LOCATION EXTRACTION AND PEOPLE COUNTING IN VIDEOS

Final Year Project Report submitted in partial fulfilment of the requirements for the degree of B.Tech. and M.Tech in Computer Engineering

by

M SWETHA

(Roll No: CED17I012)

Under the Supervision of

Dr. MASILAMANI V

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DESIGN AND MANUFACTURING, KANCHEEPURAM

May 2022

Certificate

I, **M SWETHA**, with Roll No: **CED17I012** hereby declare that the material presented in the Project Report titled **LOCATION EXTRACTION AND PEOPLE COUNTING IN VIDEOS** represents original work carried out by me in the **Department of Computer Science and Engineering** at the **Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram** during the year **2022**. With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student's Signature

In my capacity as supervisor of the above-mentioned work, I certify that the work presented in this Report is carried out under my supervision, and is worthy of consideration for the requirements of a Project report in the final year.

Advisor's Name: Advisor's Signature

Abstract

This report will address the techniques explored and implemented for capturing and extracting the location data in a video. The highlights of my work include implementing a mobile application for the project. The task includes understanding image Processing Techniques, Concepts of geotagging in a video and understanding text detection from an image.

The following details related to the project is covered by the report:

- Overview of current approach to geotagging and location capture of videos.
- Explanation of popular Object Detection algorithms and Comparison of existing algorithms that can be used for the current problem statement.
- Different implementation techniques for the same problem is analysed for the problem statement.
- Details of the experiments done to improve accuracy of the sample for location extraction is discussed.
- A detailed analysis of the approach used in implementing the solution for embedding location data and counting people along with their gps location.

Acknowledgements

I extend my gratitude to Indian Institute of Information Technology, Kancheepuram for providing me this opportunity to work with the college faculty. I wish to express my sincere thanks to Dr. Masilamani V for giving me this opportunity to perform my Project under his guidance and supervision. I would like to express my special thanks to the Institute and the faculty for their encouragement and making this experience a worthwhile.

Contents

List of Figures

Abbreviations

GPS	**G**lobal **P**ositioning **S**ystem
GCS	**G**lobal **C**oordinate **S**ystem
GIF	**G**raphics **I**nterchange **F**ormat
FFMPEG	**F**ast **F**orward **M**oving **P**icture **E**xpert **G**roup
OCR	**O**ptical **C**haracter **R**ecognition
CSV	**C**oma **S**eparated **V**alues
GPX	**G**PS **E**xchange **F**ormat
XML	**X**temsible **M**ark-up **L**anguage
UI	**U**ser **I**nterface
YOLO	**Y**ou **O**nly **L**ook **O**nce

Chapter 1

Introduction

1.1 Background

Location data is the information about specific geographic locations which are collected and tracked by the GPS satellite in a particular network. This location data includes latitude, longitude, elevation and other details which can help specify the exact location of the user at a given point in time. This data is to be stored and used for the video processing. Counting People in a video has a lot of practical applications and in this project we will look into counting people in a given geographical location represented by coordinates and analyzing the results and their use cases.

1.2 Problem Statement

To count the people along with the location of the frame in a given video with embedded location data. As a part of this project, we will be generating the input by creating videos with location data embedded followed by the algorithms and implementation for counting people and extracting location.

1.3 Objectives of the work

There are 2 parts to this project. Embedding location data and People Counter Application with the embedded location details. The steps involved in embedding data is as follows,

- Capturing the location information and storing the same.
- Implementation of the logic to embed location data into the videos.
- Algorithm for extracting text from an image.
- Storing the extracted data in a csv file and calculating accuracy.
- Image Processing techniques to improve the accuracy of extracting the location.

The steps involved in counting people is as follows,

- Choosing a video with embedded GPS location and extracting every frame.
- Extraction of text (location alone) using the improvised algorithm from the frame.
- Count people in every frame by choosing an existing algorithm according to the application.
- Storing the location data along with the number of people in each frame.

To understand the nuances of the sub tasks and consider the optimal implementations, various methods of implementation of the objectives are also discussed in the project along with the limitations.

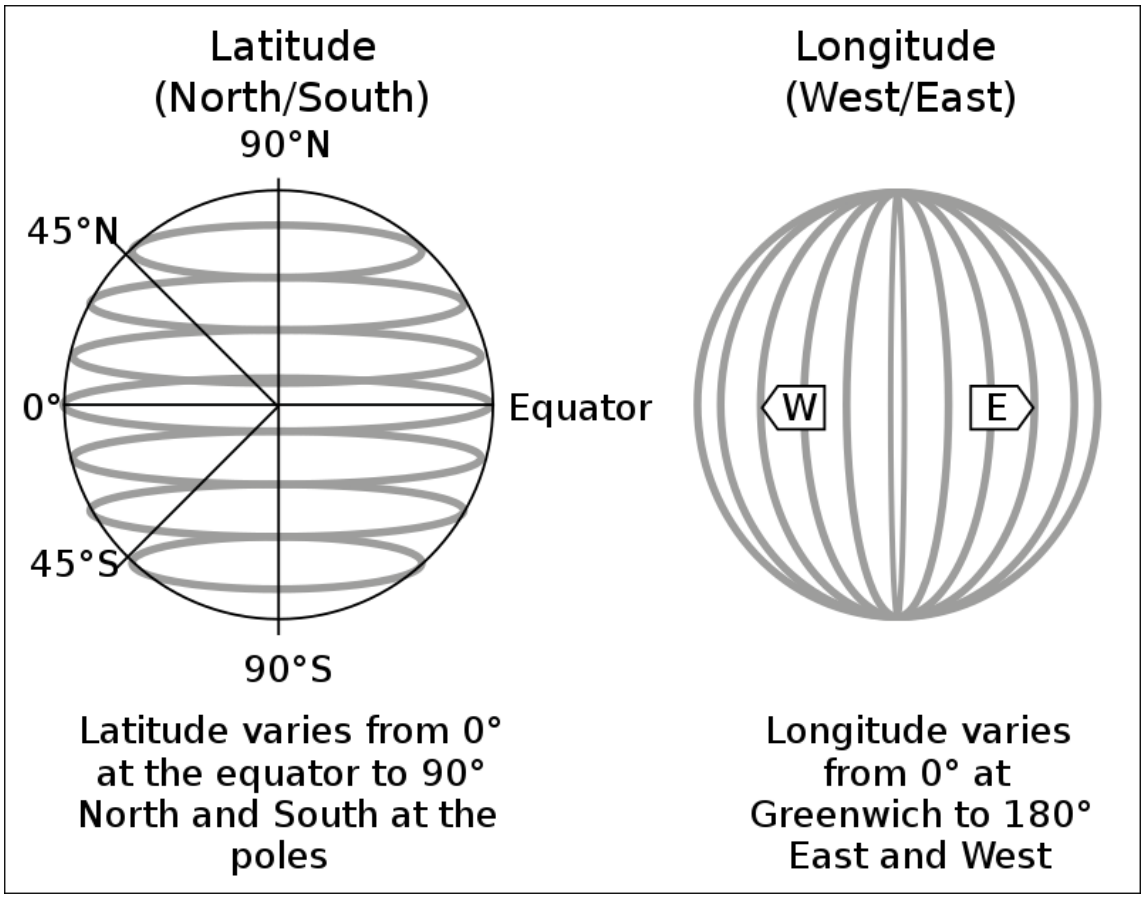


FIGURE 1.1: A picture to depict Geographical Coordinates

Chapter 2

Literature Review

2.1 Embedding and Extracting Location in Videos

2.1.1 Introduction to Geotagging

Geotagging is the process of adding Geographical Identification metadata to various media formats like images, GIFs and videos. The location data used in geotagging is derived from Global Position System where in Latitude and longitude are used to find each position on the earth from 180° west through 180° east along the Equator and 90° north through 90° south along the prime meridian.Geographical data or location data includes latitude,longitude coordinates, altitude, bearing, distance, accuracy of the obtained data, place names and time stamps.

The Geographic coordinate system is similar to the cartesian coordinate system except for the fact that it is not on a planar surface. GCS is a spherical coordinate system for communicating points on the Earth as Latitudes and Longitudes.

Latitude of a point on Earth's surface is the angle between the equatorial plane and the straight line that passes through that point and through (or close to) the center of the Earth. Longitude of a point on Earth's surface is the angle east or west of a reference meridian to another meridian that passes through that point.

Since Latitudes and longitudes are series of number representation, it is challenging to remember the numbers and the locations. So various other encoding systems were introduced like, :

- MaideHead Location System: It compresses latitude and longitude as short string of characters with limited level of precision to limit the number of characters needed for its transmission using voice. This is specifically aligned for radio usages.
- World Geographic Reference System: It is a grid based method of representing specific points on the Earth based on latitudes and longitude coordinates along with flexible notations and simpler representations. It is used for air-navigation and military purposes.
- Open Location Code: Apart from using coordinates like latitudes and longitudes, specific codes are designed to be used to represent street addresses which may be helpful for identifying buildings, landmarks and post codes.
- Geohash: Geohash is a public domain geocode system which encodes a geographic location into a short string of letters and digits. It is a hierarchical spatial data structure which subdivides space into buckets of grid shape, which is one of the many applications of what is known as a Z-order curve, and generally space-filling curves.

For simplicity and the current use case of embedding and extracting information only latitude and longitude based coordinate system is used.

2.1.2 Geotagging in Images

To understand embedding location data into video, we will first look into embedding location data into images. Incase of images or photos, the location can also be captured while capturing the photo or can be attached later in a specific format.The sample image 2.1 depicts geotagged photo. For capturing the location along with the photo, the device used to capture the frame should have GPS access so that that data can be used by the application. So the user must have a camera with an inbuilt GPS or a standalone GPS

with a digital camera. For Geotagging, we can add the image data to the frame by using image processing techniques. We can also store the data along with the file as an exif header. Exif is Exchangable Image Format file which specifies the metadata for the image file like properties, author, camera used for capturing ,etc.

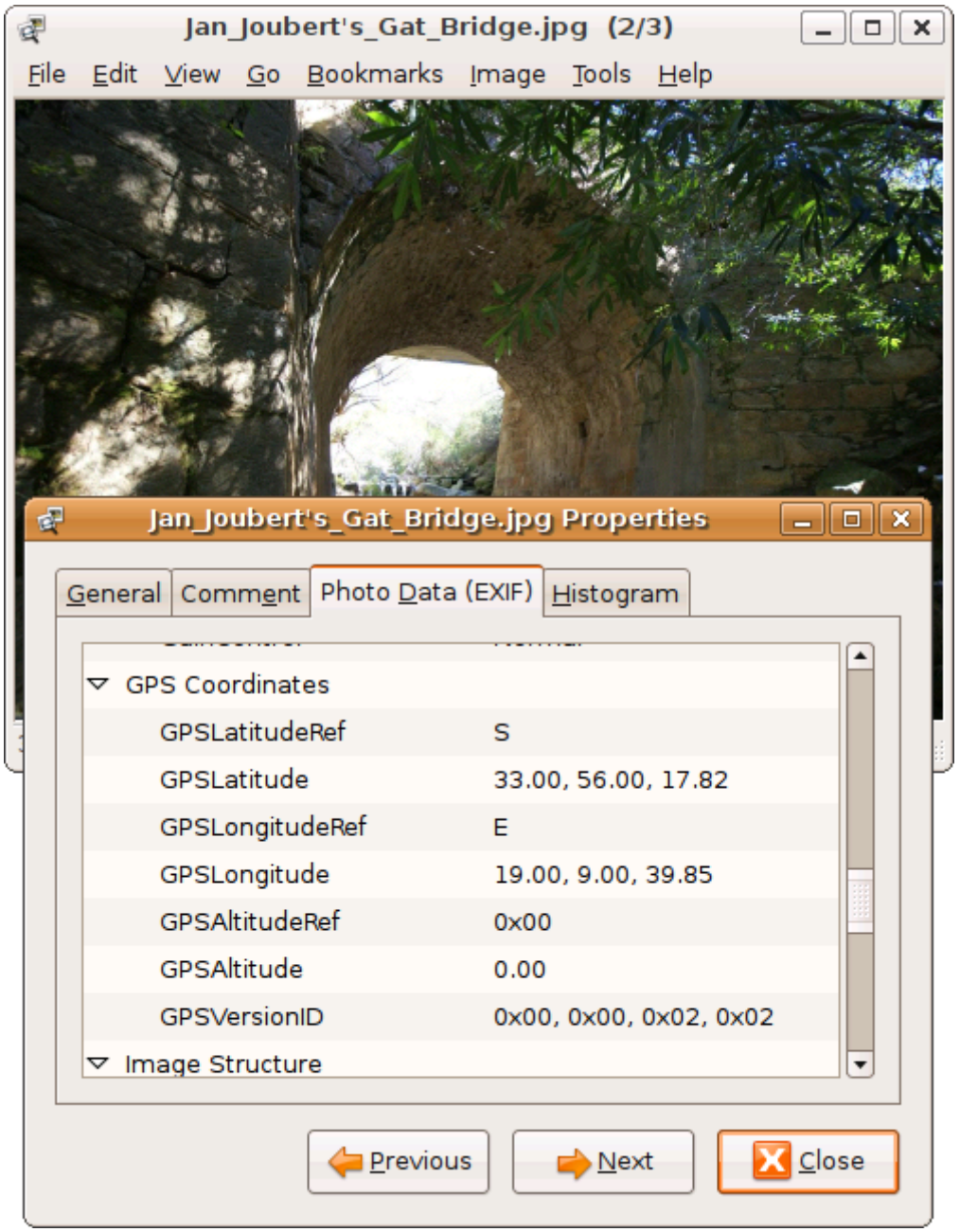


FIGURE 2.1: A picture to depict Geotagging in Images

2.1.3 Geotagging in Videos

Audio/video files can be geotagged via: metadata, audio encoding, overlay, or with companion files. Metadata records the geo-spatial data in the encoded video file to be decoded for later analysis.Find the image attached 2.2 which contains a geotagged video. Incase of videos, we cannot store it directly in the header. So a separate file or structure is used to store the data. Various methods of storing the location data is as follows:

- Database: A database can be used to store the GPS location of every timestamp or as a particular frequency according to the required method. In case of the database, the relationship and the correspondence of the frame, timestamp and the video is to be maintained carefully. Apart from this , this method uses additional external database usage.

- File format: There are various file formats that are specifically defined for Geopoints like GPX, GML and KML. GPX is an exchange data format specifically designed for GPS data information and has other features like Tracking points, defining routes and waypoints. GML is an XML based exchange data format which is used to store other information like geometry, coverage, unit of measurement and map styling rules. KML is an XML notation for expressing geographic annotation and visualization within two-dimensional maps and three-dimensional Earth browsers.

Current implementation of storing the data uses either of the above methods where the extra data is available separately. The motivation of this project is to store the latitude and longitude along with the video as a single file rather than the video separately and the corresponding file or database separately.

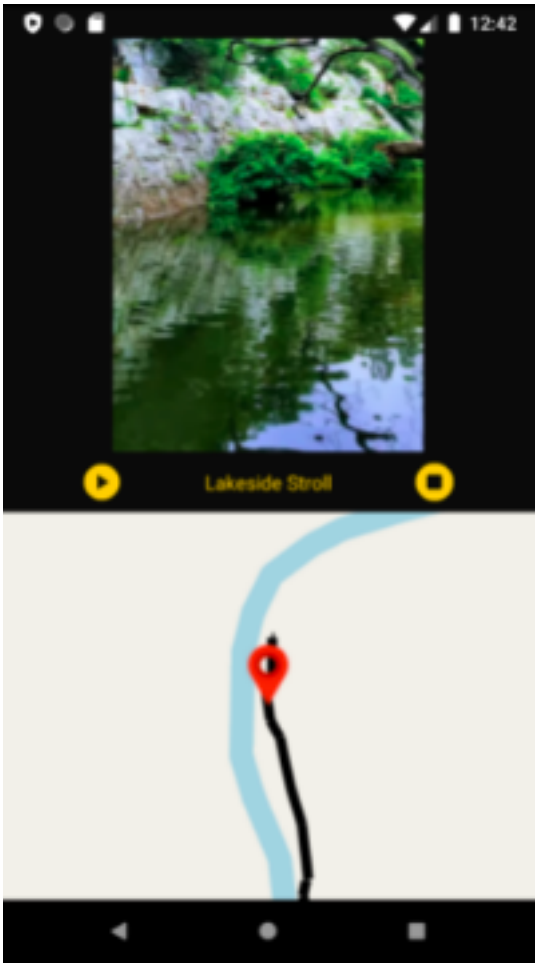


FIGURE 2.2: A picture to depict Geotagging in Videos

2.2 People Counter Application

2.2.1 Introduction to Object Detection

Object detection is a Computer Vision and image Processing technology which is used to identify various classes of objects given a media file.Object detection is a combination of object localization and Image classification since we have to find the location of multiple objects in the file along with their classes. Object detection algorithms are also extended for object tracking and for finding the total number of people in a given video. In any object detection model, features of every class in the dataset is found out in the form of a feature map such as edges, corners, ridges, etc and that feature map is used for testing and predicting the class of the object. General object detection models include neural networks based(deep learning focused) and non - neural network based(machine learning focused). We will see an overview of these algorithms and compare those in the next section.

2.2.2 Comparison of algorithms

2.2.2.1 Non Neural Network Algorithms

In non-neural network based algorithms, it is necessary to define the features in order to perform the classification. A simple example of non-neural network based algorithms is Histogram of Gradients. Histogram of Gradients is similar to a feature descriptor which calculates the occurrence of gradient orientations. According to the name, histogram of localised gradients is found out and the occurrence is used for object recognition. This algorithm gives a feature map and machine learning algorithm like Support Vector Machines is used on top of this for object detection. This method has been modified and used specifically for pedestrian detection. However in HOG, there is an issue with false positives and is unsuitable for real time processing due to slow calculation speed. According to the figure 2.3, all traditional non neural network based algorithms are generally unsuitable for real time processing due to low speed.

Method	Datasets	log-average miss rate	FPS
HOG+liner SVM [1]	INRIA	68.46%	.239
PoseInv [19] (HOG+AdaBoost)	INRIA	86.32%	.474
VJ [6] (Haar-like+AdaBoost)	INRIA	94.73%	.447
Shapelet+AdaBoost [21]	INRIA	91.37%	.051
ICF+AdaBoost [10]	INRIA	14% at 1 fppi	/
LatSvm-V1 [22] (DPM+Latent SVM)	PASCAL	79.78%	.392
LatSvm-V2 [23] (DPM+Latent SVM)	INRIA	63.26%	.629

FIGURE 2.3: Comparison of Traditional Object Detection Algorithms

2.2.2.2 Convolution Neural Networks

Regional - Convolution Neural Network is the first large scale solution. This method consists of region proposal and feature extractor after which we use a suitable classifier algorithm for classification of the object. Region Proposal step identifies and calculates bounding boxes for various classes present in the image. Feature Extraction consists of convolution neural networks for extracting features from each region of interest obtained after region Proposal.(Example of Feature Extractor is AlexNet Deep CNN). Other algorithms such as Fast R-CNN, Faster R-CNN that addressed the slow calculation speed (frames processed per second) of R-CNN. However, the training process is not very straightforward and could not meet the criteria(speed and accuracy) required for a real time detection model.

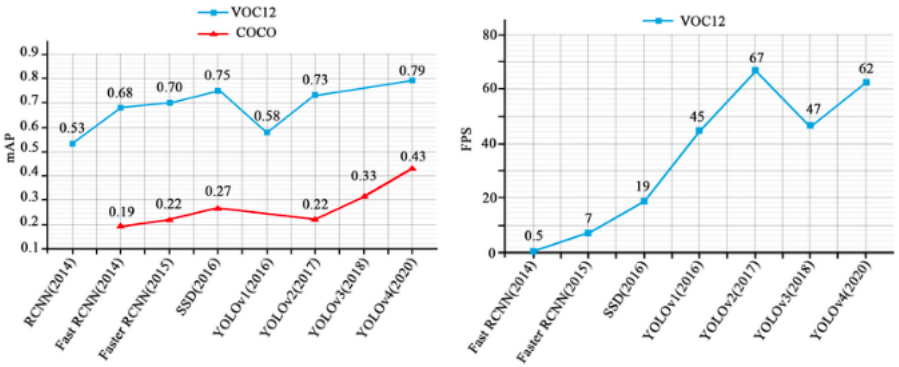
2.2.2.3 Single Shot Detector

Single Shot Detector uses a single Deep Neural Network unlike R-CNN where regions are proposed followed by multiple neural networks. In this approach, discretization of bounding boxes is done over different aspect ratios and this is used per feature map location. Combination of multiple feature maps is used for prediction. Single Shot Detector is not suitable for small object predictions and in the presence of too many objects in the image.

2.2.2.4 You Only Look Once (YOLO)

YOLO is one of the most popular algorithm for object detection in recent years and it is extremely fast and processes 45 to 155 frames per second. In this approach, bounding boxes are found out and used for predicting class labels. For predicting bounding boxes, intersection of Union and non maximal suppression is used. These two techniques help to identify the bounding boxes better and quicker where NMS helps in identifying the optimal box by considering highest probability in case of many bounding boxes of the same object and IOU helps in identifying the overlap between the actual bounding box for the object and the predicted one. YOLO has higher speed and better average precision among other algorithms , however, one drawback is difficulty in identifying small objects which have been minimised in newer versions of YOLO.

The below figure 2.4 that analyses mean Average Precision of various algorithms for Pedestrian detection. Using all this information and data [1], YOLO is the model that is most preferred for people counting in case of our problem statement.



Mean Average Precision (mAP) and Speed (FPS) overview of eight most popular object detection models on Microsoft Common Objects in Context (MS-COCO) and PASCAL Visual Object Classes (VOC) datasets.

FIGURE 2.4: A picture to depict comparison of various algorithms for object detection

Chapter 3

Methodology

3.1 Embedding and Extracting Location in Videos

3.1.1 Obtaining the permissions

This project involves development of a mobile application. Various tasks involve obtaining the permission from the device before capturing the video. The permissions required are as follows:

- Writing to external storage for storing the GPX information, CSV file information.
- Reading from external storage to write the data from the GPX file.
- Microphone and Camera permissions to capture and record the video.
- Location permission of the device.

3.1.2 Capturing video with GPS Location

The video is to be captured using this application only so that the corresponding gpx file can be produced and used. On recording the data, the video is stored in the device along with the GPX file. Event object listeners are used to implement capturing of location. The listeners detect change in GPS location and on a change of value, the timestamp and the latitude and longitude values are recorded in the GPX file. Incase of locations where GPX could not be captured, the previously captured data is used as the listener will not detect any change in location.

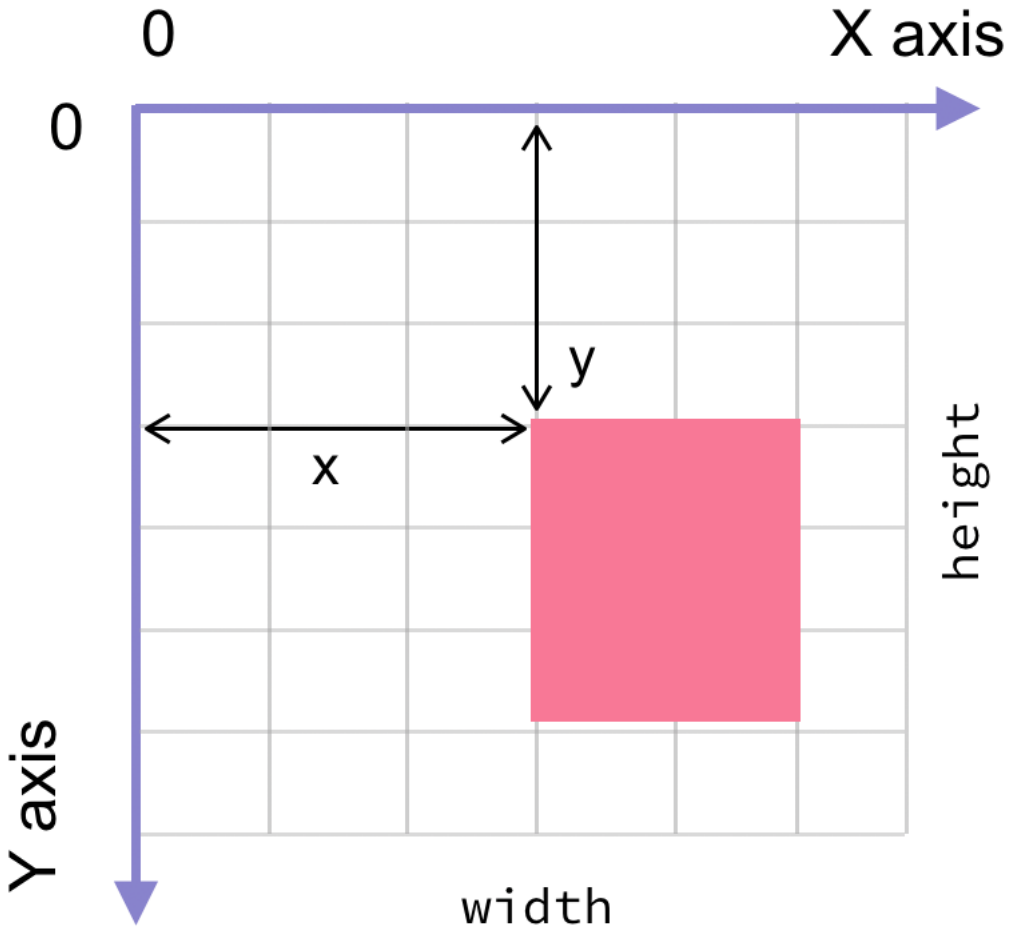


FIGURE 3.1: A picture to depict Canvas Coordinates

3.1.3 Embedding the Video with the Location

The video created is extracted frame by frame using FFMPEG Library in Android Java. Now image processing techniques is used to add the location data.

3.1.3.1 Finding the Location Data

For every frame, we have to find the corresponding location data. Since we have stored the timestamp, we use the frame number , frames per second and the starting time of the video to find this. The delta time is evaluated using the current timestamp and start time. The frame number represented by the current GPX Iterator is the product of delta time and frames per second. With this the frame number can be compared. GPX Iterator is used to read every item from the gpx file and the get Latitude and Longitude functions can be used to obtain the data.

3.1.3.2 Writing the Data on the Image

A frame is represented by a set of pixels in a coordinate system. To write data on to an image, canvas function in Android can be used. This functions draws on to the image by modifying the pixel values in the coordinate system defined by the canvas. So first the frame is drawn onto the canvas and the coordinate system is used to specify the pixel location and pixel values. The background is first made black by specifying the rectangle corners and the latitude, longitude value is written on it. This will embed data onto a frame. This is done for the entire video by processing every frame. The individual frames with a frame rate of 25 frames per second are converted back to a video using FFMPEG library.

3.1.4 Extracting the location data

The video with location data is available and now we have to extract the location data from the video and store it in a file with the corresponding frame number.

3.1.4.1 OCR

For extracting Location data, tesseract Library is used. The tesseract OCR engine uses language-specific training data in the recognize words. The OCR algorithms bias towards words and sentences that frequently appear together in a given language. It analyses the image by finding lines and the words initially. The key parts in the initial finding is blob filtering and line construction. Once the text line has been found out, the lines are fitted using a quadratic spline using a least square fit approach. Te text lines are tested to find if they are in the right pitch To find this, the words are chopped into characters. The results are improved by chopping the blob with worst confidence from the character classifier. After this an associator is used to associate broken parts. The associator makes an A* (best first) search of the segmentation graph of possible combinations of the maximally chopped blobs into candidate characters. Now a shortlist of candidate class is determined using the pruner class. Linguistic analysis is done on the class to identify the language dictionary and use the grammar rules to validate the text generated.

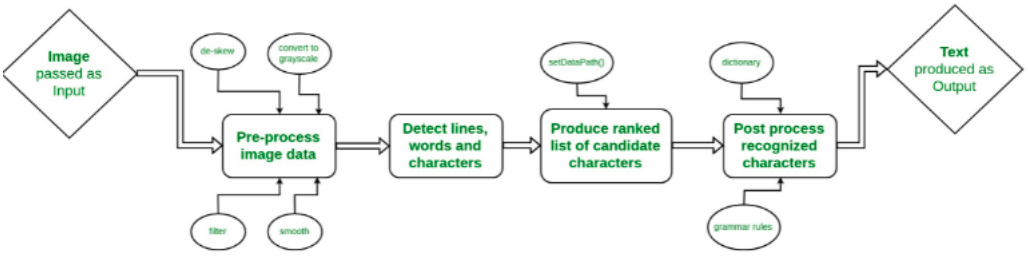


FIGURE 3.2: A picture to depict working of OCR

3.1.4.2 Writing to CSV

After getting the location data, the data is written into a CSV file using File Processing techniques in Android Java. The CSV file created contains the frame number, Latitude and Longitude for the video. This file can further be used to compare and find out the accuracy of the OCR classifier used and improve the same.

3.2 People Counter Application

3.2.1 Selecting a Video

In this step, we have to use the video that was generated with the embedded location data on it. So the video that was generated has to be stored in the project external directory and all the videos with the embedded data should be made available for choosing the video to count people frame by frame. All files with embedded data is stored separately and that folder is accessed and the files are fetched. onclick listener functions in android java is used to map the file being fetched and process it.

3.2.2 Extracting Location Data

This process is similar to the algorithm used before for extracting location data. We apply OCR on the video using an improvised version for our use case by pre-processing and post-processing the images frame by frame. In this, we do not have the original latitude and longitude data and hence we consider the extracted data as the actual GPS location in every frame.

3.2.3 Detecting and counting People

For detecting and counting people in every frame, YOLOv4 is the algorithm used. The model of Yolov4 is directly implemented after training and testing the model on an existing data set.

3.2.3.1 YOLO Architecture

From the details present in figure 2.4, YOLO v4 architecture is considered for the implementation. According to Yolov4 architecture [2], the input image is taken and the backbone is considered for feature extraction. Backbone comprises of algorithms like RESNET, DARKNET. It also consists of head and neck. To enhance feature discriminability and robustness, models like Path Aggregator Networks is used as neck. Dense Prediction models like SSD , YOLO , etc. form the head of the model. Considering the data from the research paper, CSPDARKNET53 is used as backbone followed by Spatial Pyramid Pooling (SPP) as neck and Yolo V3 as head of the model for optimal accuracy and mean Average Precision.

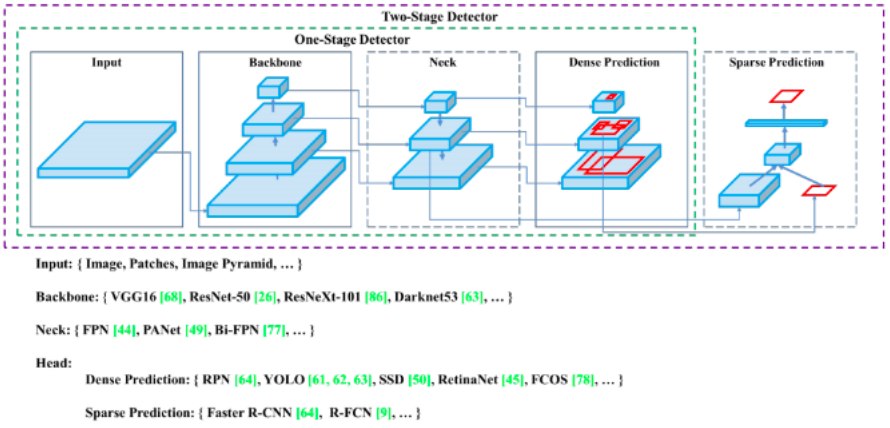


FIGURE 3.3: A picture to depict Yolov4 Architecture

3.2.3.2 Implementing YOLO in the app

COCO dataset is used for training the YOLO v4 model with the architecture explained above. The darknet weights are converted to tensorflow and the model in the form of .tflite file is obtained and used in the application. The image size is converted according to the first step in the YOLO model (for ensuring optimal predictions the size is modified) and this image is fed to the object detector which detects the list of classes and obtains the corresponding bunding boxes. The number of objects with class people is considered for counting the total number of people in the image. This is repeated for all the frames.

3.2.4 Exporting Data to CSV

The data to be written into the CSV file is the GPS coordinates and the number of people in every frame for all the frames (Frame rate of 25 per second is considered in the implementation).The extracted location data is saved after OCR extraction and the person count obtained from YOLO is written into the file frame by frame. So the csv file contains frame number, Latitude,Longitude and Person Count.

Chapter 4

Experiments and Work Done

For the first step, extracting the location data, accuracy is to be found out to evaluate the algorithm and further improvisation is to be done to improve the accuracy.

4.1 Metrics

The GPS coordinates is represented as a floating point integer which consists of digits and a comma character is also present for differentiating latitude and longitude(Latitude,Longitude).

For any recognition algorithm, error and accuracy is a very important metric. For the current problem, the various forms of accuracy calculation for analyzing the results are discussed below.

4.1.1 Digit Wise Accuracy

In this we try to find out which all digits were interpreted as different digit as characters. One common example of misinterpretation is 0 and 'O'. So this will help us find the digits misclassified and incorporate some techniques to reduce misclassification of digits.

4.1.2 Misclassifications

We try to find out misclassification of latitudes and longitudes as a whole. This will include misclassification of digits along with coma and full stop. This metric will help us look into digits and characters as a whole and analyse the results.

4.1.3 Correct recognition for a location

This is the most important metric since the exact match of GPS along with latitude and longitude gives us the accurate location otherwise, the location gets misinterpreted. This is calculated for every frame in a video and the total accuracy is calculated by considering total number of frames in all the videos.

4.2 Work Done

4.2.1 Experiment 1

For extraction, the digits were written as it is without any processing in a legible size according to the size(height and width) of the video. The entire frame was given as input to the OCR functions and digit wise accuracy was obtained for analysing and improving the model and accuracy of exact match of GPS coordinates was fount out and the results of the same is provided in the table.

4.2.2 Observations

Five sample videos were considered for calculating accuracy and all the three metrics discussed were obtained for all the samples. The overall accuracy was found to be 92 per cent as presented in figure 4.2 Some general observations in the generated data is as follows:

- The image had extra numbers apart from the GPS locations and recognition of those interfered with the recognition of GPS coordinates.

- There were general misclassification of digits as characters like 0 and 'O' , 6 and 'G', 1 and 'T' etc.
- In some cases the decimal point and coma weren't recognized and hence the coordinates couldn't be split accordingly.

4.2.3 Experiment 2

Some preprocessing techniques were carried out to improve the accuracy according to the observations being made.

- To reduce the space for recognition, we can try to segment the image(crop the image) such that we include only the location coordinate portion. For this we can assume that coordinates will be present in the top left corner of the image and use the segmented image for further analysis. This will help us in eliminating the recognition of other extra characters in the image and also save the processing time due to lesser image size.
- To eliminate misclassification of digits as alphabets, we can modify the OCR to recognize only digits , spaces and coma values. So first we convert text using normal OCR which will recognize the 'coma' and 'decimal point'. Now we split the image according to that location and use the tessaract OCR to recognize digits in the image.
- To improve recognition, we can also try adding white spaces which also helps in better clarity of recognition and in some cases, if we get spaces in the results, we can remove them manually by parsing the text.

4.2.4 Observations

The accuracy has improved from the previous implementation by following the above preprocessing techniques. The overall accuracy has improved to 95.2 percent as presented in the figure 4.2 The digit wise accuracy is represented in the figure 4.1

In general the problems encountered in the first method were addressed and removed. However, the error due to incorrect recognition of coma and decimal point couldn't be removed and those were the points that has brought down the accuracy of the algorithm.

Digit wise accuracy After Processing

Digits	Number of Misclassifications	Total Number of Digits	Accuracy
0	140	8837	98.41575195
1	216	8518	97.46419347
2	259	4832	94.63990066
3	247	7705	96.79428942
4	217	8355	97.40275284
5	287	4238	93.22793771
6	181	5593	96.76381191
7	173	3823	95.47475804
8	236	7598	96.89391945
9	190	8944	97.87567084

FIGURE 4.1: DigitWise Accuracy

Sample Number	Accuracy Before processing	Accuracy After processing
1	100	100
2	92.89	93.1
3	89.66	94.61
4	86.8	95.20
5	93.8	92.9

Average Accuracy Before Processing : **92.63**
Average Accuracy After Processing : **95.2**

FIGURE 4.2: Overall Accuracy Comparison

Chapter 5

Results and Learnings

5.1 Results

The given tasks were completed as per the requirement and an application was built for the functionalities. The image 5.1 depicts the UI of the application. The image below 5.2 depicts the embed video screen which is used for creating videos with embedded location and the image 5.3 depicts files generated with the extracted location and count number of people in the image.

5.2 Tech Stack

The main codebase was written in Android Java. Apart from this several software tools and library were widely used and familiarised with are as follows:

- Android Java

  Android studio and android java is used for mobile application development. For this case, since device location and video taking is necessary, only mobile application can be developed.

- FFMPEG Library

  FFmpeg,Fast-forward MPEG, is a free and open-source multimedia framework,

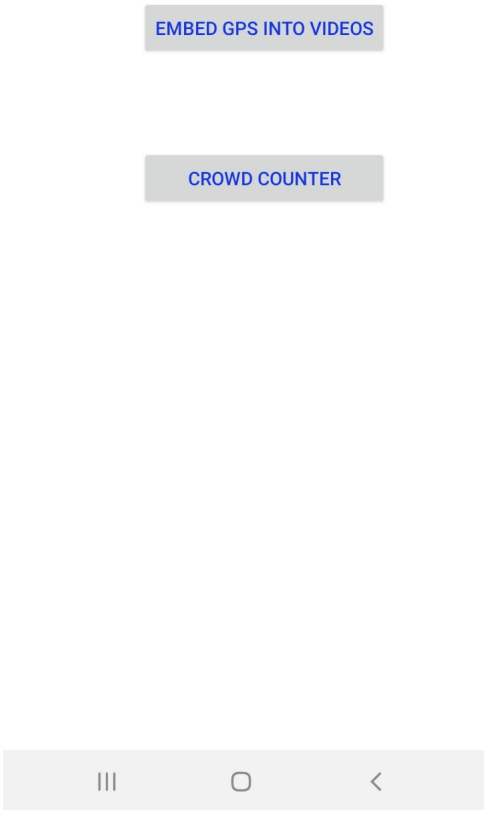


FIGURE 5.1: A picture of UI application

which is able to decode, encode, transcode, mux, demux, stream, filter and play fairly all kinds of multimedia files that have been created to date.

- Opencv and JavaCV

  OpenCV is a library for image Processing and computer vision. JavaCV wraps openCV and other image related libraries specifically for the Java programming language.

- Tessaract

  Tessaract is an open source Optical Character Recognition (OCR) Engine which is developed based on Neural Networks and available in Java and Python libraries.

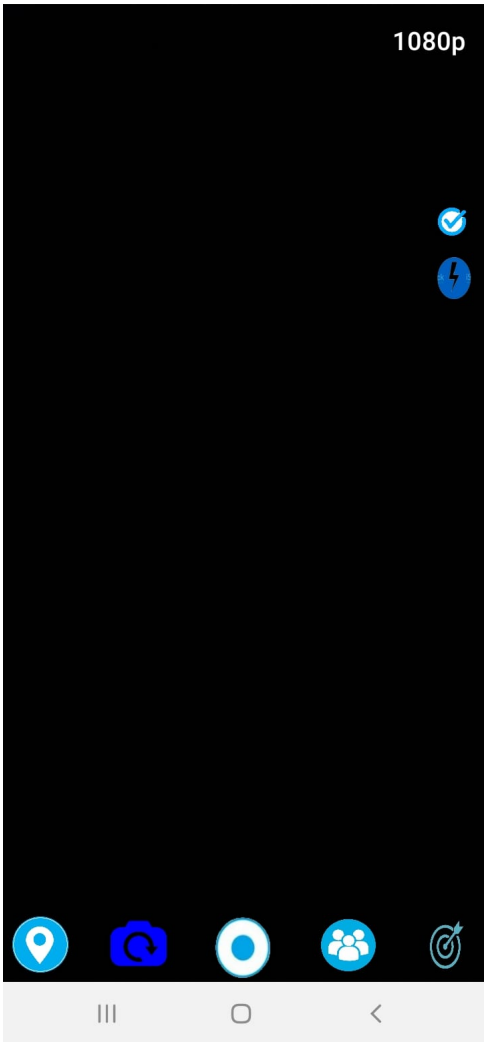


FIGURE 5.2: A picture to depict Embedding Video Screen

- TensorFlow

  Tensorflow Python tensorflow is used to generate a tflite model and tensorflow library in Java is used for loading and executing the model generated for the people counter application.

FIGURE 5.3: A picture to depict Video Files

Chapter 6

Conclusion

6.1 Limitation

The current implementation has some limitations which can be further improved upon. The limitations and the challenges are as follows.

- Since 25 frames are extracted per second , processed and then combined back to make a video, it is becoming a time consuming process. Although efficient algorithms are used for this , this is suitable only for small videos. This is not a major concern since we can make the entire process run in the background without causing any delay to the user.
- The current implementation does not restore the audio captured since we are combining the image frame. This is possible to implement by storing the audio and adding it again to the generated video. This is also not a major concern since we are analysing from an image point of view.
- The people counter algorithm doesn't detect people very far away from the camera(since they will be present as small objects in the image) since YOLO doesn't recognise small objects that precisely. This also is not a major concern since we are looking for rough estimates for a crowd analyser.

- We may also count overlapping people when we are analysing the data for a moving camera. This also is not a major point of concern since a small delta variation in the count may not matter much in case of a huge video in a densely populated area is being processed.

6.2 Use Cases

The use case scenario of the problem statement is as follows:

- This can be used for crowd management and traffic congestion where location data is helpful to find the number of people like in case of Google Maps.
- Embedding of data can be useful in case of medical data where we embed the patient details, ECG values etc. along with the captured multi media.
- The people counter application is extremely useful in case of crowd analysis where we want to estimate the number of people in a given location like finding number of people who attended a huge rally and till what time the audience were in the venue, finding number of people at a given platform in a railway station.

6.3 Future Scope

The algorithms used can be improved in terms of speed and accuracy of the predictions involved in the application built which in turn will result in a seamless experience for the user.

Bibliography

[1] B. Wang, "Research on pedestrian detection algorithm based on image," in *Journal of Physics: Conference Series*, vol. 1345, no. 6. IOP Publishing, 2019.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[3] M. Rezaei and M. Azarmi, "Deepsocial: Social distancing monitoring and infection risk assessment in covid-19 pandemic," *Applied Sciences*, vol. 10, no. 21, p. 7514, 2020.

[4] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[5] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *arXiv preprint arXiv:1905.05055*, 2019.

[6] R. Smith *et al.*, "Tesseract ocr engine," *Lecture. Google Code. Google Inc*, 2007.

[7] R. Smith, "An overview of the tesseract ocr engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2. IEEE, 2007.

[8] A. S. Agbemenu, J. Yankey, and E. O. Addo, "An automatic number plate recognition system using opencv and tesseract ocr engine," *International Journal of Computer Applications*, vol. 180, no. 43, pp. 1–5, 2018.