**Mini Project Report on**



**CONTENT ANALYSIS**

**YOLO BASED VEHICLE COUNTING APPLICATION**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name University Roll no**

**Jai Anand** **2016792**

***Under the Mentorship of***

**Dr Vikas Tripathi**

**Associate Dean (R&D)**

**Graphic Era Deemed to be University**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

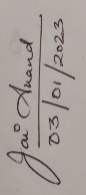
**Dehradun, Uttarakhand**

**January 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Content Analysis- Yolo based Vehicle Counting Application”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Dr Vikas Tripathi, Associate Dean (Research &Development)**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.



Jai Anand 2016792

Name University Roll no

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction | **1-3** |
| Chapter 2 | Literature Survey | **4-6** |
| Chapter 3 | Methodology | **7-9** |
| Chapter 4 | Result and Discussion | **10** |
| Chapter 5 | Future Work | **11** |
|  | References | **12** |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

* 1. **Introduction**

Content analysis is a research tool used to determine the presence of certain words, themes, or concepts within some given qualitative data (i.e., text or image or video). Using content analysis researchers can quantify and analyze the presence of relationships of certain concepts and ideas. Researchers can then make inferences about the content within the files, even the culture and time of surrounding the text.

Humans’ glances at an image or a video and instantly know what objects are in latter. The human visual system is fast and accurate, allowing us to perform complex tasks like driving with little conscious thought. Fast and accurate algorithms for object detection would allow computers to interpret real time information and act accordingly to perform certain tasks. Current detection systems repurpose classifiers to perform detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image.

* 1. **Object Detection Methods**

Generally, object detection methods are classified as either neural network-based or non-neural approaches. Also, some of them are rule-based, where the rule is predefined to match specific objects. Non-neural approaches require defining features using some feature engineering techniques and then using a method such as a support vector machine (SVM) to do the classification.

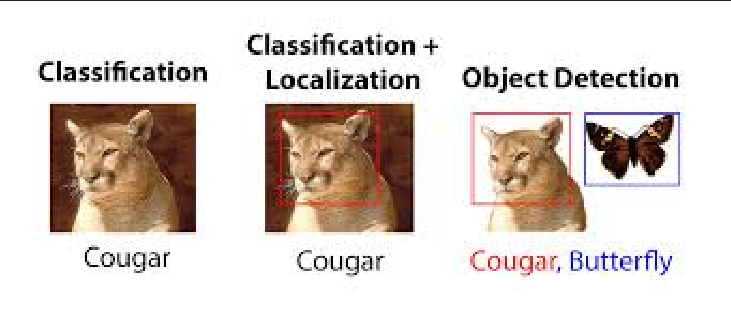


Fig 1.1) difference between classification, localization, and classification & Detection

Some of the non-neural methods are: -

1.2.) Viola-Jones object detection method based on Haar features.

1.2.2) Scale-invariant feature transform (SIFT)

1.2.3) Histogram of Oriented Gradients (HOG)

1.2.4) Other methods based on template, shape, or color matching

On the other hand, neural network techniques can do end-to-end object detection without explicitly defining features. They are far more accurate than non-neural based and are typically built on convolutional neural networks (CNN).

Some of the neural network methods are: -

1.2.5) Region-Based Convolutional Neural Networks (R-CNN, Fast R-CNN, etc.)

1.2.6) Single Shot Detector (SSD)

1.2.7) Retina-Net

1.2.8) You Only Look Once (YOLO)

* 1. **Project Motivation**

1.3.1) Traffic analysis: Vehicle counters can be used to collect data on the number of vehicles passing through a particular location over a given period. This data can be used to analyze traffic patterns and help with traffic management.

1.3.2) Transportation planning: Vehicle counters can be used to estimate the number of vehicles that use a particular road or highway, which can help with transportation planning and infrastructure development.

1.3.3) Environmental monitoring: Vehicle counters can be used to estimate the number of vehicles emitting pollutants in a particular area, which can help with environmental monitoring and policy development.

1.3.4) Revenue generation: Vehicle counters can be used to charge tolls on roads or highways based on the number of vehicles that pass through a particular location.

1.3.5) Marketing: Vehicle counters can be used to estimate the number of vehicles that pass by a particular location, which can be useful for businesses looking to advertise to a large audience.

The increasing number of vehicles on the road has led to various issues such as traffic congestion, air pollution, and road accidents. Accurate and real-time data about the number and types of vehicles on the road can help authorities and city planners address these issues effectively. Traditional manual vehicle counting methods are time-consuming and prone to errors. To address these challenges, automated vehicle counting systems have been developed, which use computer vision techniques to count vehicles accurately and efficiently in real-time.

In this project, we propose the use of YOLOv3, a state-of-the-art object detection model, to develop a vehicle counter application. YOLOv3 is a fast and accurate object detection model that can identify and locate objects in an image or video in real-time. By using YOLOv3, our vehicle counter application can accurately and efficiently count the number and types of vehicles in a video stream.

Our vehicle counter application will be able to count vehicles in a video stream captured by a camera mounted on a fixed location, such as a traffic pole or a bridge. The application will be able to count different types of vehicles, including cars, buses, trucks, and motorcycles. The application will be tested on a variety of video streams captured in different lighting and weather conditions.

We expect our vehicle counter application to be able to count the number and types of vehicles accurately and efficiently in a video stream in real-time. The application will provide valuable data about the number and types of vehicles on the road, which can inform a variety of transportation and mobility studies and analyses.

Our vehicle counter application using YOLOv3 will provide a fast and accurate solution for automated vehicle counting, which can improve the efficiency and effectiveness of transportation authorities and city planners. The application will also provide valuable data for researchers and analysts studying transportation and mobility patterns.

**Chapter 2**

**Literature Survey**

* 1. **Yolo (You Only Look Once)**

YOLO (You Only Look Once) is a popular algorithm for object detection that is fast and accurate. It was developed by Joseph Redmon and Ali Farhadi in 2015 and has been improved upon in subsequent versions, including YOLOv2 and YOLOv3.

YOLOv3 is the third version of the YOLO algorithm and is currently one of the most accurate and fastest object detection models available. It uses a convolutional neural network (CNN) to identify and locate objects in an image or video in real-time. The model takes an input image or video frame and divides it into a grid of cells, where each cell is responsible for predicting a set of bounding boxes and class probabilities for the objects in its area. The model makes a single pass over the image or video, processing it in its entirety and making predictions for each grid cell. This makes YOLOv3 very fast, as it does not need to perform multiple passes over the image or video like some other object detection algorithms.

YOLOv3 has achieved state-of-the-art performance on several benchmarks and is widely used in a variety of applications, including vehicle counting, object tracking, and pedestrian detection.

"State-of-the-art" (SOTA) refers to the best or most advanced level of performance that has been achieved in a particular field or area of study. In the context of YOLOv3, it means that the model has achieved the highest level of accuracy and speed among all object detection models that are currently available.

There are several ways to measure the performance of object detection models, such as mean average precision (mAP) and speed (frames per second). YOLOv3 has achieved high scores on both metrics, making it one of the most accurate and fastest object detection models currently available.

The term "state-of-the-art" is often used to describe the latest and most advanced technology or techniques in a particular field. It is generally used to refer to the highest level of performance that has been achieved at a particular point in time. As technology and research in a field continue to advance, the definition of "state-of-the-art" may change as new models or techniques are developed.

* 1. **Working of Yolo Algorithm**

A convolutional neural network (CNN) is a type of artificial neural network that is designed to process data from multiple sources, such as images, audio, and text. It is composed of multiple layers of interconnected nodes, which are inspired by the structure and function of the neurons in the human brain. CNNs are particularly useful for tasks such as image and video recognition because they can process and analyze data with a grid-like structure, such as the pixels in an image. They can learn and recognize patterns and features in the data by using a combination of convolutional and pooling layers, which extract and reduce the dimensionality of the data, and fully connected layers, which classify the data based on the extracted features.

Diagram

Description automatically generated

Fig 1.1) Working of Yolo Algorithm

YOLOv3 is a convolutional neural network (CNN) based object detection model that is fast and accurate. It works by dividing an input image or video frame into a grid of cells and making predictions for each cell.

Here is a high-level overview of how YOLOv3 works:

1. Preprocessing: The input image or video frame is resized and normalized.
2. Feature extraction: The model applies convolutional and pooling layers to the input to extract features.
3. Detection: The model applies several fully connected layers to the extracted features to predict bounding boxes and class probabilities for each grid cell.
4. Non-maximum suppression: The model removes overlapping bounding boxes and keeps the box with the highest probability for each object.
5. Postprocessing: The model applies confidence thresholding and class-specific filtering to the bounding boxes to improve the final detections.

**Diagram, engineering drawing

Description automatically generated**

Fig 2.2) Architecture of You Only look once

* 1. **The COCO dataset**

Common Objects in Context (COCO) Common Objects in Context (COCO) is a database that aims to enable future research for object detection, instance segmentation, image captioning, and person key points localization. COCO is a large-scale object detection, segmentation, and captioning dataset.

**Chapter 3**

**Methodology**

* 1. **Tools Requirements**

1. A computer with a GPU: YOLO requires a lot of computational power, so it is recommended to use a computer with a graphics processing unit (GPU) to train and run the model.
2. A dataset of images and videos: We need a dataset of images and videos of vehicles to train and test the model. Here we have used COCO dataset to serve the purpose. The dataset should be annotated with bounding boxes around the vehicles and labels indicating the type of vehicle.
3. Image and video processing tools: We need tools for preprocessing and postprocessing the images and videos, such as image and video editors and file format converters.
4. A development environment: we will need a development environment, such as a code editor or integrated development environment (IDE), to write and run the code for the model.
5. Additional libraries and packages: We need to install additional libraries and packages, such as OpenCV, NumPy, SciPy, imutils, argparse to support the development and deployment of the model.
   1. **Working Methodology**

## Step 1: Importing Libraries and Setting path

We need to import the video in which the objects and labels are to be recognized using the Video Capture function in cv2 open cv python. We have used argparse module in python to accept in the parameters from the user along with the confidence level and GPU if available.

## Step 2 : Load YOLOv3 Model:-

We’ll Need to load the YOLOv3 Model with weights and configuration files.

with open('E://YOLO-3-OpenCV//yolo-coco-data//coco.names') as f:

labels = [line.strip() for line in f]

network = cv2.dnn.readNetFromDarknet('E://YOLO-3-OpenCV//yolo-coco-data//yolov3.cfg',

'E://YOLO-3-OpenCV//yolo-coco-data//yolov3.weights')

layers\_names\_all = network.getLayerNames()

layers\_names\_output = \

[layers\_names\_all[i[0] - 1] for i in network.getUnconnectedOutLayers()]

probability\_minimum = 0.5

threshold = 0.3

# with function randint(low, high=None, size=None, dtype='l')

colours = np.random.randint(0, 255, size=(len(labels), 3), dtype='uint8')

## Step 3: Read Frames

We read the frame from the video file one by one. A blob is a 4D NumPy array object (images, channels, width, height).It has the following parameters:

## Step 4: Implementing Forward Pass

Pass each Blob through the network.

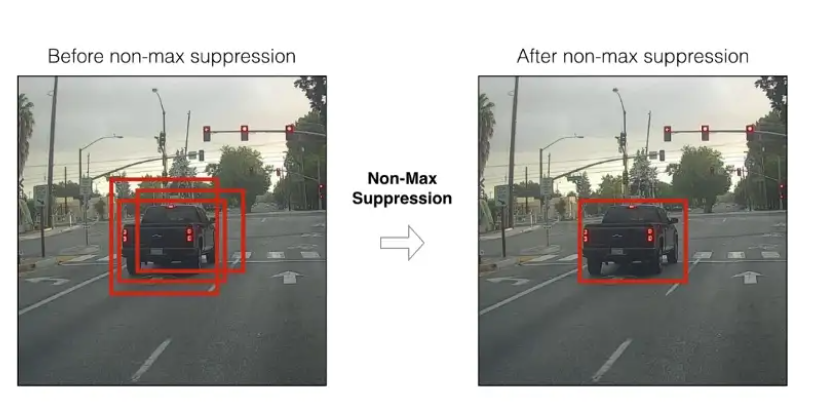


Fig 3.1) Non-Maximum Suppression

## Step 5: Non-Maximum Suppression.

The neighbourhood windows have similar scores to some extent and are considered as candidate regions. This leads to hundreds of proposals. As the proposal generation method should have high recall, we keep loose constraints in this stage. However, processing these many proposals all through the classification network is cumbersome. This leads to a technique which filters the proposals based on some criteria called Non-maximum Suppression.

## Step 6: Drawing of Bounding Boxes:

We Draw bounding boxes for each of the objects detected in the frame. We use the CV2.rectangle function to draw.



Fig 3.2) Vehicle Detection and Bounding Boxes

def drawDetectionBoxes(idxs, boxes, classIDs, confidences, frame):

# ensure at least one detection exists

if len(idxs) > 0:

for i in idxs.flatten():

# extract the bounding box coordinates

(x, y) = (boxes[i][0], boxes[i][1])

(w, h) = (boxes[i][2], boxes[i][3])

# draw a bounding box rectangle and label on the frame

color = [int(c) for c in COLORS[classIDs[i]]]

cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)

text = "{}: {:.4f}".format(LABELS[classIDs[i]],

confidences[i])

cv2.putText(frame, text, (x, y - 5),

cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, color, 2)

cv2.circle(frame, (x + (w//2), y + (h//2)),

2, (0, 0xFF, 0), thickness=2)

## Step 7: Writing processed Frames in File:

In the last step we write the proposed bounding boxes and label in the video frame and save it. Alongside the bounding of the objections we count the vehicles in it using the count vehicle function and we save the processed file in the directory.

**Chapter 4**

**Result and Discussion**

In general, a well-trained YOLO model should be able to count the number and types of vehicles accurately and efficiently in a video stream in real-time. The model should be able to handle a variety of lighting and weather conditions and should be able to count different types of vehicles, including cars, buses, trucks, and motorcycles.

The performance of a vehicle counting application using YOLO will depend on several factors, including the quality of the training dataset, the complexity of the model, and the conditions of the video stream.

Since we have used the coco dataset the accuracy of our model is very high and is able to determine and count nearly all the vehicles in the frame.

The COCO dataset contains over 200,000 images and over 250,000 annotated objects, including people, animals, vehicles, and everyday objects. The images are taken from a variety of real-world scenarios and include a wide range of object classes, scales, and poses. The annotations include bounding boxes and segmentation masks for the objects in the images, as well as captions describing the objects and their relationships. We used the 6 features out of the set of 80 features present in the coco dataset which are as follows:

["bicycle", "car", "motorbike", "bus", "truck", "train"]

* 1. **Challenges**

The challenges faced during the implementation of the vehicle counting application were mainly of GPU usage on our personal machine and whether our model will be able to work in dim light or severe weather conditions or not. The fact lies in it how will be our model be able to distinguish between the LMV and HMV in dim light and in harsh weather conditions.

**Chapter 5**

**Future Work**

There are several potential areas for future work in a vehicle counting application using YOLO. Some possible directions for further development are:

1. Improving accuracy: The model's accuracy could be further improved by using more sophisticated architectures or training on larger and more diverse datasets.
2. Incorporating additional features: The model could be enhanced by incorporating additional features, such as vehicle speed or direction, which could be useful for traffic management and analysis.
3. Enhancing real-time performance: The model's speed and efficiency could be improved by optimizing the model and the processing pipeline, which would allow the application to handle larger and more complex video streams. YOLO runs on heavy GPU usage which is not available to many organizations.
4. Integrating with other systems: The vehicle counting application could be integrated with other transportation and traffic management systems, such as traffic lights or toll booths, to provide a more comprehensive solution for traffic management.
5. Evaluating the application in different scenarios: The performance and effectiveness of the vehicle counting application could be evaluated in different scenarios, such as different types of roads or traffic conditions, to understand how it performs in different environments.

**References**

[1] Yousuf Ali, Md & Zhang, Xuan-De & Rashid, Md. Harun. (2020). Student Activity Detection Using Deep Learning with YOLOv3. Indonesian Journal of Electrical Engineering and Informatics (IJEEI). 8. 757-769. 10.11591/ijeei.v8i4.2585.

[2] https://medium.com/analytics-vidhya/object-detection-using-yolov3-d48100de2ebb

[3] You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon∗ , Santosh Divvala∗†, Ross Girshick¶ , Ali Farhadi∗† University of Washington∗ , Allen Institute for AI† , Facebook AI Research¶ <http://pjreddie.com/yolo/>

https://arxiv.org/pdf/1506.02640.pdf

[4]<https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006>

[5] A Simple Vehicle Counting System Using Deep Learning with YOLOv3 Model Preprint · May 2020 DOI: 10.13140/RG.2.2.15026.56001

[6] <https://www.baeldung.com/cs/yolo-algorithm#:~:text=3.-> ,You%20Only%20Look%20Once%20(YOLO),main%20reason%20for%20its%20popularity.

[7] Rachita Byahatti , Dr. S. V. Viraktamath , Madhuri Yavagal, 2021, Object Detection and Classification using YOLOv3, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 02 (February 2021),