MANGALORE UNIVERSITY

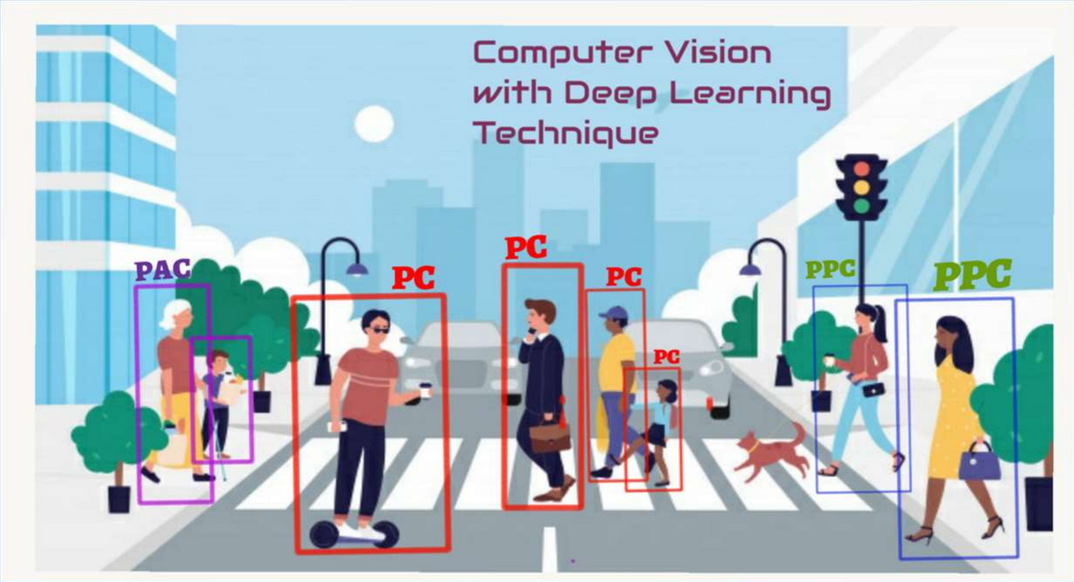
A project Report On

"REAL TIME PEDESTRIAN DETECTION USING DEEP LEARNING AND COMPUTER VISION"

Carried Out and Submitted
By
SANJAY KRISHNA HEGADE
Register No. : 193323738

Under the Guidance of
Dr. B H Shekar
Professor

Department of Computer Science
Mangalore University
Mangalagangothri-574199



REAL TIME **PEDESTRIAN DETECTION** USING DEEP LEARNING AND COMPUTER VISION

Introduction

- Pedestrian detection is a significant problem for computer vision, which involves several applications including Robotics, Surveillance, and the Automotive industry.
- It is one of the main interests of transport safety since it implies reducing the number of traffic collisions and the protection of pedestrians (i.e., children's and seniors), who are the most vulnerable road users.

What is Pedestrian Detection ?

Pedestrian Detection is a task of detecting the locations of the Pedestrians on image(frame) or Video(Sequence of frames).

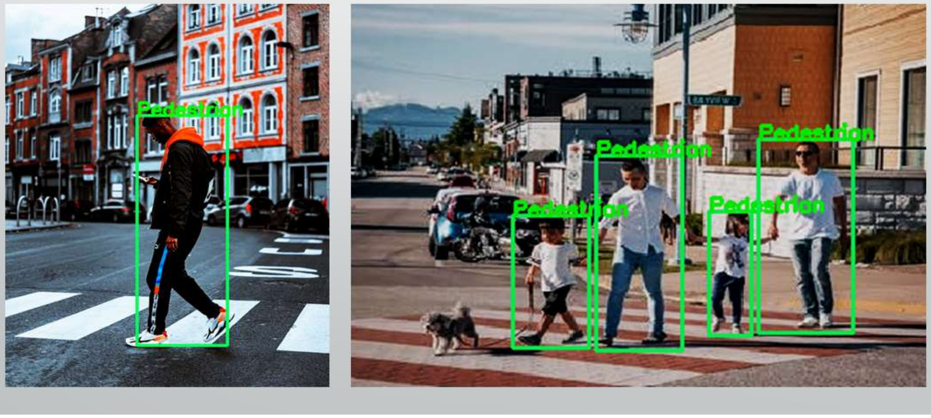


Fig.1.Some Examples of Pedestrian Detections

Why Pedestrian Detection ?

○ Pedestrian Detection System highly reduces the chances of Accidents.

○ Any Object Detection allows Automated vehicles to travel more smoothly since it understands the disturbances.

○ Improves ADAS and AI-Driving Assistants performance. Best addition to Driverless Vehicles and Driver Assistant Systems.

- Accurate pedestrian detection under occlusion can help drivers to locate pedestrians and timely remind drivers to give way to people. At the same time, the detection results are helpful to risk management of driving behavior and improve driving safety.
- This has been playing an important role in ensuring the traffic safety of modern urban areas.
- In the field of security, it has become an important task to find the target under the occlusion by monitoring.
- Therefore, the research and summary of pedestrian detection under occlusion has far-reaching significance for both individuals and society.

- There are almost 1.3 million persons die in road traffic collisions each year, and nearly 20-50 million are injured or disabled due to human errors inherited in the usual road traffic.
- Moreover, the clashes between cars and pedestrians are the leading cause of death among young people, and it could be effectively reduced if such human errors were eliminated by employing an Advanced Driver Assistance System (ADAS) for pedestrian detection.
- Over the last decade, the scientific community and the automobile industry have contributed to the development of different types of ADAS systems in order to improve traffic safety.
- More recently, The Nissan company has developed a system which recognizes the vehicle's environment, including pedestrians, other vehicles, and the road. Lexus RX 2017 has a self-driving system which is linked up to a pedestrian detection system.

Road Users	**No. of persons killed during 2016**	**No. of persons killed during 2015**
Pedestrians	15,746	13,894
Bicycles	2,585	3,125
Two-wheelers	52,500	46,070
Auto-Rickshaws	7,150	7,265
Cars, Vans, Taxis, LMVs	26,923	25,184
Trucks	16,876	16,611
Buses	9,969	10,743
Other Motor Vehicles (including E-rickshaw)	15,988	18,557
Others (Animals drawn vehicle, cycle, rickshaw, hand carts and other persons)	3,048	4,684
Total	**1,50,785**	**1,46,133**

- **These current ADAS systems still have difficulty in distinguishing between human beings and nearby objects.**
- **In recent research investigations, deep learning neural networks have frequently improved detection performance.**

Problem Statement

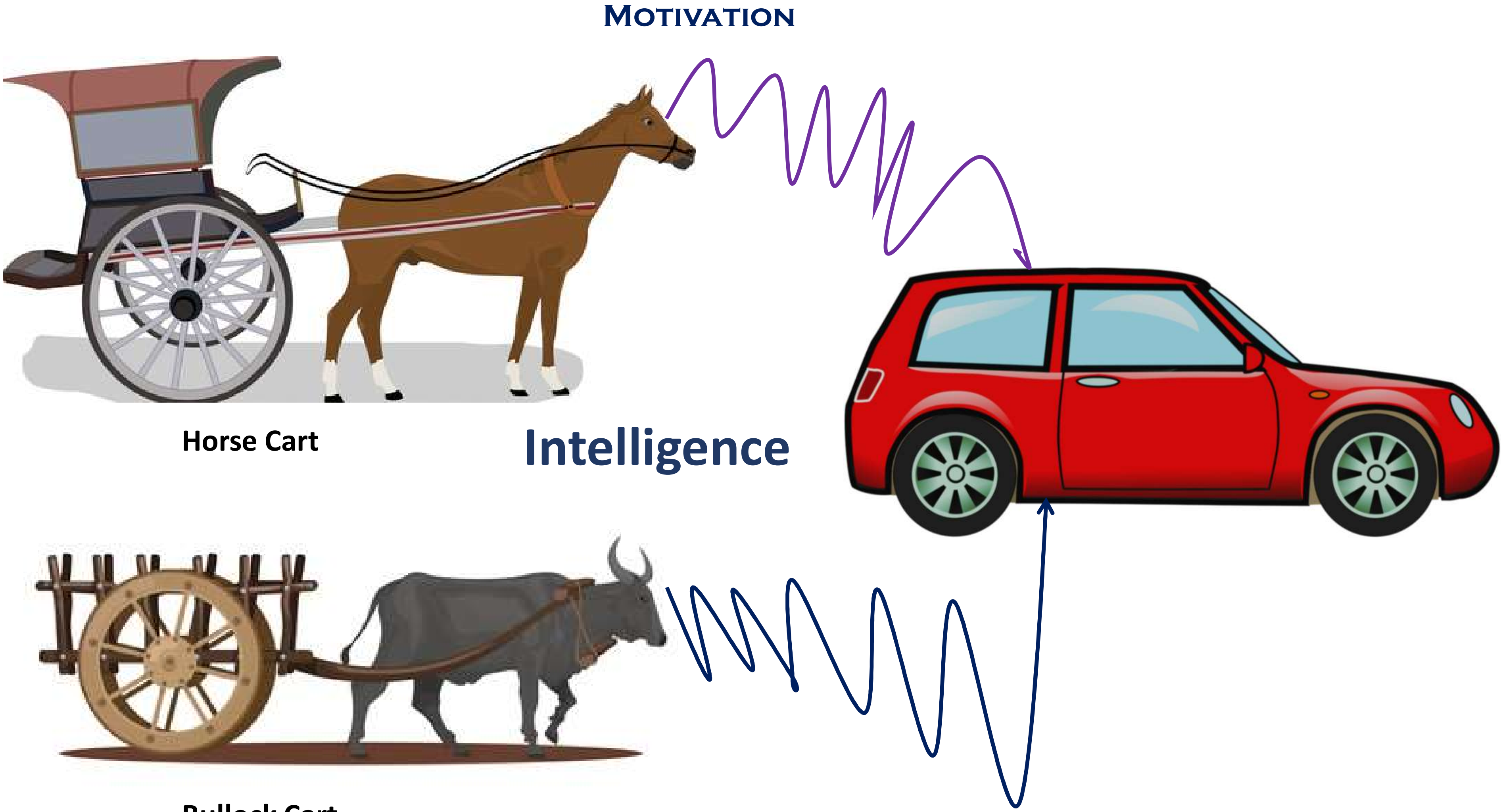
- Pedestrian detection is the problem of detecting the location of individuals who are walking in a particular indoor and outdoor environment.

- It is an essential and significant task in any intelligent Advanced Driving Assistant System (ADAS) or video surveillance system, as it provides the fundamental information for semantic understanding of the video footage.

- These applications need Real-time detection performance for timely decision making, by using limited computing power and resources available in the devices.

- Nowadays, Deep Learning based solutions are applied to the problem of pedestrian detection. One common challenge for any CNN based pedestrian detection is to meet the real time processing requirements where the Deep Learning model should run on embedded/Lower end devices with limited processing power and energy.

- In this project, a novel Convolutional Neural Network based object detection method with **Single Shot Detector (SSD)** algorithm and **MobileNet Backbone** proposed to detect Pedestrians.



Motivation

Horse Cart

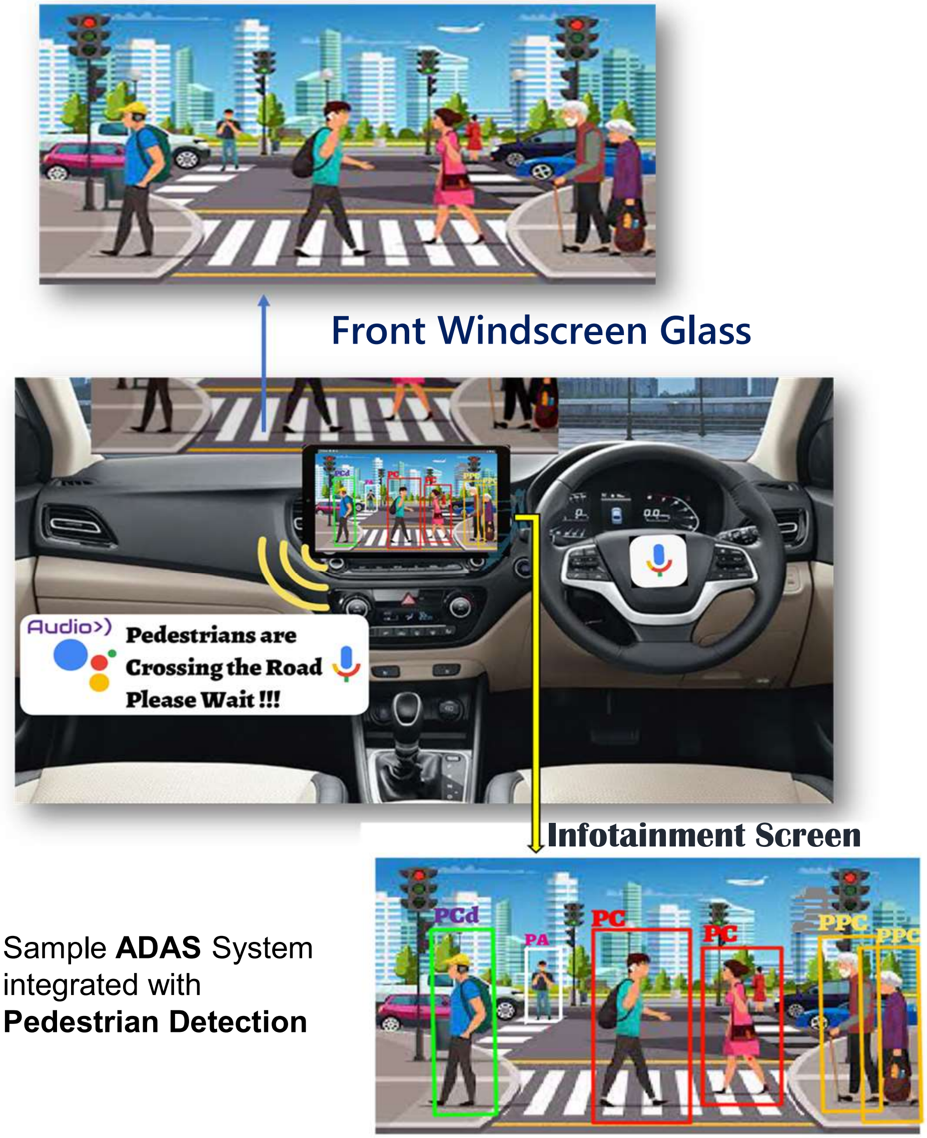
Intelligence

Bullock Cart

Applications

Autonomous Vehicles

Automatic Real Time Road objects Detection system (including Pedestrians) also applicable for huge transporting vehicles like Buses, Trucks where the other side corner of the Vehicle is not clearly visible to the driver. So Driving Assistant Systems can help them by detecting nearby objects and warning or managing accelerations automatically.



Front Windscreen Glass

Infotainment Screen

Sample **ADAS** System integrated with **Pedestrian Detection**

Video Surveillance / Security

Pedestrian detection and monitoring in a surveillance systems are critical for numerous utility areas which encompass unusual event detection, human gait, congestion or crowded vicinity evaluation, gender classification, fall detection in elderly humans, etc.

Robotics

A real-time Deep Learning based method for Pedestrian Detection (PD) is applied to the Human-Aware robot navigation problem. It is most important in Human - Robot interactions and its role in Social environments. Robots need to know if some obstacles are people or not. So better Real time PD is necessary for Human-Aware navigation System (HAN).

Traffic Analysis

Using Reliable Pedestrian Detection System, Traffic Analysis such as Pedestrian Counting, occurrences of Crowd analysis and various exploratory analysis on Pedestrian object can be performed.

Pedestrian Action recognition / Intention estimation

By Pedestrian action recognition, Vehicles can understand future movement of Pedestrians. For this action recognition Pedestrian Detection in challenging condition is fundamental.

Challenges

- Object detection is the first step that deals with detecting instances of semantic objects of a certain class, such as humans, buildings, cars, etc. in a sequence of videos.

- The different approaches of object detection are frame-to-frame difference, background subtraction and motion analysis using optical flow techniques.

- There are a number of reasons that are responsible for making pedestrian detection difficult which include existence of pedestrians in variety of postures and poses, wear different types of clothes , accessories , occlusion by other objects , variety in shapes, also the silhouette of pedestrians is diverse in nature, variation in illumination conditions, different appearances such as color, textures, carry different type of the objects such as bags, bicycles, variety of different activities such as standing, sitting, handshake position, different environmental conditions, low resolution images.

- One common challenge for any CNN based pedestrian detection is to meet the real time processing requirements where the Deep Learning model should run on embedded/Lower end devices with limited processing power and energy.

Objectives

▪ Many superior object detection algorithms have been proposed in literature; however, most of them are designed to improve the detection accuracy. As a result, the requirement of reducing computational complexity is usually ignored. To achieve real-time performance, these superior object detectors need to operate with a high-end GPU. To achieve real-time pedestrian detection without having any loss in detection accuracy, MobileNet+SSD network is proposed.

▪ This work introduces a complete system for Real time recognition of Pedestrians begins by training Pascal VOC 2007 and Pascal VOC 2012 Dataset using Convolutional Neural Network based Single Shot Detector (SSD) with MobileNet backbone.

▪ The trained model will be tested on local machine and evaluating the Predictions with Boundary Boxes for Real time video from Camera resources as well as existing video as input.

▪ Developing a System (GUI Application) which takes input as Video / Real Time Video and outputs detected Pedestrians with Boundary Boxes (BB) and it warns with Audio Instructions.

Literature review

Background

- The main purpose of object detection is to identify and locate one or more effective targets from still image or video data.
- It comprehensively includes a variety of important techniques, such as image processing, pattern recognition, artificial intelligence and machine learning.
- Many superior object detection algorithms have been proposed in literature; however, most of them are designed to improve the detection accuracy. As a result, the requirement of reducing computational complexity is usually ignored. To achieve real-time performance, these superior object detectors need to operate with a high-end GPU.

Related works :

- ***Yahia Said, Mohamed Atri & Rached Tourki* [3]**. This paper presents a method for human detection in video sequence. The Histogram of Oriented Gradients (HOG) descriptors show experimentally significantly out-performs existing feature sets for human detection. Because of HOG computation influence on performance, They chosen a more better HOG descriptor to extract human feature from visible spectrum images based on OpenCV and MS VC++ and Realized an image descriptor based on Integral Histograms of Oriented Gradients (HOG), associated with a Support Vector Machine (SVM) classifier and evaluate its efficiency.
- ***Rupesh A. Kharjul, Vinit K. Tungar, Yogesh P. Kulkarni & Samarth K. Upadhyay*[4]**. This project presents an application of a pedestrian detection system to reduce the number and severity of vehicle pedestrian accidents by active safety vehicles. In this system, they presented a pedestrian detection method based on images. They used Ada-Boost algorithm and cascading methods to segment pedestrian candidates from image to confirm whether each candidate is pedestrian or not a pedestrian. Recognizing classifier is skilled with support vector machine (SVM). They given input features used for SVM training are mined from both the sample gray images and edge images to the system.

- ***Hui Zhang, Yu Du, Shurong Ning, Yonghua Zhang, Shuo Yang & Chen Du [10].*** In this paper, the fast Region-based Convolutional Neural Network (Faster R-CNN) is used. Firstly, image features were extracted by CNN. After that, we built up a Region Proposal Network to extract regions that might contain pedestrians combined with K-means cluster analysis. And the region is identified and classified by detection network. Finally, the method was tested in the INRIA data set.
- ***Chengcheng Ning, Huajun Zhou, Yan Song & Jinhui Tang[12]. Single Shot Multi-Box Detector (SSD),*** which uses a single convolutional neural network to detect the object in an image. In this paper, we propose a method to improve SSD algorithm to increase its classification accuracy without affecting its speed. We adopt the Inception block to replace the extra layers in SSD, and call this method Inception SSD. The proposed network can catch more information without increasing the complexity.
- In practice, there are two types of mainstream object detection algorithms. Algorithms like R-CNN and Fast(er) R-CNN use a two-step approach - first to identify regions where objects are expected to be found and then detect objects only in those regions using convnet.
- On the other hand, algorithms like YOLO (You Only Look Once) and SSD (Single-Shot Detector) use a fully convolutional approach in which the network is able to find all objects within an image in one pass (hence 'single-shot' or 'look once') through the convnet. The region proposal algorithms usually have slightly better accuracy but slower to run, while single-shot algorithms are more efficient and has a good accuracy and that's what we are going to focus on in this section.

PROPOSED METHODOLOGY

- The Pedestrian Detection model begins by generating Training records from dataset (Pascal VOC 2007 and 2012).

- Here Dataset splitted into Training, Validation and Test respectively. We configured annotations of Pascal Dataset with Pascal Label Map which has a “Person” class.

- Training Record contains complete feature maps of dataset particularly for “Person” object. Then with a, pretrained model We done transfer learning with SSD+MobileNet Convolution Neural Network Method to train the desired Pedestrian Detection model shown in below architecture. Trained model tested with test images of dataset, Random Videos from Internet and Real time from camera.

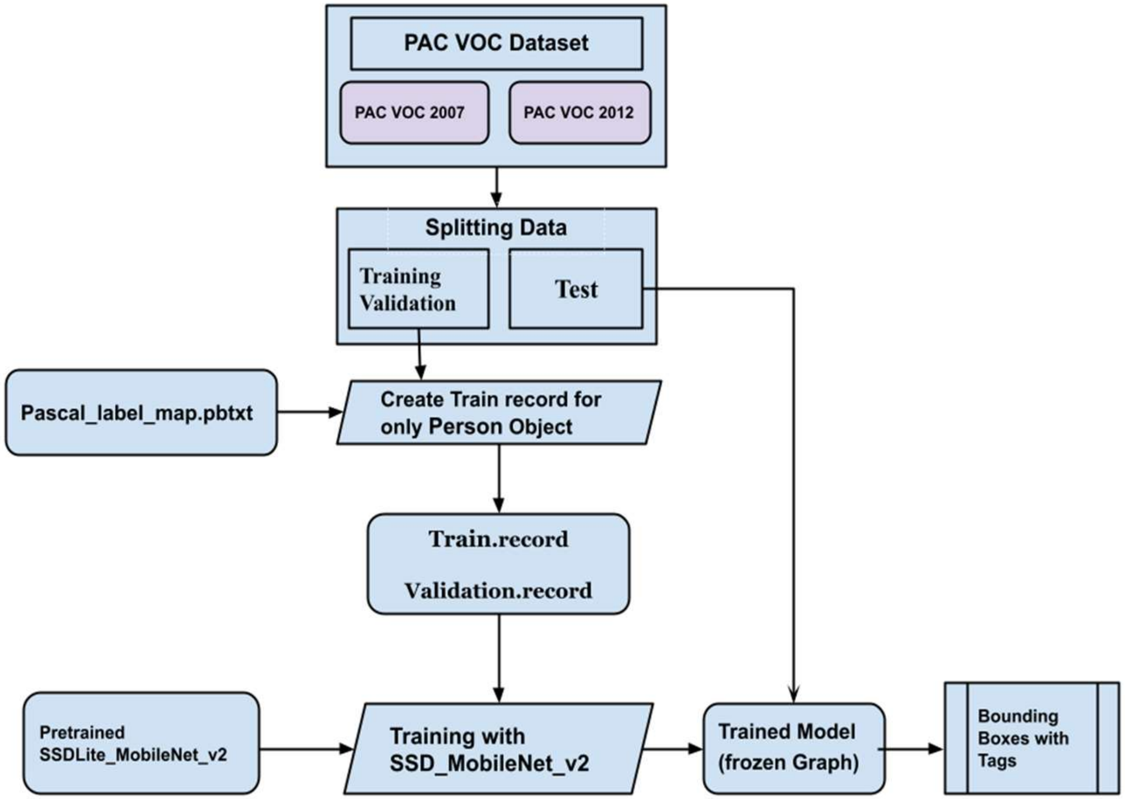


Fig. Architecture of the Pedestrian Detection System

Dataset

- The Pascal VOC challenge is a very popular dataset for building and evaluating algorithms for image classification, object detection, and segmentation.

- The Pascal Visual Object Classes (VOC) challenge consists of two components:

  (i) a publicly available dataset of images together with ground truth annotation and standardized evaluation software;

  (ii) an annual competition and workshop. There are five challenges: classification, detection, segmentation, action classification, and person layout.

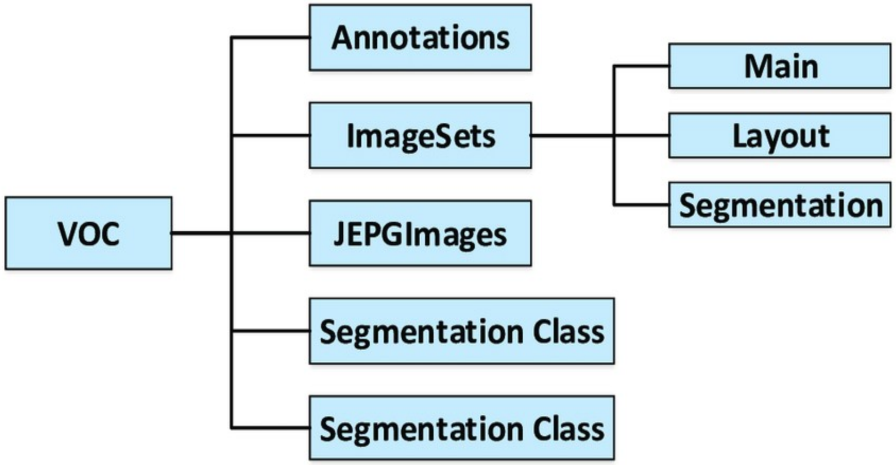
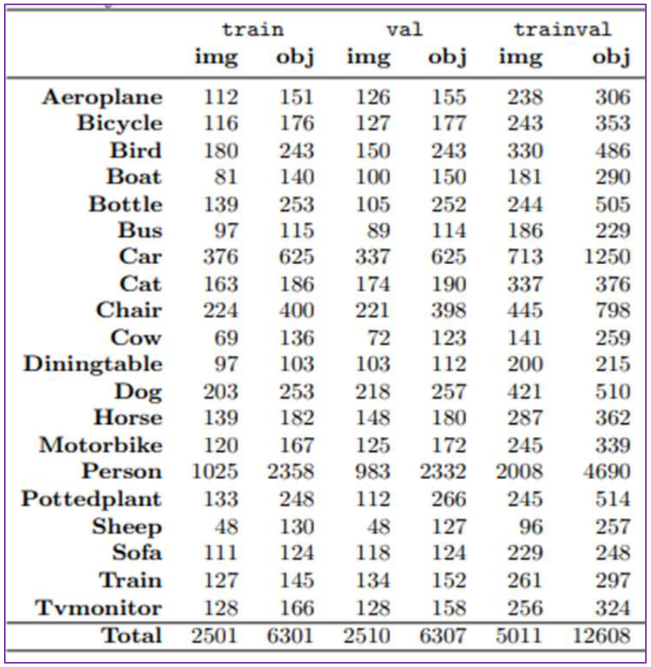


Fig. PAC VOC DataSet Structure

- **The PASCAL Visual Object Classes (VOC) 2012 dataset contains 20 object categories including vehicles, household, animals, and other: aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, TV/monitor, bird, cat, cow, dog, horse, sheep, and person.**

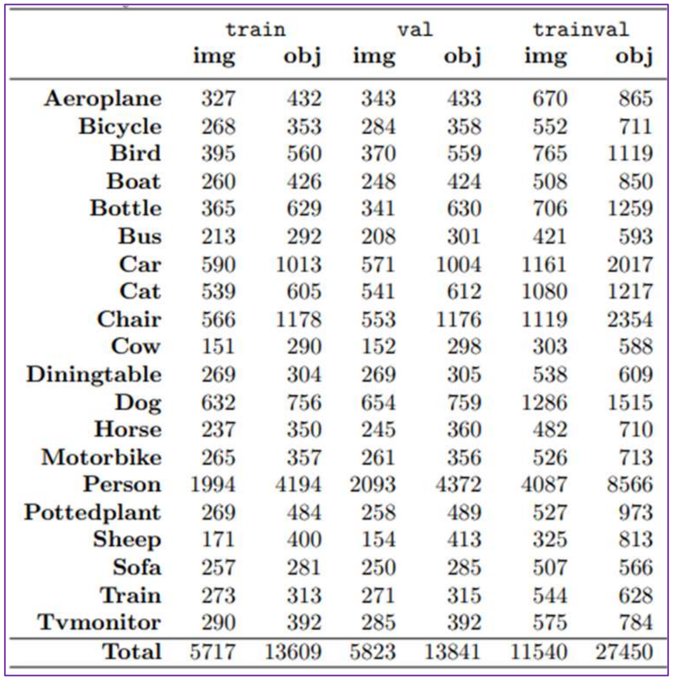
- **Each image in this dataset has pixel-level segmentation annotations, bounding box annotations, and object class annotations. This dataset has been widely used as a benchmark for object detection, semantic segmentation, and classification tasks.**

PAC VOC 2007 details



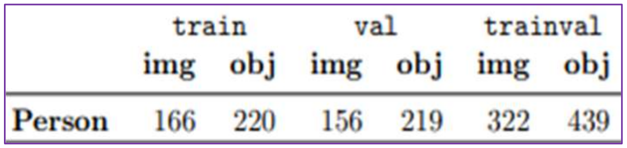
	train		val		trainval	
	img	obj	img	obj	img	obj
Aeroplane	112	151	126	155	238	306
Bicycle	116	176	127	177	243	353
Bird	180	243	150	243	330	486
Boat	81	140	100	150	181	290
Bottle	139	253	105	252	244	505
Bus	97	115	89	114	186	229
Car	376	625	337	625	713	1250
Cat	163	186	174	190	337	376
Chair	224	400	221	398	445	798
Cow	69	136	72	123	141	259
Diningtable	97	103	103	112	200	215
Dog	203	253	218	257	421	510
Horse	139	182	148	180	287	362
Motorbike	120	167	125	172	245	339
Person	1025	2358	983	2332	2008	4690
Pottedplant	133	248	112	266	245	514
Sheep	48	130	48	127	96	257
Sofa	111	124	118	124	229	248
Train	127	145	134	152	261	297
Tvmonitor	128	166	128	158	256	324
Total	2501	6301	2510	6307	5011	12608

PAC VOC 2012 details

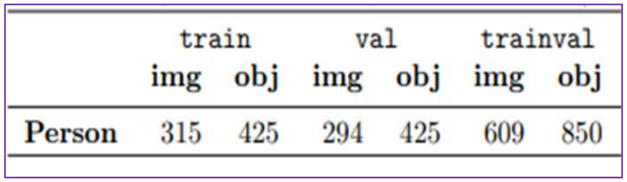


	train		val		trainval	
	img	obj	img	obj	img	obj
Aeroplane	327	432	343	433	670	865
Bicycle	268	353	284	358	552	711
Bird	395	560	370	559	765	1119
Boat	260	426	248	424	508	850
Bottle	365	629	341	630	706	1259
Bus	213	292	208	301	421	593
Car	590	1013	571	1004	1161	2017
Cat	539	605	541	612	1080	1217
Chair	566	1178	553	1176	1119	2354
Cow	151	290	152	298	303	588
Diningtable	269	304	269	305	538	609
Dog	632	756	654	759	1286	1515
Horse	237	350	245	360	482	710
Motorbike	265	357	261	356	526	713
Person	1994	4194	2093	4372	4087	8566
Pottedplant	269	484	258	489	527	973
Sheep	171	400	154	413	325	813
Sofa	257	281	250	285	507	566
Train	273	313	271	315	544	628
Tvmonitor	290	392	285	392	575	784
Total	5717	13609	5823	13841	11540	27450

Table. Statistics of the main image sets (Object statistics list only the ‘non-difficult’ objects)



	train		val		trainval	
	img	obj	img	obj	img	obj
Person	166	220	156	219	322	439

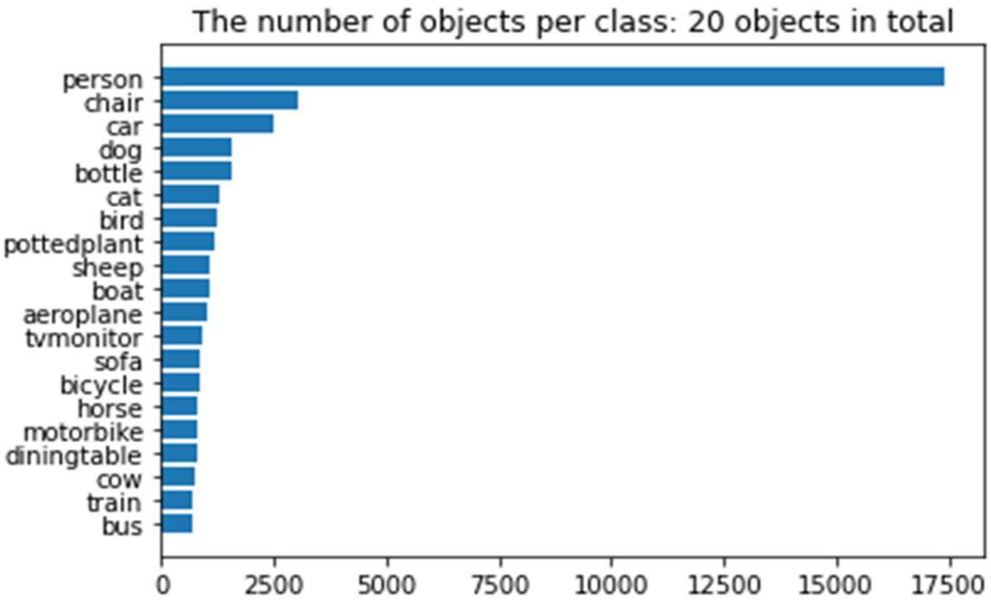


	train		val		trainval	
	img	obj	img	obj	img	obj
Person	315	425	294	425	609	850

Table. Statistics of the person layout taster image sets. Object statistics list only the ‘person’ objects for which layout information (parts) is present.

For this project, We merged PAC VOC 2007 and 2012 Dataset and saved it in VOCDevkit directory. It will be accessed when Training.

In **Fig.** Bar plot shows that "person" class is by far the largest and there are **17401** "person" objects in the entire data.



Nobj=4, height=375, width=500
Ferrari
person
person
person
car
Nobj=2, height=500, width=375
pottedplant
person
Nobj=1, height=240, width=320
person
Nobj=4, height=334, width=500
person
person
person
person

Fig. Visualizing randomly selected frames with annotations from **PAC VOC** Dataset

Training method

SSD network with MobileNet backbone used to train the model from **generated trained records** and pretrained weights i,e ssdlite_mobilenet_v2_coco.config. Here the advantages of our proposed method and its architectures are explained.

Single Shot MultiBox Detector (SSD)

The paper about SSD**: Single Shot MultiBox Detector** (by C. Szegedy et al.) was released at the end of November 2016 and reached new records in terms of performance and precision for object detection tasks.

To better understand SSD, let's start by explaining where the name of this architecture comes from:

- **Single Shot:** This means that the tasks of object localization and classification are done in a single forward pass of the network
- **MultiBox:** This is the name of a technique for bounding box regression
- **Detector:** The network is an object detector that also classifies those detected objects.

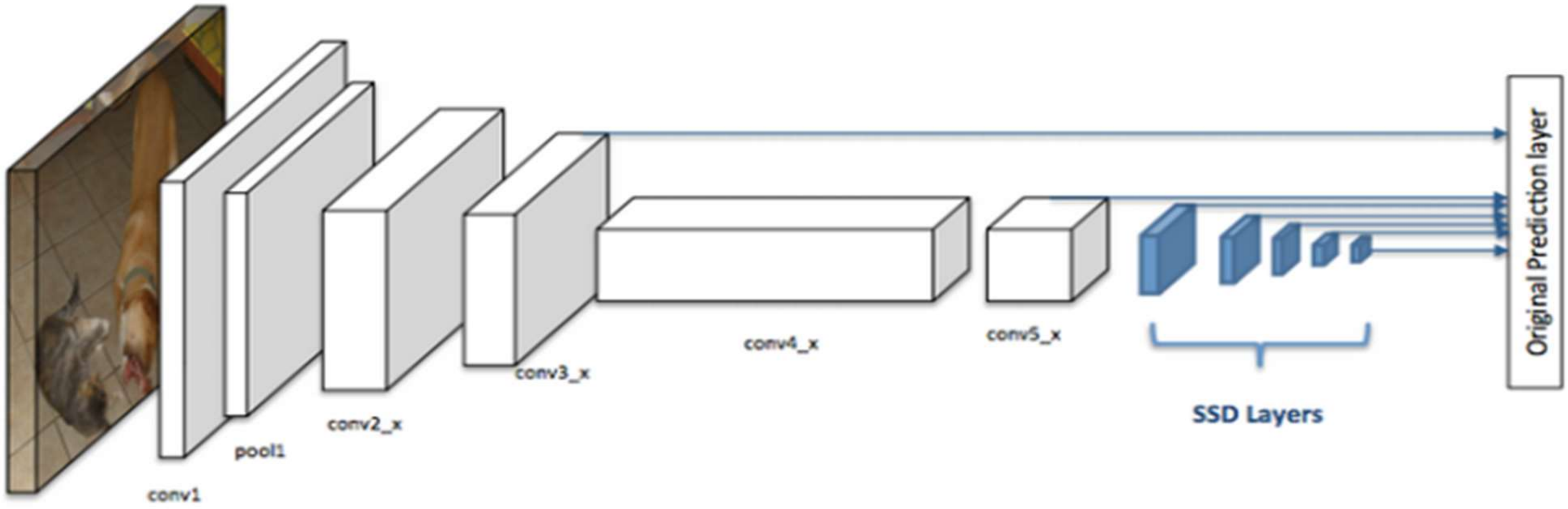


Fig - Architecture of a convolutional neural network with a SSD detector

Single Shot object detection or SSD takes one single shot to detect multiple objects within the image. The SSD approach is based on a feed-forward convolutional network that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes.

It's composed of two parts:

1. Extract feature maps.
2. Apply convolution filter to detect objects

SSD is designed to be independent of the base network, and so it can run on top of any base networks such as VGG, YOLO, MobileNet.

MobileNet Architecture

MobileNet is a lightweight deep neural network architecture designed for mobiles and embedded vision applications.

It is based on an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity. As a whole, the architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers.

- In MobileNetV2, there are two types of blocks. One is a residual block with stride of 1. Another one is block with stride of 2 for downsizing.
- There are 3 layers for both types of blocks.
- The **first layer** is **1×1 convolution with ReLU6.**
- The **second layer** is the **depthwise convolution**.
- The **third layer** is another **1×1 convolution but without any non-linearity.** ReLU6 is used due to its robustness when used with low-precision computation, based on the deep networks that only have the power of a linear classifier on the non-zero volume part of the output domain.

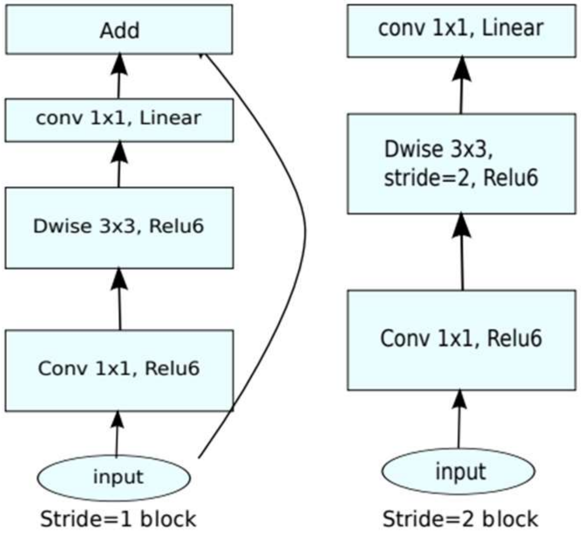
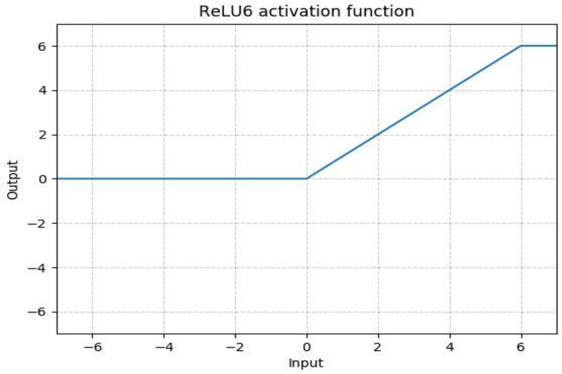


Fig - Layers of MobileNetV2 Architecture



- Fig. Shows Our Proposed Architecture that MobileNet architecture uses depthwise separable convolutions instead of standard convolution.

- This reduces the number of parameters significantly as compared to the network with normal convolution with the same amount of depth in the network, which results in lightweight deep neural networks.

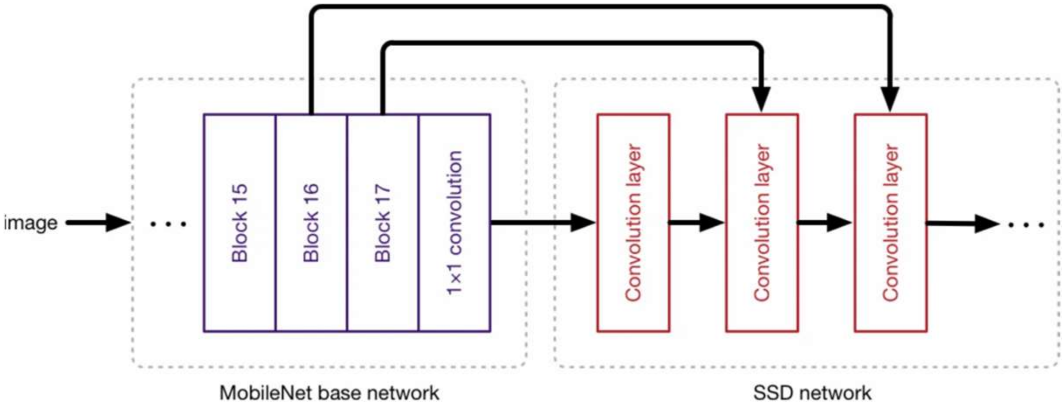


Fig. Proposed SSD MobileNet overview

The activation function “ReLU” is replaced by “ReLU6”, and the “Batch Normalization” layer was included in each layer of the newly appended structure to prevent the gradient’s disappearance. MobileNet is easy to train and takes relatively less time while training, which is highly desired for real-time implementation. This makes the network more reliable compared to VGG-16 and other available architectures.

Implementation

After successfully saving the model in the frozen_inference_graph.pb format, We deployed it in Desktop Application.

- Here existing video frames or Web Camera footage are passed to the Trained model, which returns detected bounding boxes of locations i.e, rows (x), columns (y), right and bottom.

- Then frames with Bounding Boxes and Tags are simultaneously displayed.

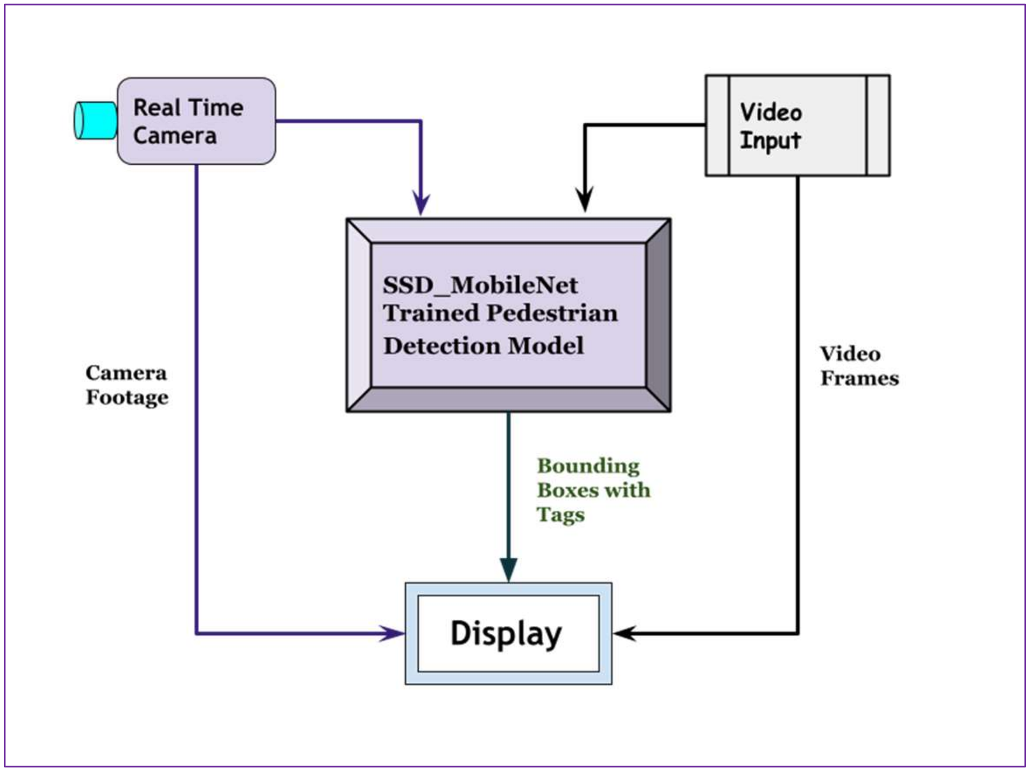


Fig. shows designed architecture for Real Time Pedestrian detection.

Technology and Tools Used

Google Colab : Google Colab is a free Jupyter notebook environment that runs entirely in the cloud. It is used for training Pedestrian detection model with the help of a single 16GB NVIDIA Tesla K80 free **GPU.**

OpenCV: OpenCV used all sorts of image and video analysis, like data preprocessing, testing and placing detected Bounding Boxes with tags into the frames. Also used in Real Time Pedestrian Detection Application.

TensorFlow : TensorFlow version 1.15 used for training the model with transfer learning and TensorFlow object detection API used for data processing, feature extraction.

JUPYTER NOTEBOOK : It is used for testing the system in the local environment and RealTime Pedestrian Detection.

Xml: xml.etree.ElementTree used for configuring Pascal Dataset's annotations.

MATPLOTLIB : Used for Data Explorations of Pascal dataset and displaying Testing images.

PIL : Pillow is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

PANDAS: Provides ready to use high-performance data structures and data analysis tools. Pandas module runs on top of NumPy and it is popularly used for data science and data analytics.

Kivy: Kivy is a graphical user interface open-source Python library that allows you to develop multi-platform applications on Windows, macOS, Android, iOS, Linux, and Raspberry-Pi. In addition to the regular mouse and keyboard inputs, it also supports multitouch events. The applications made using Kivy will similar across all the platforms but it also means that the applications fell or look will differ from any native application.

Modules

PAC VOC Dataset_Exploration.ipynb : Here We have done PAC VOC Data setup and exploratory analysis and plotted samples of dataset. Preprocessing is also done here.

PAC_VOC_Training.ipynb : This module represents main process of the system where Training records are generated with Pascal Label Map (pascal_label_map.pbtxt) and pipelined to ssdlite_mobilenet_v2_coco.config. Then it trained on SSD_MobileNet Architecture with a pretrained model (model.ckpt) and output Detection model saved in .pb (frozen_Graph).

Create_pascal_tf_record_only_person.py : This script extracts only one class, We used "person" here for example.

train.py : It configures ssdlite_mobilenet_v2_coco.config which has information of modified SSD_Mobilenet_v2 algorithm and Neural Network configurations.

Ped_BBox_Module.py : This module loads the saved (saved_model.pb) for predicting Boundary boxes on localized objects. It returns Boundary boxes locations as list.

Testing_SSD.ipynb : This module is used for testing images from PAC VOC Dataset and random images from internet sources. Also it performs Boundary box detection for existing videos as well as RealTime video from camera.

TensorBoard: TensorBoard provides the visualization and tooling needed for machine learning experimentation. Metrics are visualised by this tool.

Experimental Results

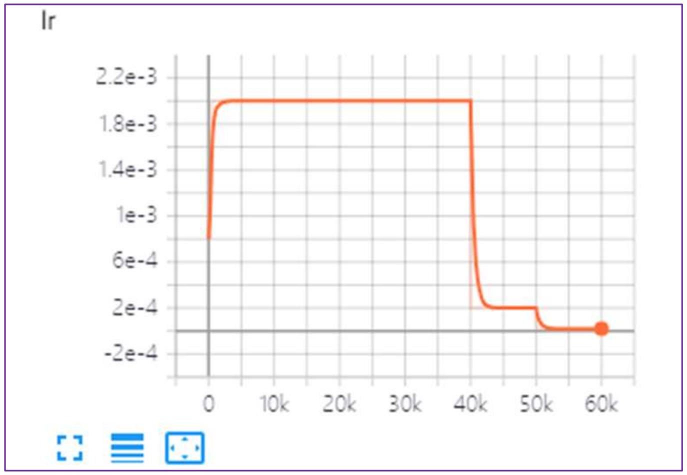


Fig. Learning Rate curve

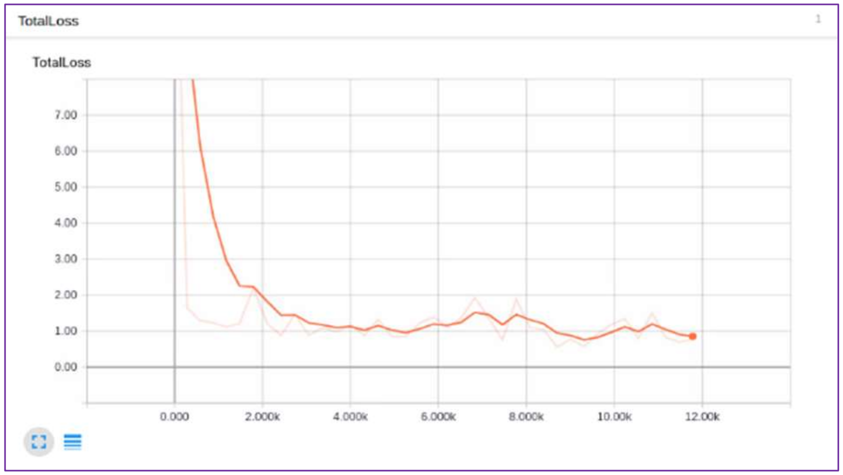


Fig. Total Loss per steps

Mean Average Precision (mAP) is the average of Average Precision (AP) . In some context, we compute the AP for each class and average them.

$$mAP = \frac{1}{n}\sum_{k=1}^{k=n} AP_k$$

$$AP_k = \text{ the AP of class } k$$

$$n = \text{ the number of classes}$$

The model was trained on PAC VOC Dataset and observed results on TensorBoard. We achieved 61.26% mAP at 30 FPS in RealTime Video.

Pedestrian detection results

Fig.- Results of Test images from PAC VOC Dataset

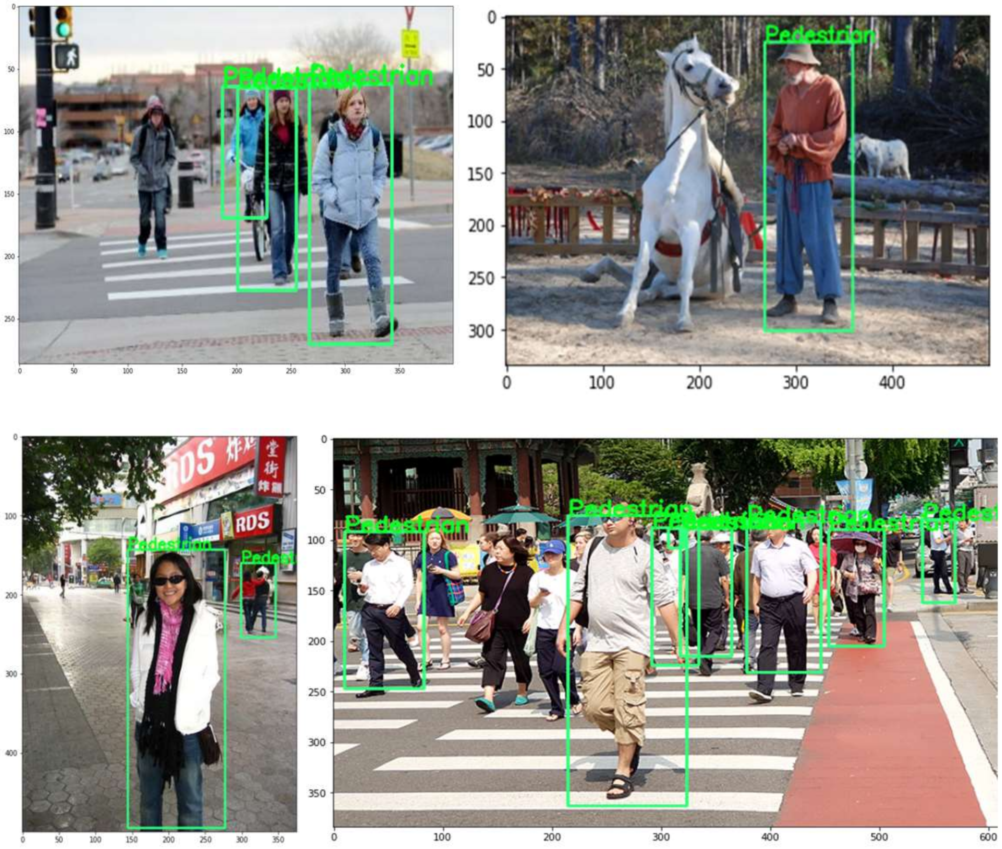
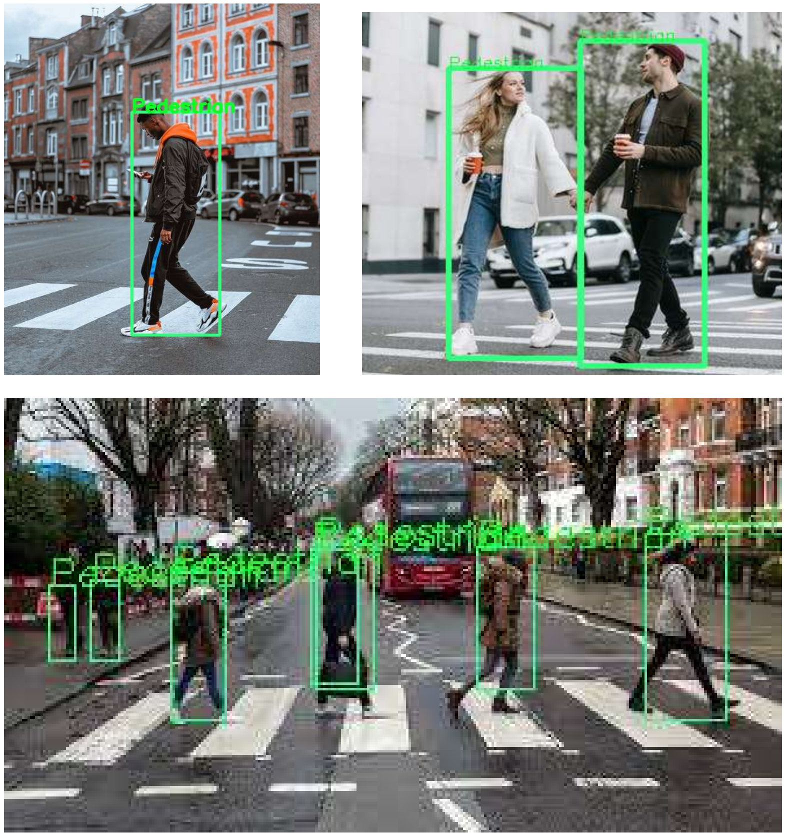


Fig. - Results of Test images from Internet



Graphical User Interface (GUI) using Kivy Python Library

GUI helps users to test detection performance easily. We developed a Simple Kivy based Cross Platform Application which runs on Windows, macOS, Android, iOS, Linux, and even on Raspberry-Pi.. Here users can input video from local machine using File Dialog or Real Time Detection from Web camera. Below Figures shows User Interface of Pedestrian Detection Apps usage.

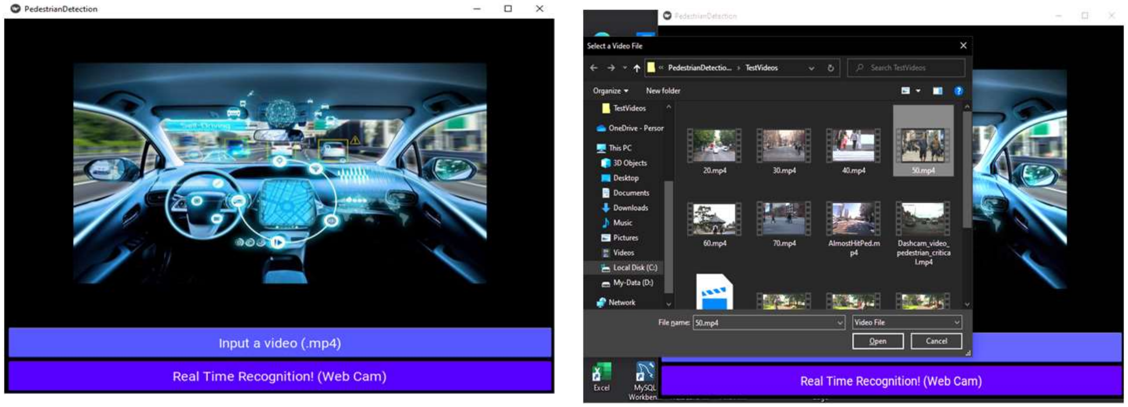


Fig. User Interface of Pedestrian Detection App

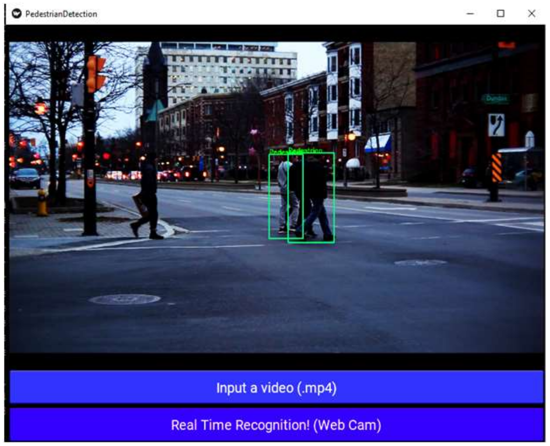


Fig - Output of selected video From Local Machine

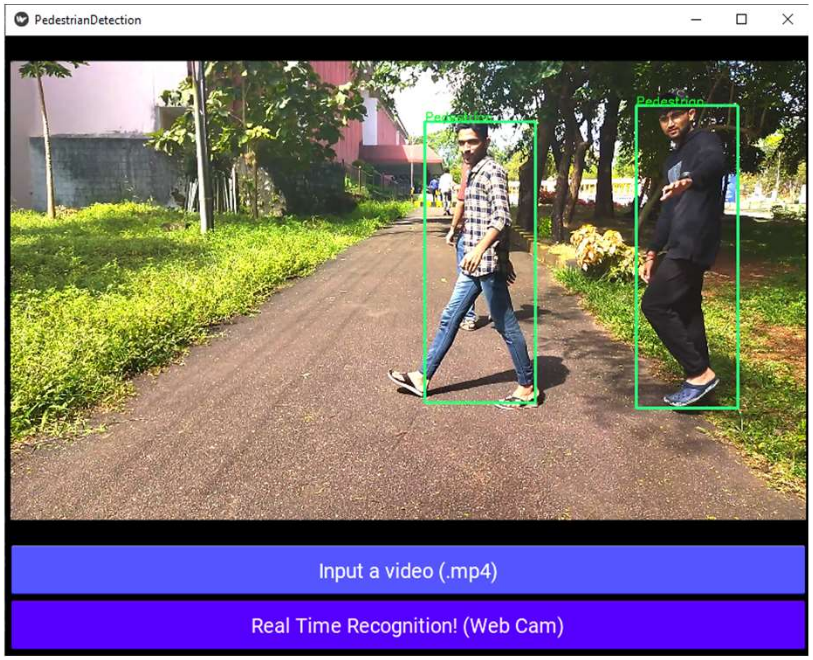


Fig - Output of Real Time video

Conclusion and Future Works

Conclusion

- It is a quite challenging task to reliably detect multi-scale pedestrians on a low-end edge device due to their limited resolution and information in images.
- This proposed new RealTime Pedestrian Detection System has better precision on the popular PAC VOC Dataset. And also describes how our new method is better than existing VGG 16-SSD method.
- In addition, the memory capacity of the entire network model is approximately 13MB. This is a great advantage for embedded platforms with limited resources.
- Experimental results show that the proposed Mobilenet-SSDv2 detector not only retains the advantage of fast processing of the original Mobilenet-SSD detector, but also greatly improves the detection accuracy. These advantages indicate that the SSD-MobileNetv2 detection model proposed in this project is more suitable for embedded or Lower end platforms.

Future Works

In future work, we will continue to optimize our detection network model, including reducing memory usage and increasing network computing speed. Many different adaptations, tests, and experiments have been left for the future due to lack of time (i.e. the experiments with real data are usually very time consuming, requiring even days to finish a single run). Future work concerns Optimized **Multiple Pedestrian object Action Recognition** on lower end devices.

THANK YOU

MANGALORE UNIVERSITY