Traffix

Juan Buhagiar 464096(M) & Adrian Apap 422496(M)
juanbuhagiar@gmail.com & adrian.apap.14@um.edu.mt

Abstract

Traffix is an intelligent system that provides automatic live updates of the traffic situation on the Maltese roads. This systems gathers data from live video sources from around the Maltese islands and provides information regarding the traffic situation in certain roads. This data can be used for in several applications but as a starting point we have produced a map to show the consumers the current situation on the Maltese roads.

Part I
Business Aspect

1. Problem Description

Traffic tops the list of issues that maltese citizens identify as the greatest problem facing the country of Malta (MaltaToday, 2016c). Currently there are around 300,000 licensed vehicles in Malta of which around 250,000 are passenger vehicles. Licensed vehicles increased at a rate of around 28 per day in 2013/2014 (MaltaToday, 2016b). Malta has a population of around 400,000 meaning that in every 2 people, 1 owns a car. In a recent survey done on a local newspaper around 77% of people think that traffic is bad (Independent, 2016). Transport Malta has been continuously doing projects to improve the traffic situation but even the latest project still expect congestion at certain times (MaltaToday, 2016a). Factors which ensure that the traffic problem is a long term problem include the following:

- As described above the traffic situation is already in a bad situation.
- Malta has an influx of foreign nationals working in the country(LTD, 2016a), therefore since there is no long term for alternative transport (MaltaToday, 2016d), we believe that foreign nationals are more incline to buy passenger cars.
- Recent studies have put Malta as the second best place as an expat destination (MaltaToday, 2016e) which will continue to contribute to the previous point.
- A solution which other countries have used to reduce traffic is increasing cycling but since Malta has unfortunately been ranked as the worst country in the EU for cycling (LTD, 2016b), we do not expect a major increase in cyclists.

All the factors above contribute to the traffic problem currently present in Malta. Although there are also great initiatives that are trying to solve this issue we expect that traffic will be present for years to come.

2. Proposed Solution

We have designed a solution consisting of two major parts: The Traffix API consisting of the data gathering and the Traffix App which allows end-users to visualise the data obtained from the Traffix API.

2.1. Traffix API

The Traffix API is the backbone of our solution, the main aim of this API is to provide traffic information for application developers. The Traffix API targets to provide near real-time traffic information by connecting to several live streams from across the island. Once the video stream passes through our video analysis algorithm, an average speed of traffic and average car count is calculated. This is purely an automated process, only requiring the initial set-up. Since, most of the cameras are set-up in conditions that we cannot effect, the average speed is not a value which can be converted into real-world measures such as kilometres per hour but is a digital-world value of pixels per frame. Therefore, since this is different for every camera, we have studied the data to obtain a threshold to identify when traffic is present. This product mainly targets application developers which would need constant traffic information. While our current prototype only provides live information from one camera, we aspire to continue to contribute to this project to provide live information from all 14 available traffic cameras (webcams, 2016) or even install new cameras.

2.2. Traffix App

With the information obtained from the Traffix API we are able to produce different types of applications. As a starting point we have identified the most simple application that can be developed with this data, a map with the locations of the cameras and the current situation near that camera. This application continuously provides the information of the API to end-users. Therefore, users can relay on the application to always have the latest information, even in the most unexpected situations. Our current prototype will only show current information from one location but can be scaled accordingly.

3. Market Research

As described above we have identified that traffic is a problem that is of national interest, we have also identified that passenger vehicles are amongst the most used methods of transport. Therefore we understand that the major of individuals use vehicles in their daily endeavours. Since, we also know that most Maltese citizens have access to a smart phone, we believe that we are targeting the majority of the driving population of Malta.

Our research into the traffic solutions currently in the Maltese market did not yield us with a lot of competition but we have identified one main competitor that tries to target the same community. MalteseTraffic.com (Traffic, 2016) targets the same market as our Traffix App. Although, this solution targets the same audience, we have also identified that the application realise mainly on a community of users to input the data. Therefore, the solution does not provide constant real-time information. Therefore we understand that once we are able to market our product and showcase that our data sources are more reliable, we can easily infiltrate the market. Another strategy that we have identified is that we are able to sell our Traffix API, as a service, to this competitor.

Since we have already identified a precedent, there is no need to identify if users would use solutions for the traffic problem. Instead what we want to identify is if users prefer having real-time data collected via a reliable source instead of crowd sourced information.

4. Business Feasibility Study

Above we have mentioned two parts of our solution the Traffix API and App. Our plan is to monetise both products we have defined above. Firstly, we will be targeting the API towards companies which would be interested in the data we are gathering by offering the API's as a service. Meaning that we would charge companies monthly for connecting and using our API's. To gather interest in our product we are looking at the following options:

- Contact the current competitor to discuss if they would be interested in such a product.

- Use the Traffix App as a means to advertise the power of our API's.

- Contact Transport Malta, which is a government agency that deals primarily with the transport infrastructure in Malta, to discuss any possible collaboration that would help the agency to improve the current situation by using our API's.

As for the Traffix App we are targeting more the general public by making it available online for anyone to download. With a marketing campaign, we are confident that users will gain interest in the product. Our initial idea is to make the Traffix App free for our users. We would then monetise this consumer application by including advertisements within the application. This can be done by using subtle advertising rather than the usual advertisements in applications. This can be done by placing markers on local business on the map offering some relief to customers on their commute through traffic congested areas.

To further our solution we have identified a number of additional features which we can implement:

- Ability to use navigation in the map while avoiding locations which are congested with traffic.
- Learn the time users are on their daily commute so that the application can update the user with the information they require only.
- Obtain traffic patterns which could help industry and government to make more informed decisions on future projects.

5. Technical Feasibility Study

The solution we are proposing relies on the current live streams provided by Skyline Cameras (webcams, 2016). For our prototype we are using only 1 live stream, to showcase our solution. If we were to take the solution into market there are several things to consider. First of all, we have to understand the law regarding using skyline cameras as our data source. If we would proceed with Skyline Cameras as our data source we can scale our application to include the 14 traffic cameras they currently have available. Regardless if we would use Skyline Cameras as a data source, seeing that it is possible to record videos in public spaces, we can set up our own cameras across the Maltese Islands. This would enable us to identify the most problematic routes and set up cameras in such locations.

Currently we are using MATLAB which is more of a prototyping program, for the visual processing of the live stream which is not as efficient as c++, nonetheless we have found similar libraries for c++ that could enable us to redesign the system in this language. This would enable us to have a more efficient system, which could work on most machines that would allow us to process more live streams concurrently. An issue that we would need to consider is the fact that we would need hardware to run the system continuously, which would increase depending on how many streams we will be tackling. One other dependency which we would have to keep in mind when scaling is the internet connection since we would have to stream more information.

Part II
Technological Aspect

6. Technical Introduction

The solution we proposed works with a client server approach. The Traffix Server, were it processes the video stream and extracts the information needed. This information is exposed via API's that enable applications to connect to the server and use the latest information available. This can be made available to any paying customer, once we implement a security mechanism. Our client is the Traffix App which connects to the API described above and shows the latest information in a map, to indicate if there is traffic.

As shown in Figure 1, the process starts by using a Java application which obtains the M3U8 files from the skyline website (sky, 2016). These M3U8 files are then converted using the FFMPEG tool (ffm, 2016) to generate chunks of video clips where the initial setup defines the chunk video length. Then the Matlab (MATLAB, 2016) application uses the VLFeat (Vedaldi and Fulkerson, 2008) library to calculate the speed and amount of cars in the video file which is currently being processed. Finally the Matlab (MATLAB, 2016) application updates the JSON data which can be used by the clients connected through the API.

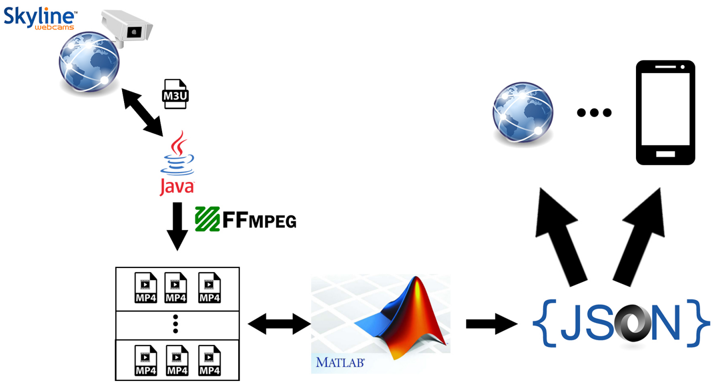

Figure 1: Traffix Process

7. Design & Implementation

Below we describe the details of the implementation for our products.

7.1. Traffix Server

The Traffix server is implemented into different parts, the first part is gathering the video files from the live stream. The current implementation saves the streams into 2 minute video segments by using the Java application and the FFMPEG tool (ffm, 2016). The other part of the Traffix server is developed in MATLAB which extracts information from the video files. The Matlab application extracts the average speed of the traffic and the average amount of cars in the video. To speed up the calculation process this information is only extracted once every second or once every 11 frames.

The Matlab application first uses a mask which has to be manually created for every camera. The mask is used to remove any extra information from each frame such as the sky. Once every 11 frames the application extracts the speed of each car using the Lucas-Kande method and using a pre-trained bag of visual words model the application extracts the amount of cars in the current frame. Finally, when the whole 2 min video is processed, the median of the average velocity and the median car count are calculated and outputted to the JSON file.

The application takes around 3 minutes to process a 2 minute video therefore to keep the application from lagging behind the real-time streams some chunks are skipped and not processed. Once a video is processed it is deleted from disk to prevent the machines memory from filling up. Currently, access to this information is done by FTP, this can be later secured so that we are able to identify different users.

7.2. Traffix App

The Traffix app is implemented in Android by using the Google Maps API. For each live stream, a marker is placed on the map. The application connects to the Traffix Server and gets the latest information from the server. This is done with the File Transfer Protocol. The information is parsed and the markers colour and text changes depending on this information. This is refreshed every few minutes so that the user is kept uptodate with the latest updates.

8. Challenges & Future Works

One of the major challenges was implementing an algorithm which could work in almost real time on the videos. This was necessary because otherwise the data being outputted by the algorithm would be lagging behind the actual traffic live stream and would not be correct. The algorithm had to output a good estimate of the current traffic speed and the amount of cars that are currently in frame. Our algorithm takes about 3 minutes to process a 2 minute video. This was achieved by performing the required calculations once every 11 frames/once every second. This can be improved in the future by writing an implementation of algorithm which uses the GPU instead of the CPU. To compensate for the extra time that our algorithm takes to extract the required information we calculate how long the algorithm takes to execute in seconds. Then the following equation is used to figure out how many video segments need to be deleted/skipped to catch up to the live stream.

$$VideosToDelete = \lceil(\frac{\left(\frac{executionTime}{60}\right)}{2})\rceil \quad (1)$$

Another challenge that we faced was that the speed that was being calculated by our algorithm was in pixels per frame. Since we did not have access to the actual physical cameras we could not adjust them to make the correct calculations to convert from pixels per frame to a usable km/h. Therefore we had to analyse various pre-recorded streams which either had traffic or did not have traffic and find a good threshold which determined if the current speed and the current car count of the traffic indicated that there was traffic in the area.

In the future we hope to implement our algorithm using more cameras or even install our own cameras from which we can generate even more traffic data. We hope to make this data available to other developers who wish to use it to develop their own Apps using the data. We could also store the generated data to generate statistics and using data mining methods to predict certain information such as the time of day which has the most traffic.

9. 3rd Party Libraries & resources

For out Application we used the following third party websites and libraries:

- Skyline Webcams (sky, 2016) - The website from which we obtained the live streams.
- FFMPEG (ffm, 2016) - Used to convert the incoming M3U8 video streams into mp4s and split the stream into segments.
- Matlab (MATLAB, 2016) - The prototyping program which we used to develop out application.
- VLFeat (Vedaldi and Fulkerson, 2008) - The Matlab library which includes an implementation of SIFT which we used in our bag of visual words model.
- Google Maps - Used to create the map in the app.

10. References

(2016).

Independent. (2016). isurvey 77.4 rate the present traffic situation in malta as bad.

LTD, A. (2016a). 31,500 foreign workers in malta.

LTD, A. (2016b). Malta ranked worst eu country for cycling.

MaltaToday. (2016a). Congestion still expected in 32 million regional road tunnel.

MaltaToday. (2016b). Maltas car population hits worrying record.

MaltaToday. (2016c). Maltatoday survey concern on traffic reaches alltime high.

MaltaToday. (2016d). Transport minister alternative modes of transport key to solving traffic problems.

MaltaToday. (2016e). Worlds best expat destinations expat insider ranks aspiring malta second.

MATLAB. (2016). *version 9.0.0.341360 (R2016a)*. The MathWorks Inc., Natick, Massachusetts.

(2016).

Traffic, M. (2016).

Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of computer vision algorithms.

webcams, S. (2016). skyline webcams malta.