to doing his M.Sc. in Deep Learning. His research interests are based on Computer Vision, Object detection, image classification, Machine Learning, Artificial Intelligence. .

Saqib Jahir Chowdhury was born and raised in Kuwait, he was born on August 3rd 1999. He is currently enrolled in American International University-Bangladesh (AIUB) pursuing a degree in Computer Science and Engineering (CSE). He is set to complete his graduation by end of 2022. His area of interests comprises of Web Development, Computer Vision and Database Management.

Mohammed Ibrahim a Bangladeshi, is currently pursuing his B.Sc. Engg. degree in Computer Science and Engineering at American International University-Bangladesh (AIUB). His area of specialisation include Data Science, Database, Software Engineering and Web Development. Besides this, he has also been an active member and a promising team leader of AIESEC in Bangladesh.

.

Fahad Bin Ismail By birth a Bangladeshi, he is currently pursuing his Bachelor in computer science and Engineering at American International university Bangladesh (AIUB).His area of specialization is Data Science, computer vision and pattern recognition, Software Engineering, Database . He is an active member of different professional organizations, including IEEE (member), ICT Olympiad Bangladesh-AIUB (member).

Dr. Debajyoti Karmaker He is working as an Assistant professor in the department of computer science at American International University-Bangladesh. he worked as Postdoctoral Research Fellow at Australian National University (ANU), and Stanford University. Before joining ANU, He completed his Ph.D. from The University of Queensland (UQ). his research interests are in Deep Learning, Computer Vision, & Machine Learning. his particularly interested in the areas of image classification, object detection, segmentation, bio-inspired collision avoidance strategies, and Robust Decision-

making and Learning. Before starting his Ph.D., he was working as a Lecturer at the American International University-Bangladesh (AIUB) - in the Department of Computer Science. He also worked as a software engineer at Infra Blue Technology (IBT Games)..