A Project Report

on

VEHICLE OBJECT DETECTION

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

Under The Supervision of
Dr. Vipin Rai
Associate Professor

Submitted By

Anshuman Singh
18SCSE1010329

Mihir Raj Singh
18SCSE1010318

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA

DECEMBER,2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled **"VEHICLE OBJECT DETECTION"** in partial fulfillment of the requirements for the award of the Bachelor of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September, 2021 to November,2021 under the supervision of **Dr. Vipin Rai** Associate Professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

ANSHUMAN SINGH(18SCSE1010329)

MIHIR RAJ SINGH(18SCSE1010318)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Vipin Rai

Associate Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of **Anshuman Singh (18SCSE1010329) & Mihir Raj Singh (18SCSE1010318)** has been held on _____ and his/her work is recommended for the award of Bachelor of Technology in Computer Science and Engineering.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

ACKNOWLEDGEMENT

Apart from the efforts of our, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show my greatest appreciation to "**Dr. Vipin Rai"** and also our dean "**Dr. Munish Sabarwal"**. We can't say thank you enough for his tremendous support and help. We feel motivated and encouraged every time. We attend his meeting. Without his encouragement and guidance this project would not have materialized.

The guidance and support received from all the members who contributed and who are contributing to this project, was vital for the success of the project. We are grateful for their constant support and help.

Abstract

Detection and counting of automobiles are becoming more important inside the field of toll road control. The detection and counting of motors have significant role in a shrewd transportation device, especially for site visitors' management. Traffic problems has become a largest problem in which city planners going through for years. The detection of moving automobiles extra accurately, numerous laptop vision strategies, vehicle counting is done by way of the usage of a digital detection area. Traffic evaluation will calculate for the variety of automobiles in an area consistent with a few arbitrary time periods and classify the vehicles. But the moving automobiles and their detection, monitoring, and counting are very crucial for drift of site visitors in monitoring, planning, and controlling. By studying the recorded video of site visitors waft sequence from a digital camera, a video- based solution era is applied in aggregate with digital detector and blob tracking technologies and YOLO are necessary. With this era, we carried out Open CV for real time video applications. Such strategies assist us to come across, music, rely and classify the shifting cars. However, because of the incredible size of cars, their discovery remains a task that once and for all contributes to the accuracy of vehicle statistics. To address this challenge, this paper proposes a calculator based on a car vision.

Keywords—Image segmentation, Vehicle dataset, Vehicle counting, Classification and Vehicle detection

List of Figure

Table of Contents

1. Introduction:

A few years ago, the creation of the software and hardware image processing systems was mainly limited to the development of the user interface, which most of the programmers of each firm were engaged in. The situation has been significantly changed with the advent of the Windows operating system when the majority of the developers switched to solving the problems of image processing itself. However, this has not yet led to the cardinal progress in solving typical tasks of recognizing faces, car numbers, road signs, analyzing remote and medical images, etc. Each of these "eternal" problems is solved by trial and error by the efforts of numerous groups of the engineers and scientists. As modern technical solutions are turn out to be excessively expensive, the task of automating the creation of the software tools for solving intellectual problems is formulated and intensively solved abroad. In the field of image processing, the required tool kit should be supporting the analysis and recognition of images of previously unknown content and ensure the effective development of applications by ordinary programmers. Just as the Windows toolkit supports the creation of interfaces for solving various applied problems.

Object recognition is to describe a collection of related computer vision tasks that involve activities like identifying objects in digital photographs. Image classification involves activities such as predicting the class of one object in an image. Object localization is refers to identifying the location of one or more objects in an image and drawing an abounding box around their extent. Object detection does the work of combines these two tasks and localizes and classifies one or more objects in an image. When a user or practitioner refers to the term “object recognition“, they often mean “object detection“. It may be challenging for beginners to distinguish between different related computer vision tasks.

So, we can distinguish between these three computer vision tasks with this example:

Image Classification: This is done by Predict the type or class of an object in an image.

Input: An image which consists of a single object, such as a photograph.

Output: A class label (e.g. one or more integers that are mapped to class labels).

Object Localization: This is done through, Locate the presence of objects in an image and indicate their location with a bounding box.

Input: An image which consists of one or more objects, such as a photograph.

Output: One or more bounding boxes (e.g. defined by a point, width, and height).

Object Detection: This is done through, Locate the presence of objects with a bounding box and types or classes of the located objects in an image.

Input: An image which consists of one or more objects, such as a photograph.

Output: One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box.

One of the further extension to this breakdown of computer vision tasks is object segmentation, also called “object instance segmentation” or “semantic segmentation,” where instances of recognized objects are indicated by highlighting the specific pixels of the object instead of a coarse bounding box. From this breakdown, we can understand that object recognition refers to a suite of challenging computer vision tasks.

For example, image classification is simply straight forward, but the differences between object localization and object detection can be confusing, especially when all three tasks may be just as equally referred to as object recognition.

Humans can detect and identify objects present in an image. The human visual system is fast and accurate and can also perform complex tasks like identifying multiple objects and detect obstacles with little conscious thought. The availability of large sets of data, faster GPUs, and better algorithms, we can now easily train computers to detect and classify multiple objects within an image with high accuracy. We need to understand terms such as object detection, object localization, loss function for object detection and localization, and finally explore an object detection algorithm known as “You only look once” (YOLO).

Image classification also involves assigning a class label to an image, whereas object localization involves drawing a bounding box around one or more objects in an image. Object detection is always more challenging and combines these two tasks and draws a bounding box around each object of

interest in the image and assigns them a class label. Together, all these problems are referred to as object recognition.

Object recognition refers to a collection of related tasks for identifying objects in digital photographs. Region-based Convolutional Neural Networks, or R-CNNs, is a family of techniques for addressing object localization and recognition tasks, designed for model performance. You Only Look Once, or YOLO is known as the second family of techniques for object recognition designed for speed and real-time use.

OPEN CV

OpenCV is the primary open supply library for laptop vision, picture handling and AI, and now includes GPU acceleration for synchronal operation. OpenCV supports lots of algorithms related to PC Vision and AI and it's far extending step-by way of-step. OpenCV turned into meant for computational productiveness and with an immersed attention on actual-time packages. OpenCV-python supports libraries like NumPy, utilils, etc. And also gives MATLAB- fashion index.

OpenCV does some programs like avenue video photo stitching, Robot and driving force-less car navigation and control, medical photograph analysis and Video/photograph seek and healing so on.

COCO DATASET

COCO is a sizable scope object reputation, segmentation, and captioning dataset. COCO has few highlights:

Object Segmentation:

Image department is the manner of partitioning picture into various segments (preparations of pixels, additionally referred to as photo gadgets). Image department is continuously used to discover objects and obstacles (lines, curves, and so on.) in pics. Image department is the technique of dispensing a label to

every pixel in a photo to such an extent that pixels with the similar label share a number of the attributes.The effect of picture division is a fixed of fragments that each one in unfold the complete image or a hard and fast of shapes extracted from the picture (see edge reputation). Nearby areas are fundamentally special with appreciate to the same qualities.

Recognition in Context:

Context is a fundamental wellspring of information approximately an object's identity, place and scale. Here we gift a simple shape for displaying the relationship amongst context and object residences established upon the relationship among the insights of low-stage functions over the whole scene and the objects that it includes.

Super Pixel Stuff Segmentation:

Super pixel is a meeting of related pixels with equal hues or gray degrees. Super pixel segmentation is partitioning a image into many non-protecting super pixels. There are two sizeable blessings for making use of great pixels they are you could parent functions on more important areas and you may limit the enter factors for the following calculations. Recurrent Neural Network and see how it performs on the categorical data.

2. Background:

The aim of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image. Generally, only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur and that need to somehow be explored. Each detection of the image is reported with some form of pose information. This is as simple as the location of the object, a location and scale, or the extent of the object defined in terms of a bounding box. In some other situations, the pose information is more detailed and contains the parameters of a linear or non-linear transformation. For example for face detection in a face detector may compute the locations of the eyes, nose and mouth, in addition to the bounding box of the face. An example of a bicycle detection in an image that specifies the locations of certain parts is shown in Figure 1. The pose can also be defined by a three-dimensional transformation specifying the location of the object relative to the camera. Object detection systems always construct a model for an object class from a set of training examples. In the case of a fixed rigid object in an image, only one example may be needed, but more generally multiple training examples are necessary to capture certain aspects of class variability

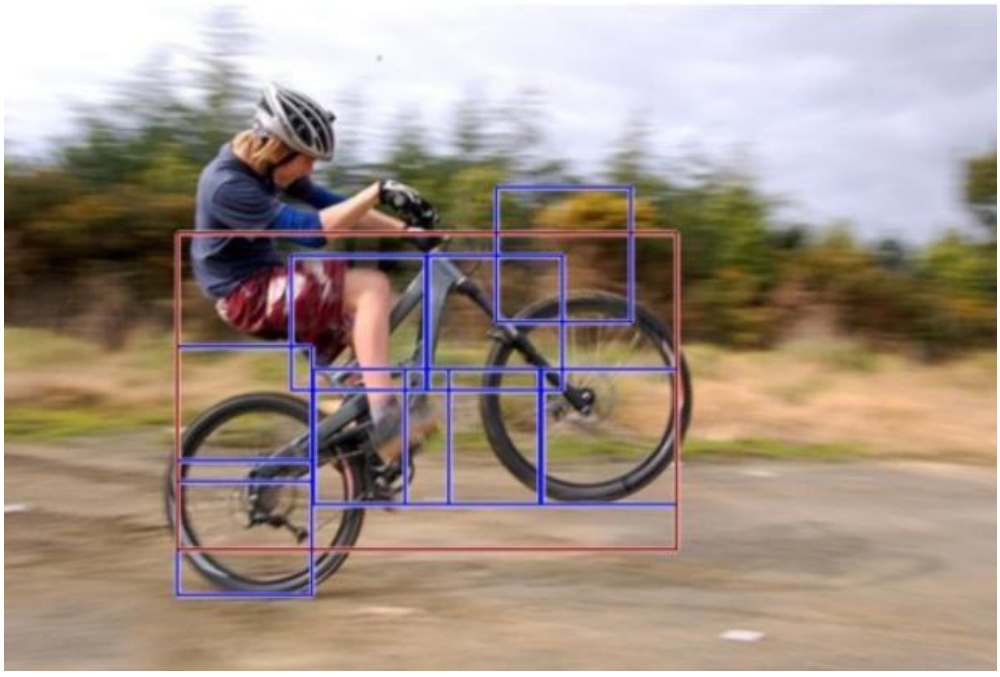

Figure 1

Convolutional implementation of the sliding windows Before we discuss the implementation of the sliding window using convents, let us analyze how we can convert the fully connected layers of the network into convolutional layers. Fig. 2 shows a simple convolutional network with two fully connected layers each of shape .

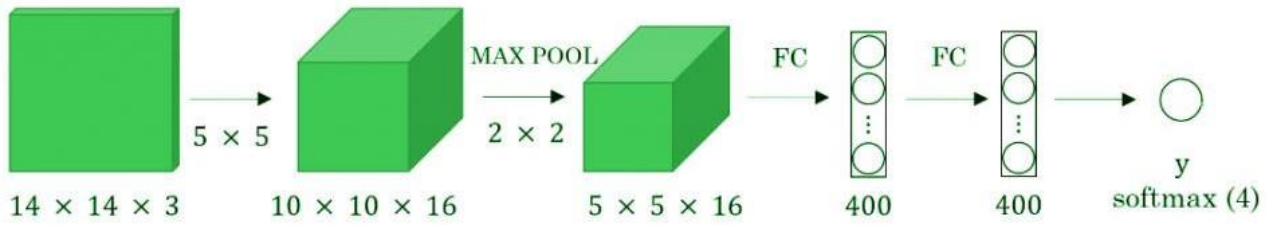

Figure 2: simple convolution network

A fully connected layer can be converted to a convolutional layer with the help of a 1D convolutional layer. The width and height of this layer is equal to one and the number of filters are equal to the shape of the fully connected layer. An example of this is shown in Fig 3.

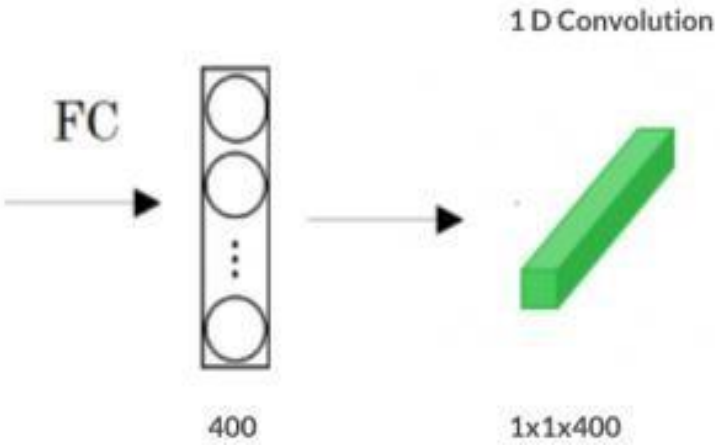

Figure 3

We can apply the concept of conversion of a fully connected layer into a convolutional layer to the model by replacing the fully connected layer with a 1-D convolutional layer. The number of filters of the 1D convolutional layer is equal to the shape of the fully connected layer. This representation is shown in Fig 4. Also, the output softmax layer is also a convolutional layer of shape (1, 1, 4), where 4 is the number of classes to predict.

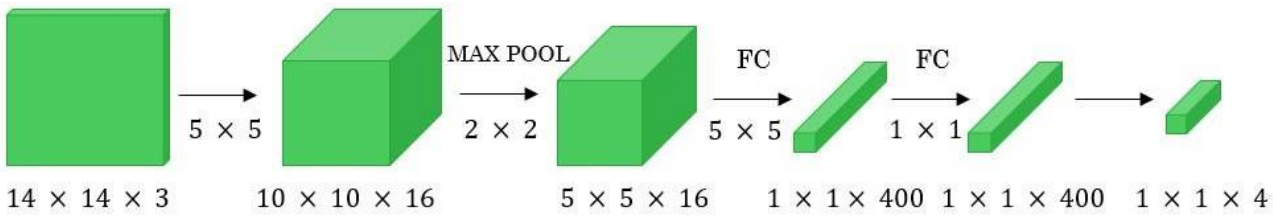

Figure 4

Now, let's extend the above approach to implement a convolutional version of the sliding window. First, let us consider the ConvNet that we have trained to be in the following representation (no fully connected layers).

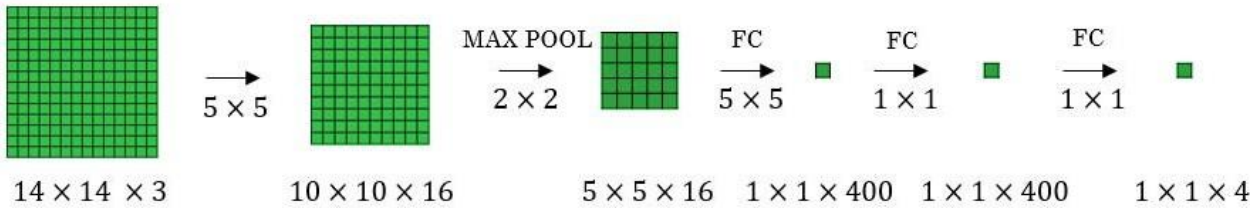

Figure 5

Let's assume the size of the input image to be 16 × 16 × 3. If we are using the sliding window approach, then we would have passed this image to the above ConvNet four times, where each time the sliding window crops the part of the input image matrix of size 14 × 14 × 3 and pass it through the ConvNet. But instead of this, we feed the full image (with shape 16 × 16 × 3) directly into the trained ConvNet (see Fig. 6). This results will give an output matrix of shape 2 × 2 × 4. Each cell in the output matrix represents the result of the possible crop and the classified value of the cropped image. For example, the left cell of the output matrix(the green one) in Fig. 6 represents the result of the first sliding window. The other cells in the matrix represent the results of the remaining sliding window operations.

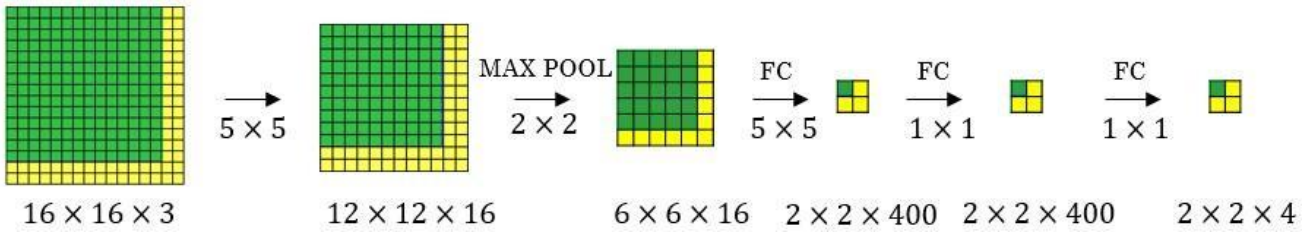

Figure 6

The stride of the sliding window is decided by the number of filters used in the Max Pool layer. In the example above, the Max Pool layer has two filters, and for the result, the sliding window moves with a stride of two resulting in four possible outputs to the given input. The main
advantage of using this technique is that the sliding window runs and computes all values simultaneously. Consequently, this technique is really fast. The weakness of this technique is that the position of the bounding boxes is not very accurate.

A better algorithm that tackles the issue of predicting accurate bounding boxes while using the convolutional sliding window technique is the YOLO algorithm. YOLO stands for you only look

once which was developed in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. It is popular because it achieves high accuracy while running in real-time. This algorithm requires only one forward propagation pass through the network to make the predictions.

This algorithm divides the image into grids and then runs the image classification and localization algorithm (discussed under object localization) on each of the grid cells. For example, we can give an input image of size 256 × 256. We place a 3 × 3 grid on the image (see Fig. 7).

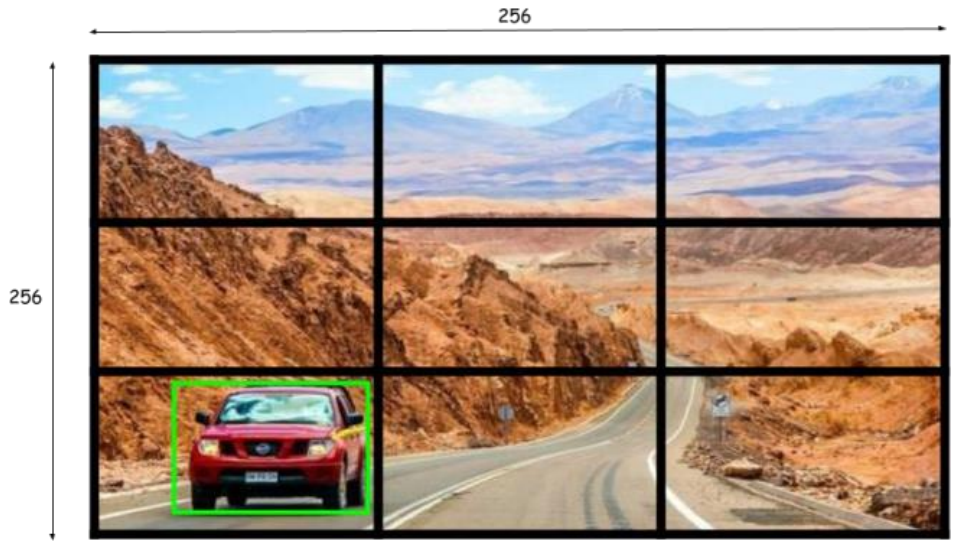

Figure 7

Next, we shall apply the image classification and localization algorithm on each grid cell.In the image each grid cell, the target variable is defined as

Yi,j=[pcbxbybhbwc1c2c3c4]T(6)

Do everything once with the convolution sliding window. Since the shape of the target variable for each grid cell in the image is 1 × 9 and there are 9 (3 × 3) grid cells, the final output of the model will be:

$$Final\ Output = \underbrace{3 \times 3}_{\text{Number of grid cells}} \times \underbrace{9}_{\text{Output label for each grid cell}}$$

The advantages of the YOLO algorithm is that it is very fast and predicts much more accurate bounding boxes. Also, in practice to get the more accurate predictions, we use a much finer grid, say 19 × 19, in which case the target output is of the shape 19 × 19 × 9.

3. LITERATURE SURVEY:

In various fields, there is a necessity to detect the target object and also track them effectively while handling occlusions and other included complexities. Many researchers (Almeida and Guting 2004, Hsiao-Ping Tsai 2011, Nicolas Papadakis and Aure lie Bugeau 2010) attempted for various approaches in object tracking. The nature of the techniques largely depends on the application domain. Some of the research works which made the evolution to proposed work in the field of object tracking are depicted as follows.

3.1 OBJECT DETECTION

Object detection is an important task, yet challenging vision task. It is a critical part of many applications such as image search, image auto-annotation and scene understanding, object tracking. Moving object tracking of video image sequences was one of the most important subjects in computer vision. It had already been applied in many computer vision fields, such as smart video surveillance (Arun Hampapur 2005), artificial intelligence, military guidance, safety detection and robot navigation, medical and biological application. In recent years, a number of successful single-object tracking system appeared, but in the presence of several objects, object detection becomes difficult and when objects are fully or partially occluded, they are obtruded from the human vision which further increases the problem of detection. Decreasing illumination and acquisition angle. The proposed MLP based object tracking system is made robust by an optimum selection of unique features and also by implementing the Adaboost strong classification method.

3.1.1 Background Subtraction

The background subtraction method by Horprasert et al (1999), was able to cope with local illumination changes, such as shadows and highlights, even globe illumination changes. In this method, the background model was statistically modelled on each pixel. Computational colour mode, include the brightness distortion and the chromaticity distortion which was used to distinguish shading background from the ordinary background or moving foreground objects. The background and foreground subtraction method used the following approach. A pixel was modelled by a 4-tuple [Ei, si, ai, bi], where Ei- a vector with expected colour value, si - a vector with the standard deviation of colour value, ai - the variation of the brightness distortion and bi was the variation of the chromaticity distortion of the ith pixel. In the next step, the difference between the background image and the current image was evaluated. Each pixel was finally classified into four categories: original

background, shaded background or shadow, highlighted background and moving foreground object. Liyuan Li et al (2003), contributed a method for detecting foreground objects in non-stationary complex environments containing moving background objects. A Bayes decision rule was used for classification of background and foreground changes based on inter-frame colour co-occurrence statistics. An approach to store and fast retrieve colour cooccurrence statistics was also established. In this method, foreground objects were detected in two steps. First, both the foreground and the background changes are extracted using background subtraction and temporal differencing. The frequent background changes were then recognized using the Bayes decision rule based on the learned colour co-occurrence statistics. Both short-term and long term strategies to learn the frequent background changes were used. An algorithm focused on obtaining the stationary foreground regions as said by Álvaro Bayona et al (2010), which was useful for applications like the detection of abandoned/stolen objects and parked vehicles. This algorithm mainly used two steps. Firstly, a sub-sampling scheme based on background subtraction techniques was implemented to obtain stationary foreground regions. This detects foreground changes at different time instants in the same pixel locations. This was done by using a Gaussian distribution function. Secondly, some modifications were introduced on this base algorithm such as thresh holding the previously computed subtraction. The main purpose of this algorithm was reducing the amount of stationary foreground detected.

3.1.2 Template Matching

Template Matching is the technique of finding small parts of an image which match a template image. It slides the template from the top left to the bottom right of the image and compares for the best match with the template. The template dimension should be equal to the reference image or smaller than the reference image. It recognizes the segment with the highest correlation as the target. Given an image S and an image T, where the dimension of S was both larger than T, output whether S contains a subset image I where I and T are suitably similar in pattern and if such I exists, output the location of I in S as in Hager and Bellhumear (1998). Schweitzer et al (2011), derived an algorithm which used both upper and lowers bound to detect ‘k’ best matches. Euclidean distance and Walsh transform kernels are used to calculate match measure. The positive things included the usage of priority queue improved quality of decision as to which bound-improved and when good matches exist inherent cost was dominant and it improved performance. But there were constraints like the absence of good matches that lead to queue cost and the arithmetic operation cost was higher. The proposed methods dint use queue thereby avoiding the queue cost rather used template matching. Visual tracking methods can be roughly categorized in two ways namely, the feature-based and region-based method as proposed by Ken Ito and Shigeyuki Sakane (2001). The feature-based approach estimates the 3D pose of a target object to fit the image features the edges, given a 3D

geometrical model of an object. This method requires much computational cost. Region-based can be classified into two categories namely, parametric method and view-based method. The parametric method assumes a parametric model of the images in the target image and calculates optimal fitting of the model to pixel data in a region. The view-based method was used to find the best match of a region in a search area given the reference template. This has the advantage that it does not require much computational complexity as in the feature-based approach.

4. Existing Methods:

4.1 ResNet

To train the network model in a more effective manner, we herein adopt the same strategy as that used for DSSD (the performance of the residual network is better than that of the VGG network). The goal is to improve accuracy. However, the first implemented for the modification was the replacement of the VGG network which is used in the original SSD with ResNet. We will also add a series of convolution feature layers at the end of the underlying network. These feature layers will gradually be reduced in size that allowed prediction of the detection results on multiple scales. When the input size is given as 300 and 320, although the ResNet–101 layer is deeper than the VGG–16 layer, it is experimentally known that it replaces the SSD's underlying convolution network with a residual network, and it does not improve its accuracy but rather decreases it.

4.2 R-CNN

To circumvent the problem of selecting a huge number of regions, Ross Girshick et al. proposed a method where we use the selective search for extract just 2000 regions from the image and he called them region proposals. Therefore, instead of trying to classify the huge number of regions, you can just work with 2000 regions. These 2000 region proposals are generated by using the selective search algorithm which is written below.

Selective Search:

1. Generate the initial sub-segmentation, we generate many candidate regions
2. Use the greedy algorithm to recursively combine similar regions into larger ones
3. Use generated regions to produce the final candidate region proposals

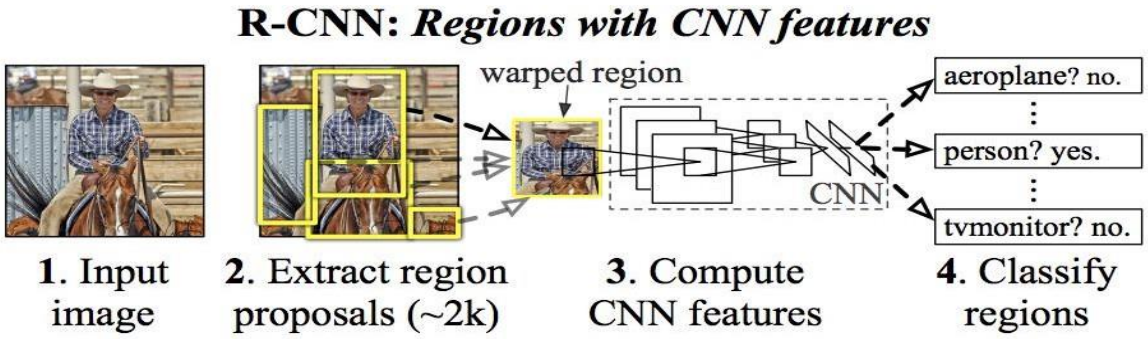

Figure 8

These 2000 candidate regions which are proposals are warped into a square and fed into a convolutional neural network that produces a 4096-dimensional feature vector as output. The CNN plays a role of feature extractor and the output dense layer consists of the features extracted from the image and the extracted features are fed into an SVM for the classify the presence of the object within that candidate region proposal. In addition to predicting the presence of an object within the region proposals, the algorithm also predicts four values which are offset values for increasing the precision of the bounding box. For example, given the region proposal, the algorithm might have predicted the presence of a person but the face of that person within that region proposal could have been cut in half. Therefore, the offset values which is given help in adjusting the bounding box of the region proposal.

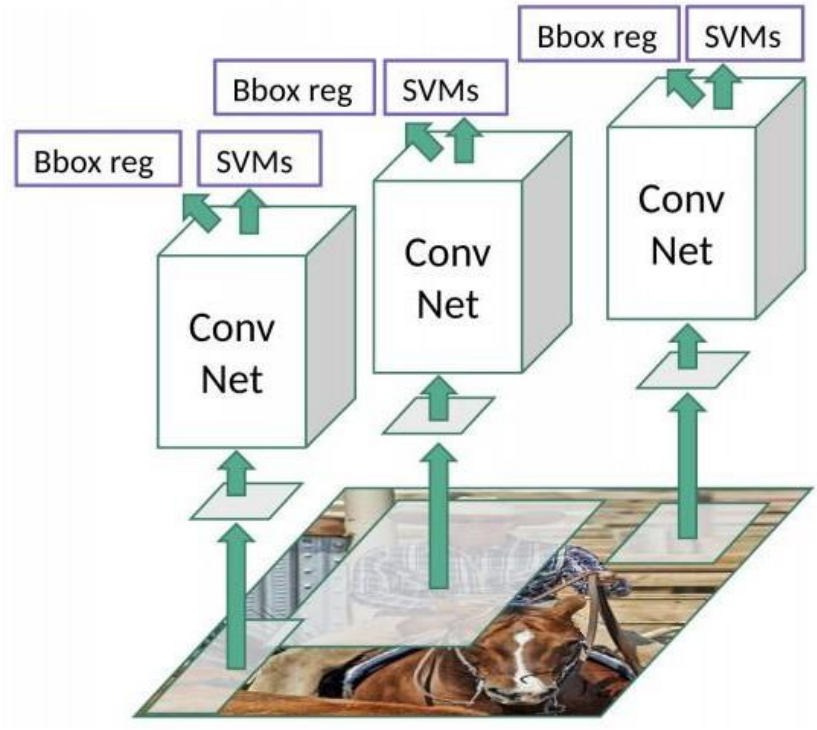

Figure 9: R-CNN

4.2.1 Problems with R-CNN

- It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image.
- It cannot be implemented real time as it takes around 47 seconds for each test image.

- The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals.

4.3 Fast R-CNN

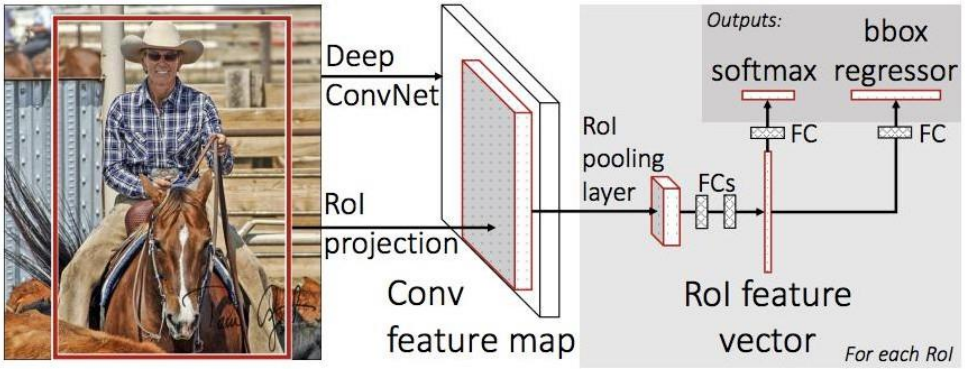

Figure 10: Fast R-CNN

The same author of the previous paper(R-CNN) solved some of the drawbacks of R-CNN to build a faster object detection algorithm and it was called Fast R-CNN. The approach is similar to the R-CNN algorithm. But, instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map. From the convolutional feature map, we can identify the region of the proposals and warp them into the squares and by using an RoI pooling layer we reshape them into the fixed size so that it can be fed into a fully connected layer. From the RoI feature vector, we can use a softmax layer to predict the class of the proposed region and also the offset values for the bounding box.

The reason “Fast R-CNN” is faster than R-CNN is because you don’t have to feed 2000 region proposals to the convolutional neural network every time. Instead, the convolution operation is always done only once per image and a feature map is generated from it.

From the above graphs, you can infer that Fast R-CNN is significantly faster in training and testing sessions over R-CNN. When you look at the performance of Fast R-CNN during testing time, including region proposals slows down the algorithm significantly when compared to not using region proposals. Therefore, the region which is proposals become bottlenecks in Fast R-CNN algorithm affecting its performance.

4.4 Faster R-CNN

Both of the above algorithms(R-CNN & Fast R-CNN) uses selective search to find out the region proposals. Selective search is the slow and time-consuming process which affect the performance of the network.

Similar to Fast R-CNN, the image is provided as an input to a convolutional network which provides a convolutional feature map. Instead of using the selective search algorithm for the feature map to identify the region proposals, a separate network is used to predict the region proposals. The predicted the region which is proposals are then reshaped using an RoI pooling layer which is used to classify the image within the proposed region and predict the offset values for the bounding boxes.

From the above graph, you can see that Faster R-CNN is much faster than it's predecessors. Therefore, it can even be used for real-time object detection.

YOLO works by taking an image and split it into an SxS grid, within each of the grid we take m bounding boxes. For each of the bounding box, the network gives an output a class probability and offset values for the bounding box. The bounding boxes have the class probability above a threshold value is selected and used to locate the object within the image.

YOLO is orders of magnitude faster(45 frames per second) than any other object detection algorithms. The limitation of YOLO algorithm is that it struggles with the small objects within the image, for example, it might have difficulties in identifying a flock of birds. This is due to the spatial constraints .

4.5 SDD :

The SSD object detection composes of 2 parts:

1. Extract feature maps, and
2. Apply convolution filters to detect objects.

SSD uses VGG16 to extract feature maps. Then it detects objects using the Conv4_3 layer. For illustration, we draw the Conv4_3 to be 8 × 8 spatially (it should be 38 × 38). For each cell in the image(also called location), it makes 4 object predictions.

Each prediction composes of a boundary box and 21 scores for each class (one extra class for no object), and we pick the highest score as the class for the bounded object. Conv4_3 makes total of 38 × 38 × 4 predictions: four predictions per cell regardless of the depth of featuremaps. A expected, many predictions contain no object. SSD reserves a class “0” to indicate

SSD does not use the delegated region proposal network. Instead, it resolves to a very simple method. It computes both the location and class scores using small convolution filters. After extraction the feature maps, SSD applies 3 × 3 convolution filters for each cell to make predictions. (These filters compute the results just like the regular CNN filters.) Each filter gives outputs as 25 channels: 21 scores for each class plus one boundary box.

Beginning, we describe the SSD detects objects from a single layer. Actually, it uses multiple layers (multi-scale feature maps) for the detecting objects independently. As CNN reduces the spatial dimension gradually, the resolution of the feature maps also decrease. SSD uses lower resolution layers for the detect larger-scale objects. For example, the 4× 4 feature maps are used for the larger-scale object.

SSD adds 6 more auxiliary convolution layers to image after VGG16. Five of these layers will be added for object detection. In which three of those layers, we make 6 predictions instead of 4. In total, SSD makes 8732 predictions using 6 convolution layers.

Multi-scale feature maps enhance accuracy.The accuracy with different number of feature map layers is used for object detection.

4.6 MANet:

Target detection is fundamental challenging problem for long time and has been a hotspot in the area of computer vision for many years. The purpose and objective of target detection is, to determine if any instances of a specified category of objects exist in an image. If there is an object to be detected in a specific image, target detection return the spatial positions and the spatial extent of the instances of the objects (based on the use a bounding box, for example). As one of cornerstones of image understanding and computer vision,target and object detection forms the basis for more complex and higher-level visual tasks, such as object tracking, image capture, instance segmentation, and others.Target detection is also widely used in areas such as artificial intelligence and information technology, including machine vision, automatic driving vehicles, and human–computer interaction.
In recent times, the method automatic learning of represented features from data based on deep learning has effectively improved performance of target detection. Neural networks are foundation of deep learning. Therefore, design of better neural networks has become an key issue toward improvement of target detection algorithms and performance. Recently developed object detectors that has been based on convolutional neural networks (CNN) has been classified in two types:The first is two-stage detector type, such as Region-Based CNN (R–CNN), Region-Based Full Convolutional Networks (R–FCN), and Feature Pyramid Network (FPN), and the other is single-stage detector, such as the You Only Look Once (YOLO), Single-shot detector (SSD), and the RetinaNet. The former type generates an series of candidate frames as samples of data , and then classifies the samples based on a CNN; the latter type do not generate candidate frames but directly converts the object frame positioning problem into a regression processing problem.

To maintain realtime speeds without sacrificing precision in various object detectors described above, Liu et al proposed the SSD which is faster than YOLO and has a comparable accuracy to that of the most advanced region-based target detectors.SSD combines regression idea of YOLO with the anchor box mechanism of Faster R–CNN, predicts the object region based on the feature maps of the different convolution layers, and outputs discretised multiscale and multi proportional default box coordinates. The convolution kernel predicts frame coordinates compensation of a series of candidate

frames and the confidence of each category. The local feature maps of multiscale area are used to obtain results for each position in the entire image. This maintains the fast characteristics of YOLO algorithm and also ensures that the frame positioning effect is similar to that is induced by the Faster R–CNN. However, SSD directly and independently uses two layers of the backbone VGG16 and four extra layers obtained by a convolution with stride 2 to construct feature pyramid but lacks strong contextual connections.

To solve these problems. A single-stage detection architecture, commonly referred to as MANet, which aggregates feature information at different scales. MANet achievs 82.7% mAP on the PASCAL VOC 2007 test.

5. SYSTEM REQUIREMENT:

Install Python on your computer system

1. Install ImageAI and its dependencies like tensorflow, Numpy,OpenCV, etc.

2. Download the Object Detection model file(Retinanet)

5.1 Steps to be followed :-

1) Download and install Python version 3 from official Python Language website

https://python.org

2) Install the following dependencies via pip:

i. Tensorflow:

Tensorflow is an open-source software library for dataflow and differentiable programming across a range of tasks. It is an symbolic math library, and is also used for machine learning application such as neural networks,etc.. It is used for both research and production by Google.

Tensorflow is developed by the Google Brain team for internal Google use. It is released under the Apache License 2.0 on November 9,2015.

Tensorflow is Google Brain's second-generation system.1st Version of tensorflow was released on February 11, 2017.While the reference implementation runs on single devices, Tensorflow can run on multiple CPU's and GPU (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on various platforms such as64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

The architecture of tensorflow allows the easy deployment of computation across a variety of platforms (CPU's, GPU's, TPU's), and from desktops - clusters of servers to mobile and edge devices.

Tensorflow computations are expressed as stateful dataflow graphs. The name Tensorflow derives from operations that such neural networks perform on multidimensional data arrays, which are referred to as tensors.

pip install tensorflow -command

ii. Numpy:

NumPy is library of Python programming language, adding support for large, multi-dimensional array and matrice, along with large collection of high-level mathematical function to operate over these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several developers. In 2005 Travis Olphant created NumPy by incorporating features of computing Numarray into Numeric, with extension modifications. NumPy is open-source software and has many contributors.

pip install numpy -command

iii. SciPy:

SciPy contain modules for many optimizations, linear algebra, integration, interpolation, special fumction, FFT, signal and image processing, ODE solvers and other tasks common in engineering.

SciPy abstracts majorly on NumPy array object,and is the part of the NumPy stack which include tools like Matplotlib, pandas and SymPy,etc., and an expanding set of scientific computing libraries. This NumPy stack has similar uses to other applications such as MATLAB,Octave, and Scilab. The NumPy stack is also sometimes referred as the SciPy stack.

The SciPy library is currently distributed under BSDlicense, and its development is sponsored and supported by an open communities of developers. It is also supported by NumFOCUS, community foundation for supporting reproducible and accessible science.

pip install scipy -command

iv. OpenCV:

OpenCV is an library of programming functions mainly aimed on real time computer vision. originally developed by Intel, it is later supported by Willow Garage then Itseez. The library is a cross-platform and free to use under the open-source BSD license.

pip install opencv-python -command

v. Pillow:

Python Imaging Library is a free Python programming language library that provides support to open, edit and save several different formats of image files. Windows, Mac OS X and Linux are available for this.

pip install pillow -command

vi. Matplotlib:

Matplotlib is a Python programming language plotting library and its NumPy numerical math extension. It provides an object-oriented API to use general-purpose GUI toolkits such as Tkinter, wxPython, Qt, or GTK+ to embed plots into applications.

pip install matplotlib - command

vii. H5py:

The software h5py includes a high-level and low-level interface for Python's HDF5 library. The low interface expected to be complete wrapping of the HDF5 API, while the high-level component uses established Python and NumPy concepts to support access to HDF5 files, datasets and groups.

A strong emphasis on automatic conversion between Python (Numpy) datatypes and data structures and their HDF5 equivalents vastly simplifies the process of reading and writing data from Python.

pip install h5py

viii. Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

pip install keras

ix. ImageAI:

ImageAI provides API to recognize 1000 different objects in a picture using pre-trained models that were trained on the ImageNet-1000 dataset. The model implementations provided are SqueezeNet, ResNet, InceptionV3 and DenseNet.

pip3 install imageai --upgrade

3) Download the RetinaNet model file that will be used for object detection using following link

https://github.com/OlafenwaMoses/ImageAI/releases/download/1.0/resnet50_coco_best_v2.0.1.h5

Copy the RetinaNet model file and the image you want to detect to the folder that contains the python file.

6. METHODOLOGY:

6.1 SqueezeNet:

SqueezeNet is name of a DNN for computer vision. SqueezNet is developed by researchers at DeepScale, University of California, Berkeley, and Stanford University together. In SqueezeNet design, the authors goal is to create a smaller neural network with few parameters that can more easily fit into memory of computer and can more easily be transmitted over a computer network.

SqueezeNet is originally released in 2016. This original version of SqueezeNet was implemented on top of the Caffe deep learning software framework.The open-source research community ported SqueezeNet to a number of other deep learning frameworks. And is released in additions,in 2016, Eddie Bell released a part of SqueezeNet for the Chainer deep learning framework. in 2016, Guo Haria released a part of SqueezeNet for the Apache MXNet framework. 2016, Tammy Yang released a port of SqueezeNet for the Keras framework. In 2017, companies including Baidu, Xilnx, Imagination Technologies, and Synopsys demostrated SqueezedNet running on low-power processing platforms such as smartphones, FPGAs, and customprocessors.

SqueezeNet ships as part of the source code of a number of deep learning frameworks such as PyTorch, Apache MXNet, and Apple CoreML.In addition, 3rd party developers have created implementation of SqueezeNet that are compatible with frameworks such as TensorFlow. Below is summary of frameworks that support SqueezeNet.

DNN Model	Application	Original Implementation	Other Implementations
SqueezeDet	Object Detection on Images	TensorFlow	Caffe, Keras
SqueezeSeg	Semantic Segmentation of LIDAR	TensorFlow	
SqueezeNext	Image Classification	Caffe	TensorFlow, Keras, PyTorch
SqueezeNAS	Neural Architecture Search for Semantic Segmentation	PyTorch	

Table 1

6.2 InceptionV3:

Inception v3 is widely used as image recognition model that has showed to obtain accuracy of greater than 78.1% on the ImageNet dataset. The model is the culmination of many ideas developed by researchers over years. It is based on “Rethinking the Inception Architecture Computer Vision” by Szegedy

The model is made of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batchnorm is used more throughout the model and applied to activation inputs. Loss is computed via Softmax.

A high-level diagram of the model is shown below:

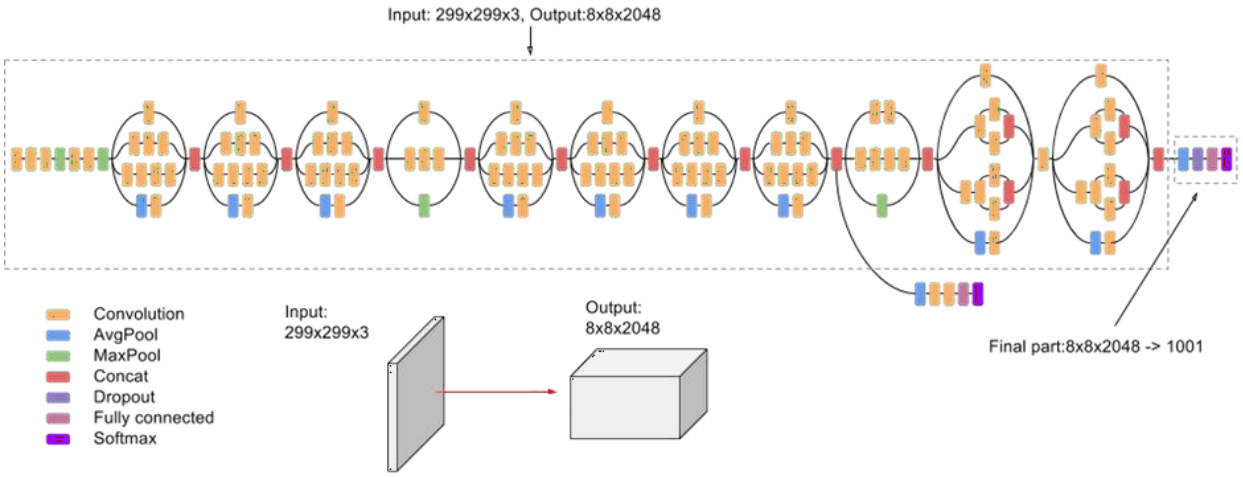

Figure 11

This post assumes past knowledge of neural networks and convolutions.This mainly focus on two topics:

- Why does dense net differs from another convolution networks.
- What are the difficulties during the implementation of DenseNet in tensorflow.

If you know how DenseNets works and interested only in tensorflow implementation feel free to jump to the second chapter or check the source. If you are not familiar with any of these topics to attain knowledge

Compare DenseNet with other convolution networks available

Usually Convolutionnetworks work such a way that
We have an initial image, say having a shape of (29, 34, 31). After we apply set of convolution or pooling filters on it, squeezing dimensions of width and height and increasing features dimension.
So the output from the Li layer is input to the Li+1 layer.

ResNet architecture is proposed for Residual connection, from previous layers to the present layer. input to Li layer is obtain by addition of outputs from previous layers

In contrast, DenseNet paper proposes concatenating outputs from the previous layers instead of using the summation.

So, let's imagine we have an image with shape(28, 28, 3). First, we spread image to initial 24 channels and receive the image (28, 28, 24). Every next convolution layer will generate k=12 features, and remain width and height the same.

The output from L_i layer will be (28, 28, 12).

But input to the L_{i+1} will be (28, 28, 24+12), for L_{i+2} (28, 28, 24 + 12 + 12) and so on.

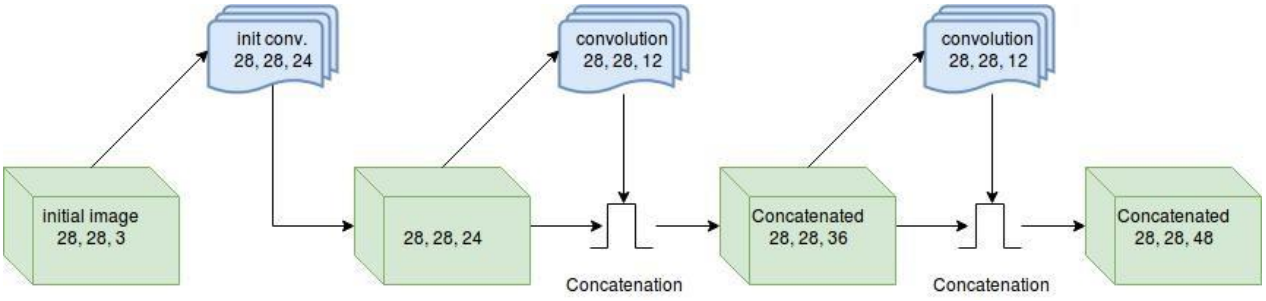

Figure 12: Block of convolution layers with results concatenated

After a while, we receive the image with same width and height, but with plenty of features (28, 28, 48).
All these N layers are named Block in the paper. There's also batch normalization, nonlinearity and dropout inside the block.
To reduce the size, DenseNet uses transition layers. These layers contain convolution with kernel size = 1 followed by 2x2 average pooling with stride = 2. It reduces height and width dimensions but leaves feature dimension the same. As a result, we receive the image with shapes (14, 14, 48).

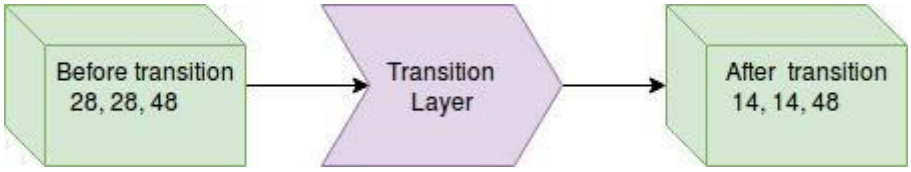

Figure 13: Transition layer

Now we can again pass the image through the block with N convolutions.
With this approach, DenseNet improved a flow of information and gradients throughout the network, which makes them easy to train.
Each layer has direct access to the gradients from the loss function and the original input signal, leading to an implicit deep supervision.

Also at transition layers, not only width and height will be reduced but features also. So if we have image shape after one block (28, 28, 48) after transition layer, we will get (14, 14, 24)

Where theta — some reduction values, in the range (0, 1).

When using bottleneck layers with DenseNet, maximum depth will be divided by 2. It means that if you have 16 3x3 convolution layers with depth 20 previously (some layers are transition layers),you will now have 8 1x1 convolution layers and 8 3x3 convolutions. Last, but not least, about data preprocessing. In the paper per channel normalization was used. With this approach, every image channel should be reduced by its mean and divided by its standard deviation. In many implementations was another normalization used — just divide every image pixel by 255,.

Note about numpy implementation of per channel normalization. By default images provided with data type unit. Before any manipulations, It is advised to convert the images to any float representation. Because otherwise, a code will fail without any warnings or errors.

7. RESULTS

Figure 14

So, these are the output images and videos of our project. In first image there is a background of highway. On the basis of the image, we are successfully detecting and counting the vehicles.

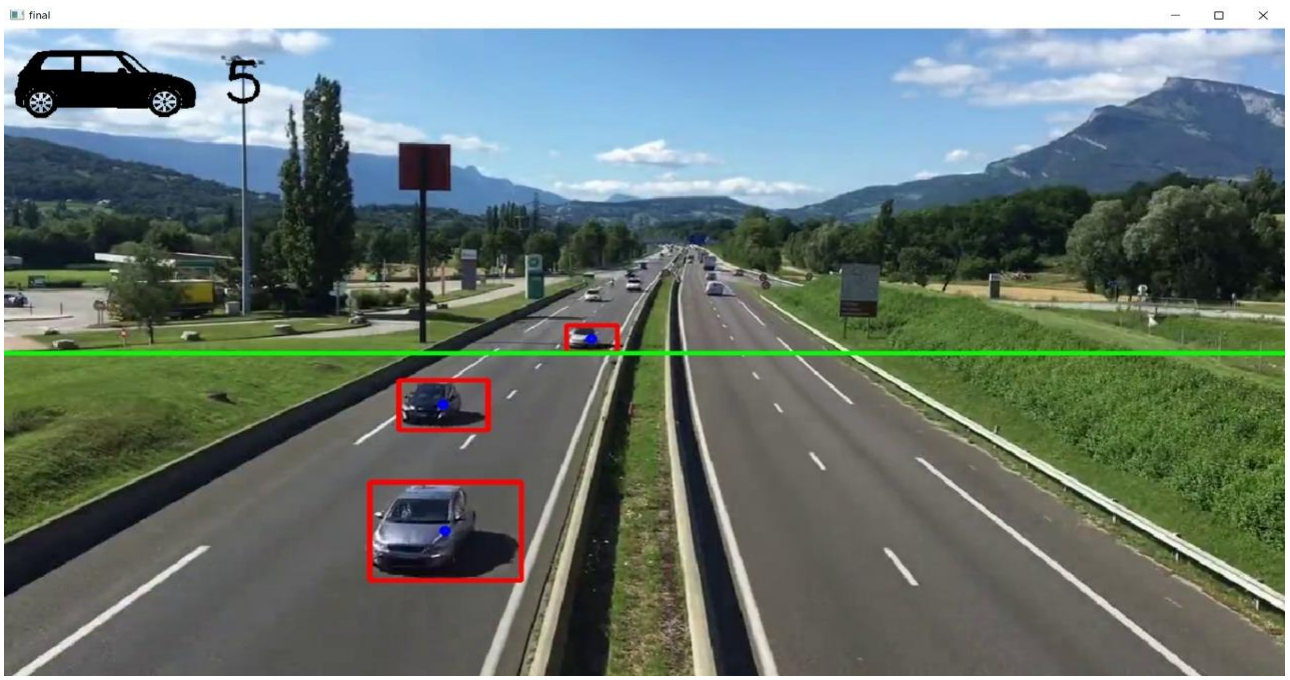

Figure 15

8. CONCLUSION:

A system has been advanced to stumble on and count number dynamic and green cars on highways, streets and closely congested lanes. The device has effective work and knowledge about vehicle be counted and classification with time domain and to perceive the specific vehicles within the presence of any traffic areas and in a number of the heavy congested roads, the heritage is successfully rejected. The experimental outcomes shown have the accuracy of counting automobiles turned into in range 70%-ninety-six%. At lengthy ultimate, we planned a vehicle counting and category version, that may exactly remember the motors with the base set of rules as YOLO educated on COCO dataset. We have used the object label to take gadgets which are labelled as cars, which consist of automobiles, motorbikes, vans, bicycles and other heavy automobiles. The proposed device demonstrates that vehicle counting and classification version for roadways can get over 93% exactness and 25 FPS pace on automobile detection, counting and category of tracked motors.

9. FUTURE SCOPE:

There are several other destiny upgrades ought to be feasible to the device, as an example, detection, monitoring, counting and classification of transferring automobile must be viable on ongoing live movies. So that we will put in force it on the stay motion pictures shooting immediately from the site visitors to music, matter and classification. And some other addition to it may be achieved as speed detection of motors within the visitors in order that it may assist to lessen the injuries and other safety measures can be taken to it by using alarming. This can also help to visitors' improvement and for safety for passengers who journey on motors. Region of hobby (RIO) it can be delivered to the proposed system so which can improve the machine with the aid of choosing the vicinity manually from this we are able to pick the lane which is wanted. From that area most effective we can calculate the automobiles passing thru the tracker line so that high volume of automobiles on that lane may be diverted to other a part of roads. Through this video it'll be helpful to test the vehicle density and accidents befell on that lane this device is helpful in all measures of the visitor's improvement.

10.REFERENCES:

[1] Luo, W., Xing, J., Milan, A., Zhang, X., Liu, W., Zhao, X., Kim, T.K. (2014). Multiple object tracking: A literature review. arXiv preprint arXiv:1409.7618.

[2] Al-Smadi, M., Abdulrahim, K., Salam, R.A. (2016). Traffic surveillance: A review of vision-based vehicle detection, recognition and tracking. *International Journal of Applied Engineering Research*, *11*(1), 713–726.

[3] Radhakrishnan, M. (2013). Video object extraction by using background subtraction techniques for sports applications. *Digital Image Processing*, *5*(9), 91–97.

[4] Qiu-Lin, L.I., & Jia-Feng, H.E. (2011). Vehicle's detection based on three-frame-difference method and cross-entropy threshold method. *Computer Engineering*, *37*(4), 172–174.

[5] Luo, Z. (2018). Traffic analysis of low and ultra-low frame-rate videos, Doctoral dissertation. Université de Sherbrooke.

[6] Geiger, A. (2012). Are we ready for autonomous driving? the kitti vision
benchmark suite, In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3354–3361): IEEE.

[7] Zhe, Z., Liang, D., Zhang, S., Huang, X., Hu, S. (2016). Traffic-sign detection and classification in the wild, In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[8] Nilesh J. Uke and Ravindra C. Thool, "Moving Vehicle Detection for Measuring Traffic Count Using OpenCV", Sinhgad College of Engineering, Department of Information Technology Pune, India, Vol 1, No 4.

[9] Mr. Majeti V N, Hemanth Kumar and Mr. B. Vasanth, "Vehicle Detection, Tracking and Counting Objects for Traffic Surveillance System Using Raspberry-Pi", International Journal of Modern Trends in Engineering and Research (IJMTER) Volume 02, Issue 09, [September – 2015].

[10] Sheeraz Memon, "A Video based Vehicle Detection, Counting and Classification System", Mehran University of Engineering & Technology, Jamshoro, Pakistan, pp. 34-41