Vision-based Obstacle Detection and Autonomous Navigation of Unmanned Ground Vehicles

Project Phase-1 Report

Submitted by

Sujeet Kumar Singh

Roll No.- 214363009

Under the Supervision of

Prof. S.K. Dwivedy & Dr. Prithwijit Guha

Department of Electronics and Electrical & Mechanical Engineering Indian Institute of Technology, Guwahati ASSAM – 781039,

Contents

Abstract

In the recent period, unmanned ground vehicles (UGVs) have increased rapidly in every field. Due to fast growth in artificial intelligence technology, UGVs are more capable of doing tough tasks very efficiently and accurately without human intervention. In the recent, period Covid19 pandemic and Russia Ukraine war UGVs have been used extensively. The application of UGVs broadly increased in every field whether it is an agricultural field, medical field, defense field, or space field. Due to the advent of deep learning, works that are not solvable can be easily solved by deep learning. Many computer vision tasks with the help of deep learning can be handled effectively and efficiently, these are possible because of the availability of large datasets and the high hardware capability of computers. Obstacle detection and classification is one of the most important tasks for an unmanned ground vehicle. Vision-based approaches are popular for this task because it is cost-effective and makes human-like perception. So many artificial neural network techniques developed in the past few years to mimic the human visual system. Convolutional neural networks (R-CNN, YOLO, etc.) are widely used in object detection. A literature survey has been performed for identifying suitable vision-based obstacle detection for UGVs.

Introduction

Unmanned Ground Vehicle (UGV)

Unmanned ground vehicles are robotic systems that operate on the ground without the onboard human operator. Nowadays UGVs are widely used for civilian and military purposes, particularly in environments that are harmful or unpleasant to humans.

Based on its application, unmanned ground vehicles will generally include the following components: platform, sensors, control systems, guidance interface, communication links, and systems integration features.

Platform

The platform can be based on an all-terrain vehicle design and includes the locomotive apparatus, sensors, and power source. Wheels, tracks, and legs are the common forms of locomotion. wheels are power efficient and allow the highest speed on flat ground but are not suitable for traversing off-road and uneven terrain, as they get stuck or sink a low contact area and thus higher pressure. The track is good for rugged terrain but less efficient and low speed due to mechanical complexity and high vibration. Legs types can be used in a variety of terrain but it has low speed and require complex control and stability hardware.

Control systems

UGVs are controlled remotely or autonomously or by both with the help of artificial intelligence technology.

Guidance Interface

Depending on the type of control system, the interface between machine and human operator can include a joystick, computer programs, or voice command.

Sensors

Sensors are used for navigation and to perceive the environment. Some commonly deployed sensors on UGVs are lasers, ultrasound, RADAR, camera, odometer, gyroscope, inclinometer, etc.

Sensors are crucial for efficient work UGVs. Sensors can be classified into two types

- Exteroceptive Sensors (ESs)
- Proprioceptive Sensors (PSs)

Exteroceptive sensors are used to perceive external environmental information, e.g., LiDAR, millimeters-wave, ultrasonic, and cameras. While Proprioceptive sensors are used for real-time information about the platform itself, e.g., vehicle speed, acceleration, altitude angle, wheel speed, and position to ensure real-time state estimation of UGV itself. Common Proprioceptive Sensors are GNSS and IMU.

Light Detection and Ranging (LiDAR)

LiDAR is used for object position, orientation, and velocity information by transmitting and receiving laser beams and calculating time differences. The collected data type is a series of 3D

point information called a point cloud, more specifically the coordinate of the object to the radar coordinate system and echoes intensity. Lidar is mainly used in SLAM, point cloud matching and localization, object detection, trajectory prediction, and tracking. Lidar cannot collect the color and texture information of the target.

Radio Detection and Ranging (RADAR)

The working principle of RADAR is similar to LiDAR, the key difference here we used radio waves to detect the target. Compared with Lidar, Radar has a longer detection range, smaller size, lower price, and is not easily affected by light and weather conditions.

Ultrasonic

Ultrasonic detects objects by emitting sound waves and is mainly used in the field of ships. Ultrasonic is small in size, low in cost, and not affected by weather and light conditions, but its detection distance is short, the accuracy is low, it is prone to noise, and it is also easy to interfere with other equipment.

Computer Vision

Computer vision is the area of study in which computers are empowered to visualize, recognize and process what they see in a similar way as that of humans [24]. The main aim of computer vision is to generate relevant information from image and video data in order to extract something about the world [25][26]. It can be classified as a sub-field of artificial intelligence and machine learning. Applications of computer vision include image classification, visual detection, 3D scene reconstruction from 2D images, image retrieval, augmented reality, machine vision and traffic automation

Monocular Camera

Monocular cameras store environmental information in the form of pixels by converting optical signals into electrical signals. Advantages monocular cameras as compared to Lidar, Radar, and ultrasonic can generate high-resolution images having environmental color and texture information, and also low cost. The drawback of a monocular camera is that it cannot obtain depth information, it is highly susceptible to illumination conditions and weather conditions, and the high computing power required for high-quality images challenges the real-time performance of the algorithm.

Stereo Camera

The working principle of a stereo camera is similar to a mono camera, compared to a mono camera stereo camera is equipped with an additional lens at a symmetrical position, the depth information and movement of the environment can be captured by taking two pictures at the same time through multiple viewing angles information. However, it is also susceptible to weather conditions field view is narrow, and more computing power is required.

Infrared Camera

Infrared cameras collect environmental information by receiving signals of infrared radiation from objects**.** The infrared camera is sensitive to the wavelength of 0.15 μm to 15 μm in practical applications, a corresponding infrared camera is selected according to the wavelength of different detection targets.

Computer vision is the subfield of computer science, artificial intelligence, and machine learning, it mimics the human vision system, because of deep learning and neural networks computer vision accuracy increases significantly.

Difference between image classification, object localization, and Object detection

- Image Classification: Predict the type or class of an object in an image. A single image with a single object is classified with types of class.
- Object Localization: Locate the presence of objects (one or more than one) in an image and indicate their location with a bounding box.
- Object Detection: Locate the presence of objects with a bounding box and types or classes of the located objects in an image.

Simultaneous Localization and Mapping (SLAM)

Visual SLAM is used instead of LiDAR for the recording of landmarks in a scene. Visual SLAM has the advantages of rich visual data, low-cost, lightweight, and low power consumption very less computational workload involved in post-processing. The visual SLAM problem consists of steps such as environment sensing, data matching, motion estimation, as well as location update, and recording of new landmarks [1]. Making a model of how visual objects appear in different conditions like 3D rotation, scaling, lighting, and extending from that representation using a strong form of transfer learning to one-shot learning is a difficult problem in this domain. Feature extraction and data representation methods can be useful to reduce the number of training examples needed for an ML model [2]. A two-step approach is normally used in image-based localization; place recognition is trailed by pose estimation. The place computes a global descriptor for each of the images by aggregating local image descriptors, e.g. SIFT, using the bag-of-words approach. Each global descriptor is stored in the database together with the camera pose of its associated image concerning the 3D point cloud reference map. Similar global descriptors are extracted from the query image and the closest global descriptor in the database can be retrieved via an efficient search. The camera pose of the closest global descriptor would give us a coarse localization of the query image concerning the reference map. In pose estimation, the exact pose of the query image is calculated more precisely with algorithms such as the Perspective-n-Point (PnP) [3] and geometric verification [4] algorithms. [5] The success of image-based place recognition is largely attributed to the ability to extract image feature descriptors. Unfortunately, there is no algorithm to extract local features similar to SIFT for LiDAR scans. A 3D scene is composed of 3D points and database images. One approach has associated each 3D point to a set of SIFT descriptors corresponding to the image features from which the point was triangulated. These descriptors can then be averaged into a single SIFT descriptor that describes the appearance of that point [6]. Another approach constructs multi-

modal features from RGB-D data rather than depth processing. For the depth processing part, they adopt the well-known colorization method based on surface normal, since it has been proved to be effective and robust across tasks [7]. Another alternative approach utilizing traditional CV techniques presents the Force Histogram Decomposition (FHD), a graph-based hierarchical descriptor that allows the spatial relations and shape information between the pairwise structural subparts of objects to be characterized. An advantage of this learning procedure is its compatibility with traditional bags-of-features frameworks, allowing for hybrid representations gathering structural and local features [8**].**

Machine Learning

Machine learning is one of the applications of Artificial Intelligence (AI) which enables the computers to learn on their own and perform tasks without human intervention [15]. There are numerous applications of machine learning algorithms in the field of computer vision.
There are three types of machine learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Supervised Learning
In Supervise learning the algorithm required direct supervision, data labelled, annotation etc, are done before applying the algorithm of machine learning. The algorithms learn from labelled data and predict the annotations of the new data based on the training. These are the some popular supervised learning algorithms as follows:
• Neural Networks
• Decision Trees
• Linear Regression
• Logistic Regression
• Support Vector Machines
• Random Forest
• K-Nearest Neighbours

Unsupervised Learning
In unsupervised learning labelling/annotation is not required. In this technique algorithms tries to find properties between the classes without the intervention of human. K-means cluttering, DBSCAN are example of unsupervised Learning.

Reinforcement Learning
In Reinforcement learning, the machine is allowed to train itself continually using trial and error. The machine learns from past experience and attempts to capture the best knowledge possible to predict accurately. Markov Decision Process, Q-learning, etc. are some of the examples of reinforcement learning.

Literature Survey

Deep Learning

To gain a fundamental understanding of deep learning we need to consider the difference between descriptive analysis and predictive analysis. The descriptive analysis involves defining a comprehensible mathematical model which describes the phenomenon that we want to observe. This involves collecting data about a process, forming a model on patterns in the data, and validating these models by comparing the outcome of descriptive models we form with the real outcome [9]. Producing such models is precarious however because there is always a risk of unmodelled variables that scientists and engineers neglect to take in due to ignorance or failure to understand some complex, hidden or non-intuitive phenomena [10]. The predictive analysis involves the discovery of rules that underlie a phenomenon and form a predictive model which reduces the error between the actual and the predicted outcome considering all possible interfering factors [9]. Machine learning rejects the traditional programming where problem analysis is replaced by a training framework where the system is fed a large number of the training dataset, sets of inputs for which the desired outputs are known. Which it learns and uses to compute new patterns [10]. Deep learning (DL) is a subset of machine learning and machine learning is a subset of artificial learning. DL is based largely on Artificial Neural Networks (ANNs), inspired by the functioning of the human brain. Like the human brain, it is composed of many computing cells or 'neurons' that each perform a simple operation and interacts with each other to make a decision [12]. Deep Learning is all about learning or 'credit assignment' across many layers of a neural network accurately, efficiently, and without supervision and enabling advancements in computing power [13]. Self-organization and the exploitation of interactions between small units have been confirmed to perform better than central control, particularly for complex non-linear process models [14].

Deep Learning (DL) is used extensively in digital image processing to solve difficult problems like classification, segmentation, and detection. DL methods such as Convolutional Neural Networks (CNNs) mostly improve prediction performance using big data and abundant computing resources and have extended the boundaries of what was possible earlier. The beauty of DL is that the problems that were assumed to be unsolvable are now being solved with super-human accuracy. Image classification is a prime example of this kind of problem. Since being reignited by Sutskever, Krizhevsky, and Hinton in 2012 [30], DL has dominated the domain ever since due to substantially better performance compared to traditional methods. Is DL making traditional Computer Vision techniques obsolete? Has DL superseded traditional computer vision techniques? Is there still a need to study traditional CV techniques when DL seems to be so

effective? These are all questions that have been carried up in the community in recent years [31]. There are some problems where traditional techniques with global features are a better solution. The advent of DL may open many doors to do something with traditional techniques to overcome the many challenges DL bring computing power, time, accuracy, characteristics, and quantity of inputs, and among others.

Some top deep performing model for obstacle detection

R-CNN Model Family

The R-CNN family of methods refers to the R-CNN, which stands for "*Regions with CNN Features*" or "*Region-Based Convolutional Neural Network*," developed by Grishick et al. This includes the techniques R-CNN, Fast R-CNN, and Faster-RCNN designed and demonstrated for object localization and objects recognition.

R-CNN

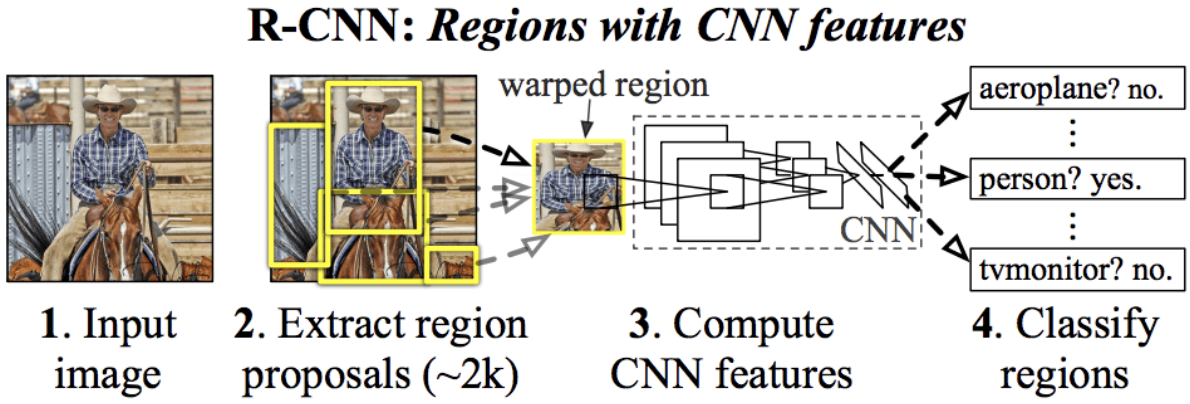
The R-CNN was described in the 2014 paper by Ross Girshick, et al.[#]

It has been one of the first large and successful applications of convolutional neural networks to the problem of object localization, detection, and segmentation. The approach was demonstrated on standard datasets, achieving then state-of-the-art results on the VOC-2012 dataset and the 200-class ILSVRC-2013 object detection dataset.

Their proposed R-CNN model has contained three modules; they are:

- **Module 1: Region Proposal**. Generate and extract category-independent region proposals, e.g., candidate bounding boxes.
- **Module 2: Feature Extractor**. Extract features from each candidate region, e.g. using a deep convolutional neural network.
- **Module 3: Classifier**. Classify features as one of the known classes, e.g., linear SVM classifier model.

The architecture of the model is summarized in the image below, taken from the paper.



The feature extractor used by the model was the Alex Net deep CNN image classification competition. The output of the CNN was a 4,096-element vector that describes the contents of the image that is fed to a linear SVM for classification, specifically one SVM is trained for each known class. It is a relatively simple application of CNNs to the problem of object localization and recognition. A downside of the approach is that it is slow, requiring a CNN-based feature extraction pass on each of the candidate regions generated by the region proposal algorithm. This is a problem as the paper describes the model operating upon approximately 2,000 proposed regions per image at test time.

Fast R-CNN

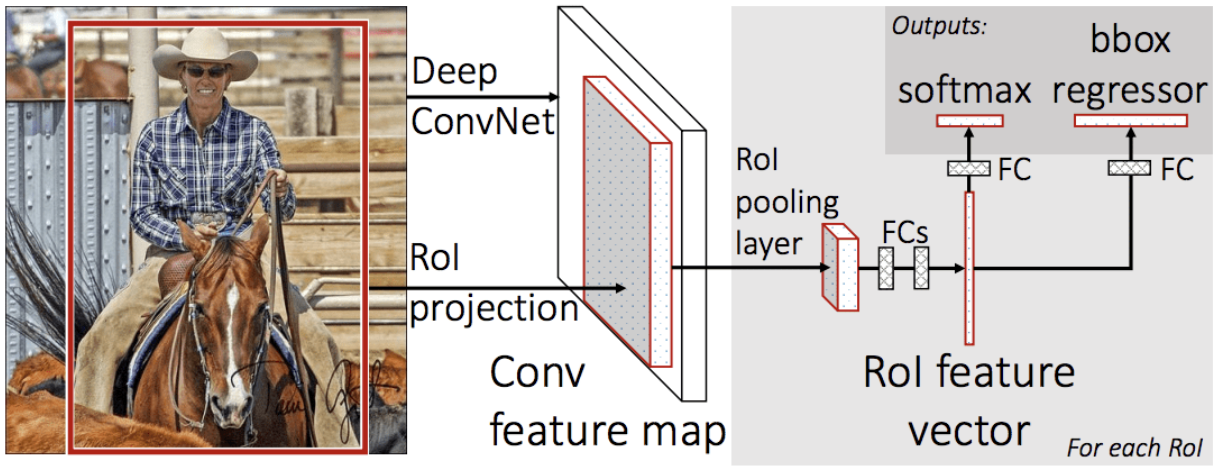
Fast R-CNN is proposed as a single model instead of a pipeline to learn and output regions and classifications directly.

The architecture of the model takes the photograph of a set of region proposals as input that are passed through a deep convolutional neural network. A pre-trained CNN, such as a VGG-16, is used for feature extraction. The end of the deep CNN is a custom layer called a Region of Interest Pooling Layer, or RoI Pooling, that extracts feature specific to a given input candidate region.

The architecture of the model takes the photograph of a set of region proposals as input that are passed through a deep convolutional neural network. A pre-trained CNN, such as a VGG-16, is used for feature extraction. The end of the deep CNN is a custom layer called a Region of Interest Pooling Layer, or RoI Pooling, that extracts feature specific to a given input candidate region.

The output of the CNN is then interpreted by a fully connected layer then the model separates into two outputs, one for the class prediction via a softmax layer, and another with a linear output for the bounding box. This process is then repeated multiple times for each region of interest in a given image.

The architecture of the model is summarized in the image below, taken from the paper.



The model is significantly faster to train and make predictions, yet still requires a set of candidate regions to be proposed along with each input image.

Faster R-CNN

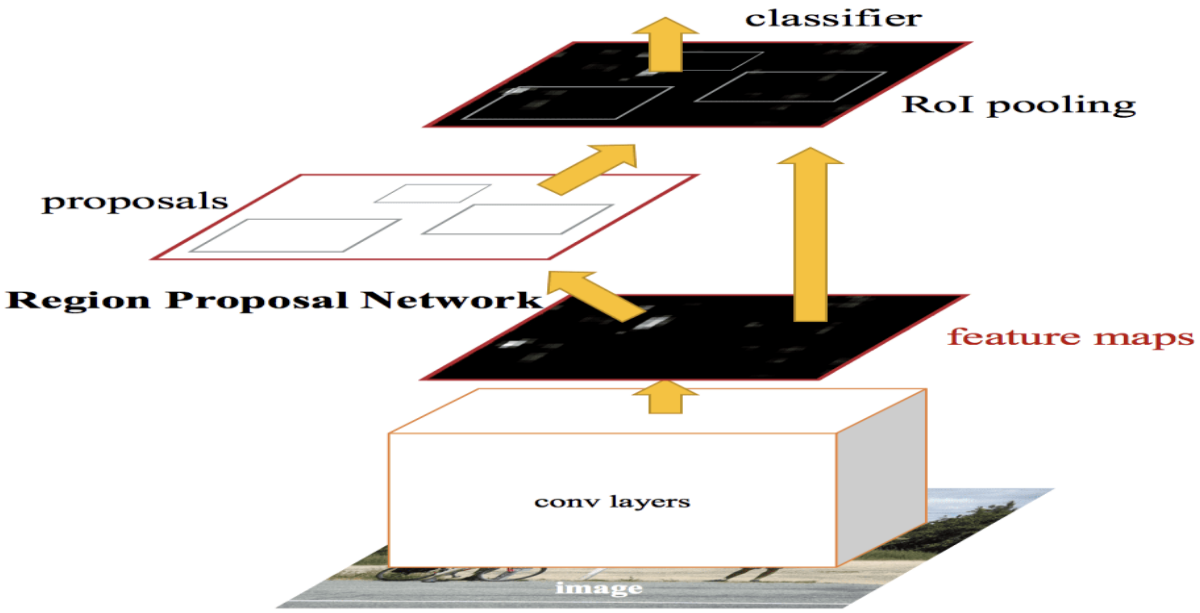
The model architecture was further improved for both speeds of training and detection by Shaoqing Ren, et al. at[@] Microsoft Research in the 2016 paper. The architecture was designed to both propose and refine region proposals as part of the training process, referred to as a Region Proposal Network, or RPN. These regions are then used in concert with a Fast R-CNN model in a single model design. These improvements both reduce the number of region proposals and accelerate the test-time operation of the model to near real-time with then state-of-the-art performance.

it is a single unified model; the architecture is comprised of two modules:

- **Module 1: Region Proposal Network**. Convolutional neural network for proposing regions and the type of object to consider in the region.
- **Module 2: Fast R-CNN**. Convolutional neural network for extracting features from the proposed regions and outputting the bounding box and class labels.

Both modules operate on the same output of a deep CNN. The region proposal network acts as an attention mechanism for the Fast R-CNN network, informing the second network of where to look or pay attention.

The architecture of the model is summarized in the image below, taken from the paper.



The RPN works by taking the output of a pre-trained deep CNN, such as VGG-16, passing a small network over the feature map, and outputting multiple region proposals and a class prediction for each. Region proposals are bounding boxes, based on so-called anchor boxes or pre-defined shapes designed to accelerate and improve the proposal of regions. The class prediction is binary, indicating the presence of an object, or not, the so-called "*objectness*" of the proposed region.

A procedure of alternating training is used where both sub-networks are trained at the same time, although interleaved. This allows the parameters in the feature detector deep CNN to be tailored or fine-tuned for both tasks at the same time.

This Faster R-CNN architecture is the peak of the family of models and continues to achieve near state-of-the-art results on object recognition tasks. A further extension adds support for image segmentation, described in the paper 2017 paper "Mask R-CNN."

YOLO Model Family

Another popular family of object recognition models is referred to collectively as YOLO or "*You Only Look Once*," developed by Joseph Redmon, et al. [#]
The R-CNN models are generally more accurate, yet the YOLO family of models are fast, much faster than R-CNN, achieving object detection in real-time.

YOLO

The YOLO model was first described by Joseph Redmon, et al. in the 2015 paper titled "You Only Look Once: Unified, Real-Time Object Detection." Note that Ross Girshick, developer of R-CNN, was also an author and contributor to this work, then at Facebook AI Research.
The approach involves a single neural network trained end to end that takes a photograph as input and predicts bounding boxes and class labels for each bounding box directly. The technique offers lower predictive accuracy (e.g., more localization errors), although operates at 45 frames per second and up to 155 frames per second for a speed-optimized version of the model. The model works by first splitting the input image into a grid of cells, where each cell is responsible for predicting a bounding box if the centre of a bounding box falls within the cell. Each grid cell predicts a bounding box involving the x, y coordinate, the width and height, and the confidence. A class prediction is also based on each cell.
For example, an image may be divided into a 7×7 grid and each cell in the grid may predict 2 bounding boxes, resulting in 94 proposed bounding box predictions. The class probabilities map and the bounding boxes with confidences are then combined into a final set of bounding boxes and class labels. The image taken from the paper below summarizes the model's two outputs.

YOLOv2 (YOLO9000) and YOLOv3

The model was updated by Joseph Redmon and Ali Farhadi to further improve model performance in their 2016 paper titled "YOLO9000: Better, Faster, Stronger."
Although this variation of the model is referred to as YOLO v2, an instance of the model is described that was trained on two object recognition datasets in parallel, capable of predicting 9,000 object classes, hence given the name "*YOLO9000*."
Several architectural changes were made to the model, such as the use of batch normalization and high-resolution input images. Like Faster R-CNN, the YOLOv2 model makes use of anchor boxes, pre-defined bounding boxes with useful shapes and sizes that are tailored during training. The choice of bounding boxes for the image is pre-processed using a k-means analysis on the training dataset.
Significantly, the predicted representation of the bounding boxes is changed to allow small changes to have a less dramatic effect on the predictions, resulting in a more stable model. Rather than predicting position and size directly, offsets are predicted for moving and reshaping the pre-defined anchor boxes relative to a grid cell and dampened by a logistic function.

Further improvements to the model were proposed by Joseph Redmon and Ali Farhadi in their 2018 paper titled “YOLOv3: An Incremental Improvement.” The improvements were reasonably minor, including a deeper feature detector network and minor representational changes.

Advantages of Deep Learning Over Traditional computer vision technique

Rapid advances in DL and improvements in device capabilities computing power, memory capacity, power consumption, and image sensor resolution have improved the performance and cost-effectiveness of further speeding up the spread of vision-based applications. Compared to traditional CV techniques, DL empowers CV engineers to achieve greater accuracy in tasks such as image classification, semantic segmentation, object detection, and Simultaneous Localization and Mapping (SLAM). Since neural networks used in DL are trained rather than programmed, applications using this approach often require less expert analysis and fine-tuning and exploit a large number of video datasets. DL also provides more flexibility because convolutional neural models can be re-trained using a custom dataset for any use, contrary to CV algorithms, which need to be more domain-specific. Taking the problem of object detection on a mobile robot as an example, we can compare the two types of algorithms for computer vision:

The traditional computer vision approach is to use well-established CV techniques such as feature descriptors like SIFT, SURF, etc., for object detection. Before the emergence of DL, a step called feature extraction was carried out for tasks such as image classification. Features are small interesting, descriptive, or informative patches in images. Several CV algorithms, such as edge detection, and corner detection may be involved in this step. As many features as practicable are extracted from images and these features form a bag-of-words of each object class. At the deployment stage, these definitions are searched for in other images. If a significant number of features from one bag of words are in another image, the image is classified as containing that specific object like a table, dog, cat, etc. The difficulty in the traditional approach is that choosing which features are important in each given image is necessary. As the number of classes increases, feature extraction becomes more and more tough work. It depends on the CV engineer’s skill and judgment and a long trial and error process to decide which features best describe different classes of objects. Additionally, each feature definition requires dealing with an excess of parameters, all of which must be adjusted by the CV engineer. DL introduced the concept of end-to-end learning where the machine is just given a dataset of images that have been marked with what classes of the object are present in each image [7]. Thereby a DL model is ‘trained’ on the given data, where neural networks discover respective patterns in classes of images and automatically work out the most descriptive and salient features to each specific class of object for each object. It has been well-established that DNNs perform far better than traditional algorithms, although it takes time to computing requirements and training time. By combining both arts DL and computer vision in CV employing this methodology, the workflow of the CV engineer has changed dramatically where the knowledge and expertise in extracting hand-crafted features have been replaced by knowledge and expertise in iterating through deep learning architectures as depicted in Fig 1

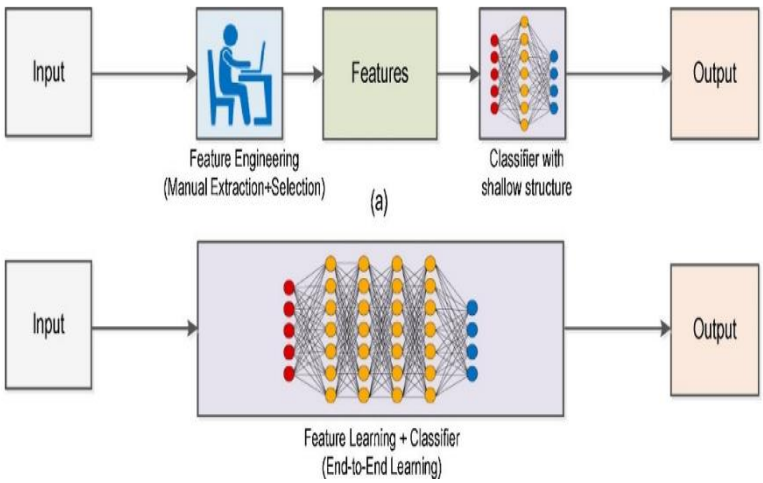


Fig. 1. Workflow of traditional computer vision and deep learning

The development of CNNs has had a great influence in the field of CV in recent years and is responsible for a big jump in the ability to identify objects [9]. This progress has been enabled by an increase in hardware capabilities like computing power, as well as an increase in the amount of data available for training neural networks. The recent explosion in widespread adoption of various deep-neural network architectures for CV is in the seminal paper ImageNet Classification with Deep Convolutional Neural Networks, CNNs make use of kernels (filters), to detect features throughout an image. A kernel is a matrix of values, which are called weights, which are trained to detect specific features. As their name indicates, the main idea behind the convolutional neural network is to spatially convolve the kernel on a given input image and check if the feature it is intended to detect is present. To provide a value representing how confident it is that a specific feature is present, a convolution operation is carried out by computing the dot product of the kernel and the input area where the kernel is overlapped. To facilitate the learning of kernel weights, the convolution layer's output is summed with a bias term and then fed to a non-linear activation function. Activation Functions are generally non-linear functions like Sigmoid and Rectified Linear Unit (RLU). Depending on the nature of data and classification tasks, these activation functions are selected accordingly [32]. Rectified Linear units are known to have more biological representation, neurons in the brain either fire or don't fire. As a result, it yields favorable results for image recognition tasks as it is less prone to the vanishing gradient problem and it produces sparser, more efficient representations [33].

Advantages of Traditional Computer Vision Techniques

This section is all about where traditional feature-based approaches are useful in improving performance in computer vision tasks.

- Scale Invariant Feature Transform (SIFT)
- Speeded Up Robust Features (SURF)
- Features from Accelerated Segment Test (FAST)
- Hough transforms
- Geometric hashing

Feature descriptors such as SIFT and SURF are generally combined with traditional machine learning classification algorithms such as Support Vector Machines and KNearest Neighbours to

solve the computer vision problems. DL is sometimes overkilling, as often traditional CV techniques can solve a problem much more efficiently and in fewer lines of code than deep learning. Algorithms like SIFT and even simple color thresholding and pixel counting algorithms are not class-specific, they are very general and perform the same for any image. In contrast, features learned from a deep neural net are specific to your training dataset which, if not well modeled, probably won't perform well for images different from the training set. Therefore, SIFT and other algorithms are often used for applications such as image-stitching mesh reconstruction which don't require specific class knowledge. These tasks are attainable by training large datasets; however, this requires a huge research effort and it is not practical to go through this effort for a closed application. One needs to practice common sense when it comes to choosing which route to take for a given CV application. However, the same can be achieved by using simple color thresholding. Some problems can be tackled with simpler and faster techniques. What if a deep neural network works poorly outside of the training data? If the training dataset is limited, then the machine may overfit the training data and not be able to generalize for the task. It would be too difficult to manually twist the parameters of the model because a DNN has large numbers of parameters inside of it each with complex interrelationships. In this way, DL models are the black box [5]. In a traditional CV, one can judge whether your solution will work outside of a training environment. The CV engineer can have insights into a problem that they can transfer to their algorithm and if anything fails, the parameters can be modified to perform well for a wider range of images. Today, the traditional techniques are used when the problem can be simplified so that they can be deployed on low-cost microcontrollers or to limit the problem for deep learning techniques by highlighting certain features in data, augmenting data [19], or aiding in dataset annotation [14].

Finally, there are many more challenging problems in CV such as augmented reality [15], automatic panorama stitching [17], virtual reality [18], 3D modeling [18], motion estimation [18], video stabilization [19], motion capture [18], video processing [19] and scene understanding, which cannot simply be easily implemented in a differentiable manner with deep learning but benefit from solutions using traditional techniques.

Dataset Annotation and Augmentation

There are arguments against the combination of CV and DL and they summarize the conclusion that we need to re-evaluate our methods from rule-based to data-driven. Traditionally, from the perspective of signal processing, we know the operational connotations of CV algorithms such as SIFT and SURF methods, but DL leads such meaning nowhere, all you need is more data. This can be seen as a huge step forward but maybe also a backward move. Some of the pros and cons of each side of this debate have been discussed already in this paper, however, if future methods are to be purely data-driven then focus should be placed on more intelligent methods for dataset creation. The fundamental problem of current research is that there is no longer enough data for advanced algorithms or models for special applications. Coupling custom datasets and DL models will be the future theme of many research papers. So many researchers' outputs consist of not only algorithms or architectures, but also datasets or methods to amass data. Dataset annotation is a major bottleneck in the DL workflow which requires many hours of manual labeling. Nowhere is this more problematic than in semantic segmentation applications where

every pixel needs to be annotated accurately. There are many useful tools available to semi-automate the process as reviewed by [14], many of which take advantage of algorithmic approaches such as ORB features [1], polygon morphing [22], semi-automatic Area of Interest (AOI) fitting [55] and all of the above [22]. The easiest and most common method to overcome limited datasets and reduce the overfitting of deep learning models for image classification is to artificially enlarge the dataset using label-preserving transformations. This process is known as dataset augmentation and it involves the artificial generation of extra training data from the available ones, for example, by cropping, scaling, or rotating images [23]. It is desirable for data augmentation procedures to require very little computation and to be implementable within the DL training pipeline so that the transformed images do not need to be stored on a disk. Traditional algorithmic approaches that have been employed for dataset augmentation include Principle Component Analysis (PCA) [1], adding noise, interpolating or extrapolating between samples in feature space [24], and modeling the visual context surrounding objects from segmentation annotations [25]

Challenges of Deep Learning

There are some challenges introduced. Deep learning works better on large datasets for substantial accuracy; however, It involves billions of additional math operations and needs large processing power. DL requires these computing resources for training and to a lesser extent for inference. It is essential to have dedicated hardware. DL is also dependent on image resolution. Achieving adequate performance in object classification, for example, requires high-resolution images or video with the increase in the amount of data that needs to be processed, stored, and transferred. Image resolution is especially important for applications in which it is necessary to detect and classify objects in the distance range video. The frame reduction techniques such as using SIFT features [24] or optical flow for moving objects [26] to first identify a region of interest are useful for image resolution and also to reduce the time and data required for training. Deep learning needs big data. Often millions of data records are required. When computing facilities are unavailable, traditional computer vision technique methods will come into play. Training a DNN takes a very long time. Depending on computing hardware availability, training can take a matter of hours or days. Moreover, training for any given application often requires many iterations as it entails trial and error with different training parameters. Transfer learning is the most common technique to reduce training time [26]. The algorithm can be used to speed up convolutions as demonstrated by [27, 28] and hence may again become of major importance. However, it must be said that easier, more domain-specific tasks than general image classification will not require as much data in the order of hundreds or thousands rather than millions. This is still a considerable amount of data and CV techniques are often used to increase training data through data augmentation or reduce the data down to a particular type of feature through other pre-processing steps. Pre-processing involves transforming the data generally with traditional CV techniques to allow relationships/patterns to be more easily interpreted before training your model. Data augmentation is a common pre-processing task that is used when there is limited training data. It involves performing shifts, shears random rotations, on the images in your training set to effectively increase the number of training images [29]. Another approach is

highlighting the features of interest before passing the data into a CNN with CV-based methods such as background subtraction and segmentation [33]

Conclusions

From the literature we find that many traditional computers vision technique are obsolete because of deep leaning technique. By utilizing traditional computer vision technique and deep learning technique we can make efficient and cost-effective model for unmanned ground vehicles. DL performance can be improved by traditional computer vision technique in a wide range of applications from reducing training time, processing and data requirements to being applied in emerging fields such as SLAM, Panoramic stitching, Geometric Deep Learning and 3D vision where DL is not developed so much.

Object detection and recognition are considered to be one of the most important tasks as this is what helps the vehicle detect obstacles and set the future courses of the vehicle [14]. Therefore, the object detection algorithms must be highly accurate. Though there are many machine learning and deep learning algorithms for object detection and recognition, such as Support vector machine (SVM), Convolutional Neural Networks (CNNs), Regional Convolutional Neural Networks (R-CNNs), YOLO model, etc., it is important to choose the right algorithm for autonomous driving as it requires real-time object detection and recognition. Since machines cannot detect the objects in an image instantly like humans, it is really necessary for the algorithms to be fast and accurate and to detect the objects in real-time [8], so that the vehicle controllers solve optimization problems at least at a frequency of one per second [14]

YOLOv3, Tiny-YOLOv3 and Faster R-CNN have been identified as the most suitable and efficient deep-learning models to perform real-time object detection and recognition for unmanned ground vehicle.

References

[1] Niall O' Mahony (Institute of Technology Tralee), Sean Campbell (Institute of Technology Tralee), Lenka Krpalkova (Institute of Technology Tralee), et al (2018) Deep Learning for Visual Navigation of Unmanned Ground Vehicles; A review

[2] Karami E, Prasad S, Shehata M Image Matching Using SIFT, SURF, BRIEF, and ORB: Performance Comparison for Distorted Images

[3] Adit Deshpande A Beginner's Guide To Understanding Convolutional Neural Networks – Adit Deshpande – CS Undergrad at UCLA ('19). https://adeshpande3.github.io/ABeginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/. Accessed 19 Jul 2018

[4] Tsai FCD (1994) Geometric hashing with line features. Pattern Recognit 27:377–389. https://doi.org/10.1016/0031-3203(94)90115-5

[5] Angelina M, Gim U, Lee H PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition

[6] Campo Seco F, Cohen A, Pollefeys M, Sattler T Hybrid scene Compression for Visual Localization

[7] Loghmani MR, Planamente M, Caputo B, Vincze M Recurrent Convolutional Fusion for RGB-D Object Recognition

[8] Clément M, Kurtz C, Wendling L (2018) Learning spatial relations and shapes for structural object description and scene recognition. Pattern Recognit 84:197–210. https://doi.org/10.1016/J.PATCOG.2018.06.017

[9] Bonaccorso G (2018) Machine Learning Algorithms Popular Algorithms for Data Science and Machine Learning, 2nd Edition. Packt Publishing Ltd

[10] Mahony NO, Murphy T, Panduru K, et al (2017) Improving controller performance in a powder blending process using predictive control. In: 2017 28th Irish Signals and Systems Conference (ISSC). IEEE, pp 1–6

[11] O'Mahony N, Murphy T, Panduru K, et al (2017) Real-time monitoring of powder blend composition using near-infrared spectroscopy. In: 2017 Eleventh International Conference on Sensing Technology (ICST). IEEE, pp 1–6

[12] O' Mahony N, Murphy T, Panduru K, et al (2016) Adaptive process control and sensor fusion for process analytical technology. In: 2016 27th Irish Signals and Systems Conference (ISSC). IEEE, pp 1–6

[13] Koehn P, Koehn P (1994) Combining Genetic Algorithms and Neural Networks: The Encoding Problem.

[14] Schöning J, Faion P, Heidemann G (2016) Pixel-wise Ground Truth Annotation in Videos - An Semi-automatic Approach for Pixel-wise and Semantic Object Annotation. In: Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods. SCITEPRESS - Science and Technology Publications, pp 690–697

[15] Alhaija HA, Mustikovela SK, Mescheder L, et al (2017) Augmented Reality Meets Computer Vision : Efficient Data Generation for Urban Driving Scenes

[16] Meneghetti G, Danelljan M, Felsberg M, Nordberg K (2015) Image Alignment for Panorama Stitching in Sparsely Structured Environments. Springer, Cham, pp 428–439

[17] Alldieck T, Kassubeck M, Magnor M (2017) Optical Flow-based 3D Human Motion Estimation from Monocular Video

[18] Zhang X, Lee J-Y, Sunkavalli K, Wang Z (2017) Photometric Stabilization for Fastforward Videos

[19] Schöning J, Faion P, Heidemann G (2016) Pixel-wise Ground Truth Annotation in Videos - An Semi-automatic Approach for Pixel-wise and Semantic Object Annotation. In: Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods. SCITEPRESS - Science and Technology Publications, pp 690–697

[20] Ioannidou A, Chatzilari E, Nikolopoulos S, Kompatsiaris I (2017) Deep Learning Advances in Computer Vision with 3D Data. ACM Comput Surv 50:1–38. https://doi.org/10.1145/3042064

[21] Devries T, Taylor GW (2017) Dataset Augmentation in Feature Space. arXiv Prepr arXiv 170205538v1

[22] Dvornik N, Mairal J, Schmid C Modeling Visual Context is Key to Augmenting Object Detection Dataset

[23] Zheng L, Yang Y, Tian Q SIFT Meets CNN: A Decade Survey of Instance Retrieval

[24] Ng H-W, Nguyen D, Vonikakis V, Winkler S Deep Learning for Emotion Recognition on Small Datasets Using Transfer Learning. https://doi.org/10.1145/2818346.2830593

[25] AlDahoul N, Md Sabri AQ, Mansoor AM (2018) Real-Time Human Detection for Aerial Captured Video Sequences via Deep Models. Comput Intell Neurosci 2018:1– 14. https://doi.org/10.1155/2018/1639561

[26] CS231n Convolutional Neural Networks for Visual Recognition. http://cs231n.github.io/transfer-learning/. Accessed 9 Mar 2018

[27] Highlander TC Efficient Training of Small Kernel Convolutional Neural Networks using Fast Fourier Transform

[28] Highlander T, Rodriguez A (2016) Very Efficient Training of Convolutional Neural Networks using Fast Fourier Transform and Overlap-and-Add

[29] Wang J, Perez L The Effectiveness of Data Augmentation in Image Classification using Deep Learning

[30] Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. NIPS'12 Proc 25th Int Conf Neural Inf Process Syst 1:1097–1105

[31] Nash W, Drummond T, Birbilis N (2018) A Review of Deep Learning in the Study of Materials Degradation. NJ Mater Degrad 2:37. https://doi.org/10.1038/s41529-018- 0058-x

[32] Hayou S, Doucet A, Rousseau J (2018) On The Selection of Initialization and Activation Function for Deep Neural Networks. arXiv Prepr arXiv 180508266v2

[33] Adit Deshpande A Beginner's Guide To Understanding Convolutional Neural Networks – Adit Deshpande – CS Undergrad at UCLA ('19). https://adeshpande3.github.io/ABeginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/. Accessed 19 Jul 2018