MONASH University
Information Technology

GROUP ASSIGNMENT COVER SHEET

Student ID Number	Surname	Given Names
28097955	NG	NICHOLAS ZHI HAO
28457439	MATHARAARACHCHY	NIKIN GEETHILA
28196635	Goh	Si Qi

* Please include the names of all other group members.

Unit name and code	FIT3161 Computer Science Project 1		
Title of assignment	Project Proposal		
Lecturer/tutor	Dr. Wong Kok Sheik		
Tutorial day and time	Wednesday 2-5pm	**Campus**	Malaysia
Is this an authorised group assignment?	☑ Yes ☐ No		
Has any part of this assignment been previously submitted as part of another unit/course?	☐ Yes ☑ No		
Due Date 31/5/2019		**Date submitted**	31/5/2019

All work must be submitted by the due date. If an extension of work is granted this must be specified with the signature of the lecturer/tutor.

Extension granted until (date) .... **Signature of lecturer/tutor** ....

Please note that it is your responsibility to retain copies of your assessments.

Intentional plagiarism or collusion amounts to cheating under Part 7 of the Monash University (Council) Regulations

Plagiarism: Plagiarism means taking and using another person's ideas or manner of expressing them and passing them off as one's own. For example, by failing to give appropriate acknowledgement. The material used can be from any source (staff, students or the internet, published and unpublished works).

Collusion: Collusion means unauthorised collaboration with another person on assessable written, oral or practical work and includes paying another person to complete all or part of the work.

Where there are reasonable grounds for believing that intentional plagiarism or collusion has occurred, this will be reported to the Associate Dean (Education) or delegate, who may disallow the work concerned by prohibiting assessment or refer the matter to the Faculty Discipline Panel for a hearing.

Student Statement:

- I have read the university's Student Academic Integrity Policy and Procedures.
- I understand the consequences of engaging in plagiarism and collusion as described in Part 7 of the Monash University (Council) Regulations http://adm.monash.edu/legal/legislation/statutes
- I have taken proper care to safeguard this work and made all reasonable efforts to ensure it could not be copied.
- No part of this assignment has been previously submitted as part of another unit/course.
- I acknowledge and agree that the assessor of this assignment may for the purposes of assessment, reproduce the assignment and:
  i. provide to another member of faculty and any external marker; and/or
  ii. submit it to a text matching software; and/or
  iii. submit it to a text matching software which may then retain a copy of the assignment on its database for the purpose of future plagiarism checking.
- I certify that I have not plagiarised the work of others or participated in unauthorised collaboration when preparing this assignment.

Signature .... *Date* 31/5/2019

* delete (iii) if not applicable

Signature ____________ Date: 31/5/2019 Signature ____________ Date: ________

Signature ____________ Date: 31/5/2019 Signature ____________ Date: ________

Signature ____________ Date: 31/5/2019 Signature ____________ Date: ________

FIT3161 Computer Science Project Proposal

LANDMARK RECOGNITION IN VIDEOS

Nicholas Ng Zhi Hao
Nikin Geethila Matharaarachchi
Goh Si Qi

Word Count: 6416

MAY 31, 2019

Table of Contents

1. Introduction:

Vlogs, short for video blogs are the type of content that would be found in a blog, but in video form. Youtubers, which are people who upload video content on Youtube will usually upload vlogs about they every day activities or events. According to Brown (2018), vlogs are the third most viewed type of video on Youtube. Our project will be focusing on a subset of vlogs called travel vlogs which are videos that Youtubers upload about their travel experiences. According to statistic from Henderson (2018) and Youtube Data (Travel video view statistics - Think with Google, 2018), among people who are considering a trip, 64% of them watch travel-related videos and views for those kind of videos increased by 41% between 2017 and 2018.

However, a common issue faced by viewers is that most travel vlogs are long in duration which generally ranges between 5 to 20 minutes. Most viewers are only interested in a few key landmarks in the videos and do not want to watch the whole video. A method to counteract that is to have descriptions that accompany the video which lists down the landmarks visited in the vlog along with the timestamp that it occurred. Unfortunately, as a Youtuber, it is very time consuming to write proper descriptions stating the landmarks, not even mentioning the timestamps of which they appear.

Our aim with this project is to create a simple and fast method for Youtubers to add proper descriptions to their travel vlogs. In addition to that, we would also want to give the viewers an option to use our model to identify landmarks in existing Youtube videos which do not have proper captions.

For semester 2, our final goal is to create an application that will solve the issue for both the uploader/Youtuber and the viewer. Using our application, a Youtuber will be able to get a list of landmarks along with the timestamps that the landmarks appear at to be added to the description of their videos automatically, The viewers would be able to enter a URL and get the same output.

2. Literature review:

2.1. Introduction to literature review

For this literature review, due to lack of papers specific to our work, we would be looking at papers that are similar. Mainly, most papers that are published on the topic can identify landmarks in images, which is different from our goal that is to be able to identify landmarks in videos. However, in the low level breakdown of how our algorithm would work, we are fundamentally still attempting to detect landmarks in images. We would also be looking into papers related to other core functions in our project such as image augmentation, feature extraction and filtering frames.

The aim for this literature review is to be able to have a better understanding of existing works and approaches to this similar problem. We will be looking at many papers on this topic with varying approaches and decide on which approach we feel will provide the best results. In addition to that, we hope to optimize the algorithms further to improve performance and accuracy. By critically evaluating and analysing past works, we also aim to identify some shortcomings and hopefully be able to find a method to overcome it in our project to improve the overall performance.

2.2. Content

2.2.1 Traditional Approaches

The first approach we considered was using an SVM classifier. This approach has been researched by Staab et al. (2009) and Crandall et al. (2009). SVMs are short for linear Support Vector Machines and according to Crandall et al. (2009) performed slightly better than their other approach which was using Bayesian classifier. For this approach, a separate SVM for each of the landmarks are trained. To do landmark recognition, all classifiers have to be run on the image and the landmark with the highest score would be chosen as the most probable landmark in the image. They capture the visual features of an image by representing it as a vector of SIFT (Scale-invariant feature transform) features, which is a feature detection algorithm to detect local features in images. As for the textual features, only tags that appear more than twice are included in the feature vector. The study got a 79.59% accuracy when both visual and textual features were used. Li, Crandall & Huttenlocher (2009) proposed a method to cluster SIFT descriptors from photos using k-means algorithm to build a visual vocabulary.

Zheng et al. (2009) suggested an improved method which uses Laplacian-of-Gaussian (LoG) filters to detect interest points. SIFT approach is still used for local feature detection.

For efficiency purposes, instead of limiting to 1000 key points like the previous approach, their team performed a Principle Component Analysis (PCA) to reduce the feature dimensionality to 40. Another improvement in their research is that instead of directly attempting to cluster the data, they constructed a match region graph by performing matching on images where the vertexes are match regions. Finally, clustering is performed on match regions to find regions of same or similar landmarks. Using their methods, they found that the accuracy has been increased to 80.8%.

Although traditional approaches have decent accuracies, with the use of ANNs it is possible to improve on this and therefore we will using ANNs.

2.2.2 Modern Approaches

We have noticed that there are 3 main models that we can use as base models to use transfer learning in order to enable them to detect landmarks, namely the VGG-16 network, ResNet and Xception. We found that there has been contradictory research in the past showing that depending on the dataset used each of the networks perform differently. As analysed by F. Chollet , (Chollet, 2017) when tested for the JTF dataset t for large-scale image classification dataset introduced by Hinton et al in (Geoffrey Hinton, 2014) the Xception network performs the best at image classification with a accuracy of 94.5%. In contrast K. He et al mentions in (Kaiming He, 2016) that when various versions of ResNet and VGG-16 are run on PASCAL VOC 2007 and 2012 (Mark Everingham, 2010) and COCO (Tsung-Yi Lin, 2014) datasets, "we obtain a 6.0% increase in COCO's standard metric (mAP@[.5, .95]), which is a 28% relative improvement." With ResNet-101 when compared to VGG-16.

However A. Nygaard in (Adil Nygaard, 2018) tests the three networks on the google landmark dataset and observers that VGG-16 performs the best achieving a test accuracy of 99% whereas its ResNet50 and Xception obtain 98% and 90% respectively. As we too are using the same dataset we shall use the VGG-16 model as, even though it performs worse on general image classification has been shown to perform superior in landmark detection.

Moreover, We may decide to use a Compressed residual-VGG16 as well depending on the performance as it has been shown to be "88.4% smaller in size and 23.86% faster in the training time" (Hussam Qassim, 2018) when compared to the generic VGG-16. However this needs to first be tested on the landmarks dataset to confirm its performance does not diminish when compared to the generic VGG-16.

2.2.3 Image Augmentation

Image augmentation may be required to ensure that the images are able to be properly detected by the model that we use. As mentioned by L. Perez (Luis Perez, 2017) traditional augmentation such as cropping, rotating, and flipping input images is very effective alone, and that while other techniques using a Generative adversarial network enabled by CycleGAN and other similar networks produce good results, do not perform much better than traditional methods and take up almost 3x the time or more to compute.

We could use the Albumentations library (Alexander Buslaev, 2018) which has been shown to provide features for fast and flexible image augmentations with many traditional image transform operations available.

2.2.4 Feature Extraction

We noticed that one of the man ways to extract features of an image to classify it it by using visualization techniques that perform qualitative analysis on the image. One of the possible approaches iis to use a activation maximization using gradient descent to the image space to maximize the units activation (Dumitru Erhan, 2009). Another method is to use occlusion which uses density to check if the network is accurately classifying the object being classified or simply its surrounding context (Fergus, 2014). R. Fergus also suggests that deconvolutional networks could be used to visualize what patterns activate each unit and B. Zhou in (Bolei Zhou, Object Detectors Emerge in Deep Scene CNNs, 2014) mentions that the CNN's used for scene recognition could be used for object-localization as well. However, in (Bolei Zhou, Learning Deep Features for Discriminative Localization, 2016) it is mentioned that these methods are not as accurate as they ignore certain layers. Another method proposed by X. Shen et al (Xiaohui Shen, 2012) is the use of a spatially-constrained similarity measure and k-NN re-ranking which generates matches based on the neighbours of a point and as mentioned by X. Shen et al, by using a voting method the SCSM outperforms other spatial models in image retrieval and localization.

Alternatively, we could use the DeLF(Deep Local Features) architecture, which is a CNN based extractor that extracts the highest scored features. In (Hyeonwoo Noh, 2017) H. Noh et al shows that it is a good method to retrieve features and that the DeLF architecture aids in preventing false positives. We have identified DeLF as the best approach due to this feature.

2.2.5 Blur Detection on Images

Edge Detection

Another possible approach developed by Ong et al. (2003) to detect blurred image is by detecting the edges in an image which can be considered as a gradient between adjacent pixels. If the image is blurry, the contrast between the edges will be low. The blur value can be calculated using the average edge-spread value of all the edges in the image. One of the downside of this method edge spread is independent of their orientation, which it would be sufficient for optical blur but not for motion blur. However, our project involves travel vlogs where there would be a lot of blurriness occured due to the motion while recording the video. Thus, this method is not suitable for our project.

Frequency Analysis

Another alternative is to analyse the image directly in the frequency domain instead of using the edges. High frequencies do not appear in blurred images (Liu et al., 2008). Thus, the percentage of high frequencies in blurred images will be far lesser. By using this method, the absence of the high frequencies can be used to detect blurry images. However, the weakness of this method is that it takes too much computation time where we would not want this to happen because we need to keep the processing which is also the matching time at the minimal.

Laplacian operator and OpenCV

As our project requires us to filter out all the unwanted blurry image frames extracted from the videos, we need to have a fast image quality assessment in terms of blur. This approach has been researched by R Bansal(2016), G Raj (2016) and T Choudhury (2016), where they are using Laplacian operator and OpenCV library. The Laplacian operator is being implemented from the OpenCV to perform computation by using gradient of images. Taking the image as single channel, the image in convolved with a 3x3 Laplacian kernel. The variance of the Laplacian kernel will be used to check for edge-like or non-edge-like responses by squaring the standard deviation. Then OpenCV library will be used for blurry detection. We will assume that a blur image will have very little edges in the image, thus if the variance falls below a pre-defined threshold, the image is considered as blurry. This method is simple to understand and be implemented with OpenCV library.

As filtering out the blurry images is not our main concern, which it is just an optimization for our final model, we decided to go for this simpler method.

2.2.6 Detecting Similar Images

Hash Code Grouping

One of the approach proposed that the image table will have group of images separated according to their most significant elements of hash codes representing the image (US Patent No. US 7,647,331 B2, 2010). The detection system will then generate a target hash code for target image, which will be matched to the group of the target image by the identifier. The images associated with those similar hash codes will then be selected by the detection system as duplicates of the target image.

Perceptual Hash and Usage of Image Key Points

Another alternative method would be first building a descriptor with identifying key points of the image with dimensions mxn which is proposed by Korsunov N. et al (2015). Perceptual hash is being used to describe the key points, where generation of hash can be used to compare the degree of difference between two images. Note that the Hamming Distance between the reference image and the possible duplicate image would be able to access the similarity of the images.

In conclusion, this proposed method has a true-positive percentage of 89.5% compared to existing known algorithm which has a percentage of 91.5%. However, in terms of the processing speed of the image, the proposed method requires 0.31c, meanwhile the known method requires 1.95c. This indicates that the proposed method can process the images with 6 times of the speed for the known method.

2.3. Conclusions drawn from the literature review

In conclusion, having analysed several approaches for landmark detection in images, we have decided to use the modern approaches, specifically the VGG-16 model with transfer learning as we found that it consistently obtains the highest accuracy in such tasks. However, we will try to combine techniques in different approaches to attempt to improve the accuracy and performance further. For image augmentation, we would be using the Albumentations library for its simplicity for our purpose. For feature extraction, we decided to use the DeLF (Deep Local Features) architecture for its superiority in preventing false positives. In order to filter blur images, as it is not the main focus of our project, we be using the Laplacian operator and OpenCV mainly due to its simplicity. Similar images would be detected using perceptual hash and usage of image key points due to its performance.

3. Project Management Plan

3.1. Project overview:

With the increased use of video content and the high rate of growth of platforms such as youtube, there is a need and a possibility for the use of machine learning techniques to improve the ease of use of such applications.

One such improvement that we identified is the use of image recognition techniques to tag landmarks within videos along with the time of the landmark being shown within the video. This can be used with great effect on travel vlogs, travel montages, etc. And can be used to aid both the person uploading the video by adding captions with the landmarks and their times as well as viewers that wish to find the landmarks shown in a video which is not properly annotated.

The first major milestone for our project would be the completion of the training process of our model. That would be followed by the confirmation that our model achieves a satisfactory level of accuracy based on our testing. After that, we would be looking towards completing our mobile application and testing it. Finally, we would be doing a review and report on the entire project.

3.2. Project Scope

3.2.1. Project deliverables

Project deliverables are the output of a team to guide towards the successful completion of that project, where it can be separated into project management related deliverables and product related deliverables.

<u>Project management-related deliverables:</u>

1. Schedule

Project scheduling is a mechanism to communicate on the allocation of organisational resources on which specific tasks required to be done in certain timeframe. This is to ensure that key milestones are met on time and that the entire project completes within the 6 months.

2. Status reports

Project status reports are the main medium of communications between the project current progress and team members, stakeholders, clients. While informing involvers about the budget, achieved targets, completed work, risks can be identified. These reports provide transparency about activities within the project and show how the project has progressed over time.

3. Final project report

Final project report is used to show the projects implementation and final status, while evaluating how did the project performed. The document must include what have been delivered by the project team, evaluation of work quality during the project and budgeting and scheduling performance.

4. Scope statement

Scope statement can be used to identify the scope of the project at the start and confirm the expected results of the project. It will be created before the project starts. The statements should be agreed by everyone who involved in the project.

Product-related deliverables:

1. Proposal

Proposal is a document that shows the objectives and details of the project at the start, which includes the strategic objectives, benefits, success criteria, budget plan and milestones to be reached.

2. Design documents

Documents that show the design of the project and the way in which the requirements will be met. For example, the design of the user interface should be drafted beforehand.

3. Project Progress Report

A report that will show the progress of the project at different times. This is to ensure every team members are on the right track and have a clearer view on the current progress in order to proceed.

4. Product Prototype

A model project should be built to be used for testing purposes. A testing model should be prepared to check the accuracy of the training and matching results before the actual model is being implemented with the mobile application.

5. Technical Interpretation

The technical interpretation shows the implementation of the frames matching processing in the mobile application.

6. Software code

The code written for the mobile application in order for the implementation at the back end to work.

3.2.2. Product characteristics and requirements, functional and non-functional requirements

Characteristics and requirements:

1. Able to check the progress bar while the video is being uploaded.
2. Able to view the video while it is being uploaded.
3. Check the timestamps and landmarks of the video by providing with the Youtube URLs.
4. Edit the generated results.
5. Upload the videos directly to users’ Youtube account.

Functional requirements:

Describes the behaviour of the system which is supposed to be accomplished

1. Data processing

   The frames extracted from the videos must be filtered and left with the least amount of highly usable frames by taking those who has the highest confidence score on certain landmarks.

2. Data between servers and users

   The videos uploaded by the users to be processed can be directly uploaded to their desired social media, with the timestamp results appended as the description or captions

3. Technical details

The video being processed should have a result text containing all the timestamps with their corresponding landmarks.

Non-functional requirements:

Elaborates a performance characteristic of the system

1. Efficiency

Time taken for the system to retrieve the preprocessed data from database and process the videos must considerable and similar to the upload speed of the entire video onto social media.

2. Availability

The mobile application should be available 24/7 with no down time if possible as there will be users from all over the world and they might want to upload their videos whenever they want to.

3. Extensibility

The processing on videos is not just limited on Youtube videos only, it can be extended to process all kind of videos on different social media platform.

4. Reliability

The users who uploaded the videos onto the application must link their social media account to the mobile application itself. The security of the uploading process will rely on the social media itself.

3.2.3. Product user acceptance criteria

Product user acceptance criteria conditions that a software product must satisfy to be accepted by a user, customer or other stakeholder. These criteria states that the boundaries of a feature and determine whether the project can work as expected.

1. The processing time for a video

The time taken to process an uploaded video should be acceptable and considerable to the users as they would not want to spend too much time on waiting the video to be processed.

2. The types of format of videos

The mobile application should accept a wide range of video format that will be uploaded onto the server to be preprocessed. The trained model should be able to handle different kind of videos by using the same methodology.

3. User friendly mobile application

he features in the mobile application should be clear and easy to use. Users should be able to identify the parts where they need to follow even though they do not understand the technical parts of the process.

4. The ability of user to edit the results generated

Users will be able to edit the description part after the video is being processed. If they found out some part of the result is wrong or not in their desired way, they can edit it before uploading to the social media.

3.3. Project Organisation

3.3.1. Process Model:

The lifecycle model we will be using for this project will be agile. We chose an agile approach as the main advantage of it is the ability to calculate value and risk overtime (Kerzner, 2017). The agile model requires us to do frequent assessments and measurement of value. This will give us a better indication of the progress of project as compared to a traditional approach such as waterfall. By keeping track of the progress, we can better predict if the project will achieve its goal within the desired timeframe and if not, adjust our schedule or approach accordingly to get the project back on track.

Besides, the agile model is also able to work with less specific requirements at the start of the project as if changes in requirements occur in any phase of the project, we can just go back to the specific phase and make the appropriate changes. This is important to use because as of this point of time, there are still many uncertainties in our project. For example, we have many variables such as the time interval between each frame, the number of layers and Epochs in our model of which the values have not been determined. We are unable to be certain of those values as of now as the technique we use to implement this project will change overtime according to what we find is the most balanced in terms of time taken and accuracy. Having the flexibility to go back to the previous phase in agile will allow us to go back to the previous phase if for example, after testing, we found that our model did not produce results with an acceptable accuracy level.

Furthermore, the agile model allows us to adapt the project to change (Gilley, 2015). As this is the first time we are doing a project of this type, we will definitely make mistakes along the way. Using the agile approach, if a mistake is found, we could simply go to the previous phase and fix it without much disruption in the overall flow of the project. The agile model will also produce an overall higher quality product as it is developed in small increments and those increments are tested thoroughly. A mistake can be found much earlier and sooner than a traditional waterfall approach which makes fixing the issues simpler.

3.3.2. Project Responsibilities:

Stage	Function/Activity	Time	Person Responsible
Training data sets	Building the Model	Week 1	Nikin
	Training the data sets	Week 2-3	Nicholas and Si Qi
Matching and Evaluating data sets	Code to filter unwanted images	Week 4	Nicholas
	Optimization of code	Week 4-5	Nikin and Si Qi
	Testing of trained model	Week 6	Nikin and Si Qi
Mobile Application	Building mobile application	Week 5-6	Nicholas
Testing	Manually identify landmarks in videos	Week 7	Nicholas and Si Qi
	Testing on the mobile application	Week 8-9	Nikin
Project Review and Reporting	Documentation	Week 10	Nicholas
	Analysis of result	Week 10	Nikin

	Comparison with initial proposal	Week 11	Si Qi
	Conclusion	Week 11	Nicholas

3.4. Management Process 2

3.4.1. Risk Management:

For risk management techniques, our team already had a team meeting to brainstorm potential risks that might arise from the project. This session was facilitated by Nikin as he is most knowledgeable among the members in the field of deep learning.

Besides, SWOT analysis, which is short for “Strengths”, “Weaknesses”, “Opportunities” and “Threats” will be applied on the project. By determining strength and weaknesses, we can identify risks in the context of opportunities and threats. For example, a current weakness of our project is the lack of understanding of market demand, which might cause the threat that the users might not accept our product.

Furthermore, we would be interviewing stakeholders and potential users through means such as email, phone and instant-messaging. This will allow us to hear their perspective on our current project and potentially identify risks from them. Interviews with people experienced with a similar project might also help us identify risks that they may have went through.

The risk register is shown as Appendix A.

3.4.2. Monitoring and Controlling Mechanisms

3.4.3. Communication and reporting plan:

Communication	Medium	Frequency	Goal	Audience
Daily updates	Meetings or Email	Daily	Share daily progress made on tasks	Team
Project status report	Email	Weekly	Review status of project and discuss issues or delays	Team
Milestone Review	Meeting	At major milestones	Presentation and delivery of major deliverables	Team and sponsor
Post-mortem meeting	Meeting	At the end of the project	Assess finished work and discuss mistakes and what we can learn from	Team

3.4.4. Review and audit mechanisms:

For versioning control, we would be using github to keep track of the history of our code. Each member would be assigned a different part of the project and will push their work daily to their own branch. Periodically, their branches will be assessed by the other members and if it is accepted, will be merged to the master branch.

For general communication purposes, a whatsapp group will be used. For more work specific tasks, our team will be communicating via Slack. The team members will give updates on the progress of their work on Slack as well as bring up some issues, concerns or decisions to be made to the team.

For cloud storage, a google team drive will be created to store all the miscellaneous files to be made accessible by all members in the team. Google Drive also comes with integration with Lucidchart which will allows us to build diagrams if needed.

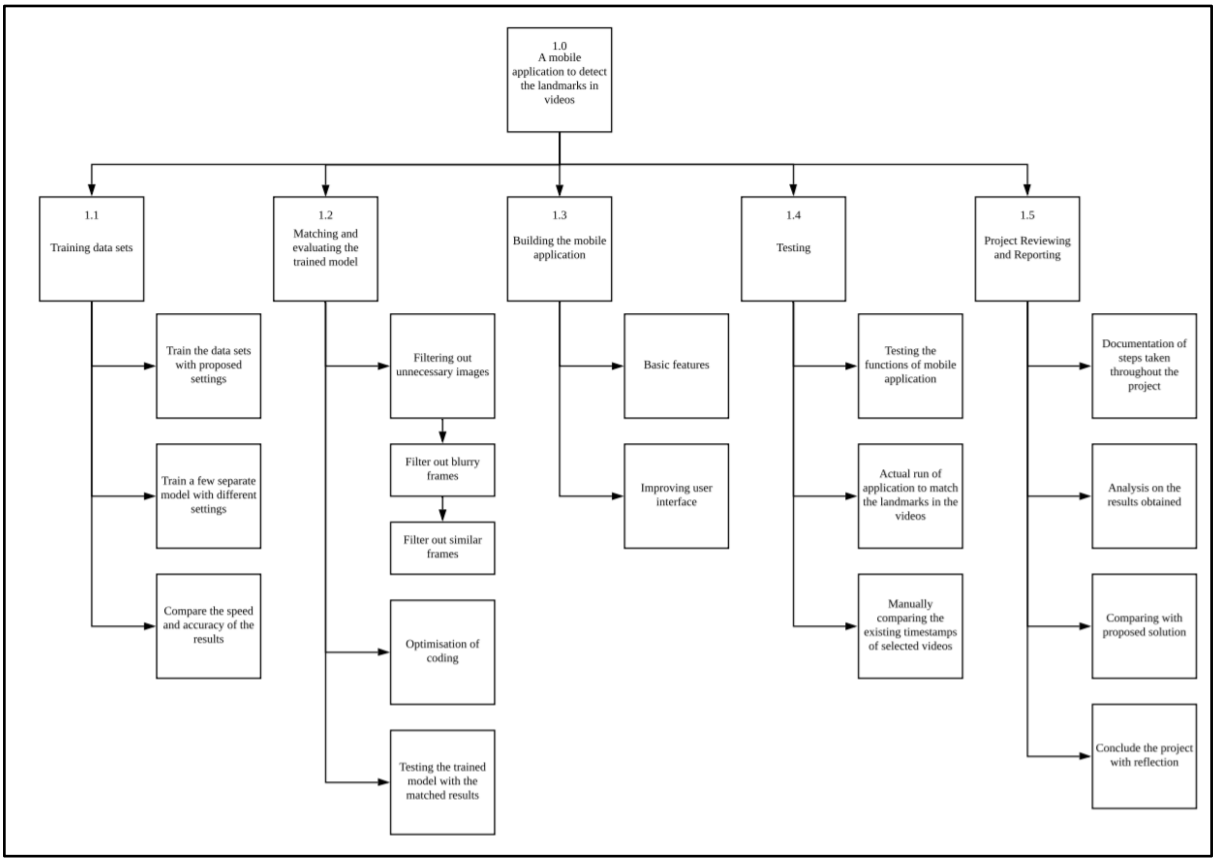
For training, all team members will be taking the “Introduction to deep learning” course on Lynda to provide a basic understanding of the project scope to all members.

3.5. Schedule and Resource Requirements

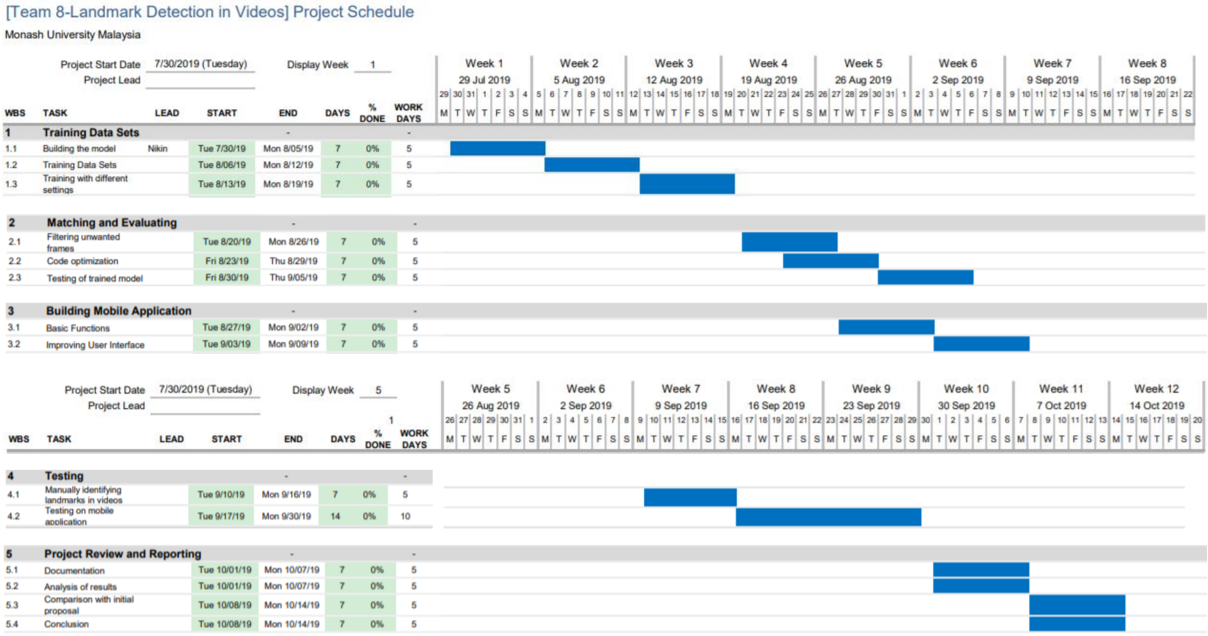
3.5.1. Schedule:

The flowchart below shows the overview of all the subtasks required throughout the project, which is divided into 5 main stages as shown:

1. Training data sets
2. Matching and evaluating the model
3. Building a mobile application
4. Testing the application
5. Project Review and Reporting



Project Schedule:



[Team 8-Landmark Detection in Videos] Project Schedule

Monash University Malaysia

Project Start Date 7/30/2019 (Tuesday) Display Week 1

Project Lead

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1	**Training Data Sets**			-			-
1.1	Building the model	Nikin	Tue 7/30/19	Mon 8/05/19	7	0%	5
1.2	Training Data Sets		Tue 8/06/19	Mon 8/12/19	7	0%	5
1.3	Training with different settings		Tue 8/13/19	Mon 8/19/19	7	0%	5
2	**Matching and Evaluating**			-			-
2.1	Filtering unwanted frames		Tue 8/20/19	Mon 8/26/19	7	0%	5
2.2	Code optimization		Fri 8/23/19	Thu 8/29/19	7	0%	5
2.3	Testing of trained model		Fri 8/30/19	Thu 9/05/19	7	0%	5
3	**Building Mobile Application**			-			-
3.1	Basic Functions		Tue 8/27/19	Mon 9/02/19	7	0%	5
3.2	Improving User Interface		Tue 9/03/19	Mon 9/09/19	7	0%	5

Project Start Date 7/30/2019 (Tuesday) Display Week 5

Project Lead

WBS	TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
4	**Testing**			-			-
4.1	Manually identifying landmarks in videos		Tue 9/10/19	Mon 9/16/19	7	0%	5
4.2	Testing on mobile application		Tue 9/17/19	Mon 9/30/19	14	0%	10
5	**Project Review and Reporting**			-			-
5.1	Documentation		Tue 10/01/19	Mon 10/07/19	7	0%	5
5.2	Analysis of results		Tue 10/01/19	Mon 10/07/19	7	0%	5
5.3	Comparison with initial proposal		Tue 10/08/19	Mon 10/14/19	7	0%	5
5.4	Conclusion		Tue 10/08/19	Mon 10/14/19	7	0%	5

3.5.2. Resource Requirements:

Tasks	**Resources Type**	**Resources Details**
Training Data Sets	1.Equipment	A virtual machine with appropriate specification to carry out model training Provided by: Amazon EC2 Virtual Server Operating System: Linux GPU: At least K80 and above CPUs: 4 or above RAM: 61GiB
	2.Technologies	i. Programming Languages **Python** because it is a high level language for machine learning and easier to use ii. Platform Pycharm Community Edition

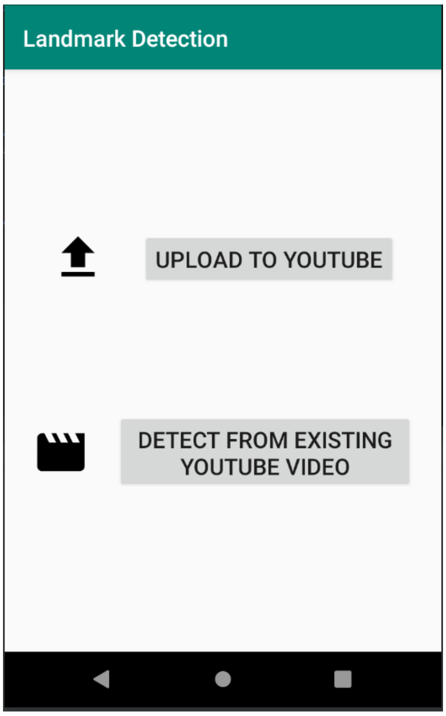
		iii. Libraries **Keras** allows fast experimentation of neural networks and modularity **TensorFlow** will be used on building models by using Keras with eager execution to achieve immediate model interaction. **HTTP** which is the Python library to implement the GET and POST method to obtain information from the server side **Google API Client Libraries** offer a simple and flexible access to Google APIs iv. Database **Oracle Database** is used to store the extracted features of each image frame **MySQL** is used for Relational Database Management System v. Server Side Language **PHP** is used to establish a connection between the server and users
	3. Materials	**Google Landmarks Data Sets** which contains 2 millions images, with 30 thousand unique landmarks
	4. Time required	Building the initial model: one week Training the data sets: one week Training with different settings: one week Total: 3 weeks

Matching and Evaluating Data Sets	1. Technologies	i. Libraries **OpenCV** to optimize the code and filter out similar and blurry frames which will will then be sent to be trained **pandas** in Python will be used to manipulate the data for a clearer presentation **matplotlib** will be used for 2D plotting **numpy** is used as an efficient multi-dimensional container of generic data and to seamlessly and speedily integrate with a wide variety of databases.
	2. Time required	Filtering unwanted image frames: one week Optimization of code: one week concurrently Testing of trained model: one week Total: < 3 weeks
Building the mobile application	1. Technologies	i. Programming Languages **Java** will be used for building mobile application because it is the native language for an mobile application ii. Platform Android Studio
	2. Time required	Basic functions: one week Improve user interface: one week Total: 2 weeks
Testing the application	1. Technologies	Uses **Youtube API** to add a function into the mobile application to be able to upload the

		videos directly from the mobile application to the Youtube platform **Apache HTTP** will be used for connection purpose between server and users
	2. Time required	Manually testing the results with selected videos: one week Testing the mobile application: one week Total: 2 weeks
Project Review and Reporting	1. Labor	All team members should keep track of the progress throughout the entire project. A record of important decisions and steps taken should be maintained over the period of the developing process.
	2. Technologies	**Pro WorkFlow** as the project management tool which allows team members to log their progress and keep track of it efficiently
	3. Time required	Documentation and analysis: one week Comparison with proposed results and Conclusion: one week Total: 2 weeks

4. External Design:

4.1 Main User Interface



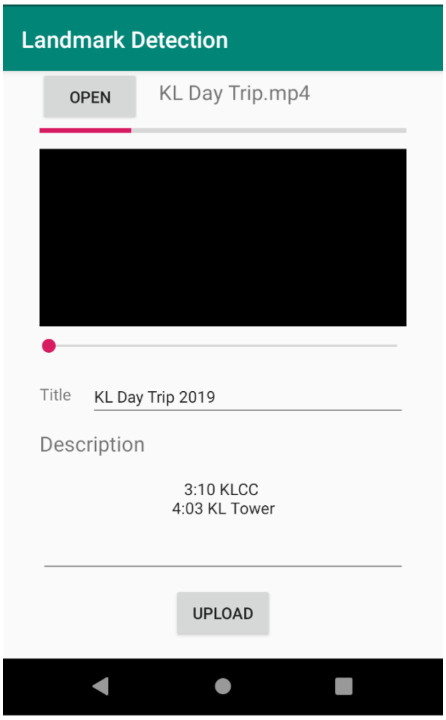
When the user first opens the application, he or she would first be shown this screen. The user will have the option to either:

(a) Upload a new video to Youtube

(b) Detect landmarks from existing Youtube Videos

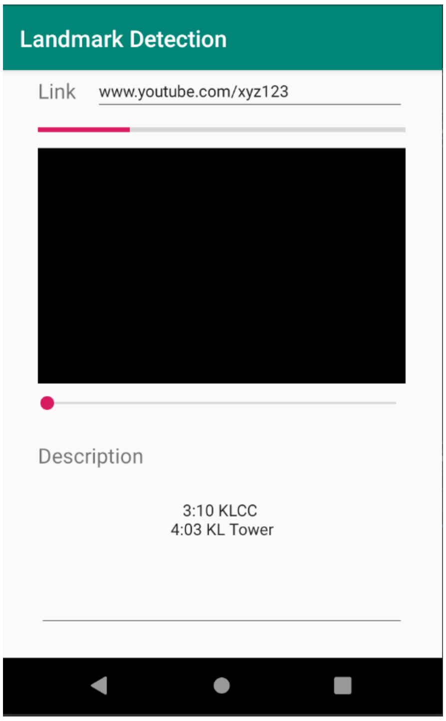
Clicking on one of the buttons will bring the user to the respective page.

4.2 Upload to Youtube Page



This page will appear when the user selects the first option (Upload to Youtube). Here, the user can select a file by clicking the “OPEN” button. Once selected, the video will be processed by our servers and the progress of the processing can be viewed in the progress bar below. Below the progress bar, the uploader can view the video in the video player below. Once the processing is complete, the uploader will be presented with a list of detected landmarks along with the timestamps of which they appear in the description box below. At this point, the uploader can choose to edit the Title and Description before clicking the “UPLOAD” button to upload it straight to Youtube.

4.3 Detect From Existing Youtube Video Page



This page will appear when the user selects the second option “Detect from Existing Youtube Video”. Here, the user can input a URL for an existing Youtube video. Similarly, a progress bar below will indicate the progress of the processing by our server. If the video has been previously processed and stored in our server, the description can be retrieved immediately without processing. Below that, the user can also view the video while it is processing. Once the processing is done, as before, the list of landmarks and the respective timestamps of when they appear will be listed below. The user can then click on any of the timestamps to directly seek to that position in the video.

4.4 Performance

The main part that will be the bottleneck for performance in our application would be the processing time of the video. Therefore, the main goals of our application is to be able to process the video in an acceptable amount of time so that the application is actually practical to use.

In terms of space, the application should not take more than a few megabytes as the model will be stored at our servers and not the device. The only space that will be used is for the user interface.

5. Methodology:

5.1 Dataset

There are multiple landmark datasets that we can choose from including the Google-Landmarks dataset (google, 2019) and the Google-Landmarks-v2 dataset (Weyand B. C., 2019). However as the Google-Landmarks-v2 has over 200,000 unique landmarks, it may be more time-consuming for the training and testing of the model and therefore we decided to use the more moderate Google-Landmarks dataset which contains more than 2 million images depicting 30 thousand unique landmarks from across the world. (Weyand A. A., 2018) In the interest of time we will use scaled down images with a fixed resolution.
This data will be then broken down into 70:30, train to test ratio using a package such as scikit-learn (Fabian Pedregosa, 2011)

5.2 Frame extraction from video

We will be extract frames at 25 ms intervals from the video. This may be subject to change based on the time the models take to run through a query frame. These frames will then be run through a code that uses OpenCV that firstly removes blurry frames (Rosebrock, Blur detection with OpenCV, 2015) and secondly removes consecutive frames that are similar using perpetual hashing and the use of the hamming distance of a descriptor with key points in the image (Nikolai I. Korsunov, 2015)

5.3 Image augmentation

The images in the dataset as well as from the frames of videos are taken at various angles, horizon shifts, zoom etc. To account for these variations, we need to perform image augmentation which will aid in image classification (Luis Perez, 2017). (Image augmentation, 2019) (Agnieszka Mikołajczyk, 2018)

5.4 Model generation

First we will select a portion of the train and test data sets and create a small dataset using which we shall generate models and test them with various variables. Once the optimal model variables have been identified, we will generate the model using the full dataset

5.5 Train strategy

We will use the VGG16 network (Zisserman, 2015) (Qassim, Verma, & Feinzimer, 2018)that is pre-trained on the Google ImageNet dataset (Olga Russakovsky, 2015). Then we shall use transfer learning to train the model for landmark detection in order to improve its efficiency.

(Marcelino, 2018) We aim to change multiple variables such as the number of layers, optimizers, image augmentation, batch size and the number of Epochs in order to obtain the best possible model for the landmark detection. (Adil Nygaard, 2018) In order to prevent overfitting we may need to use a dropout which would drop nodes below a certain probability. (Nitish Srivastava, 2014)

5.6 Image classification using DeLF (Deep Local Features)

The CNN based local-features extractor architecture (Araujo, 2019) will pick the features with the highest score and then match those features with those of the database of images in the trained set for geometric verification using improved random sample concerns (Rahul Raguram, 2012) to make the decision. (Hyeonwoo Noh, 2017).

5.7 Query result return

The model will generate the landmark with the highest probability and return it along with its timestamp to the application.

5.8 Data management post conclusion of model query

The timestamps and landmarks along with the url of the video will be stored in a database based on a hashing of the url using the hashbytes function (Ali, 2014) in order for easy extraction if the same video is to be tested in the future.

5.9 Version control

For version control we will be using git on top of the Jupiter notebooks (Schmüdde, 2019) in order for the team to be able to work on the project together.

5.10 Overview

Appendix B shows the method in which the individual components of the application will work with each other.

6. Test Planning

6.1. Test coverage

Stage	Goals
Model Optimization	● The model selected is the optimal ● Uses small dataset
Final Model Testing	● The final model gives satisfactory results for large dataset ● Model works without bugs
Application testing	● Test if user UI works as intended ● Test if user can upload image ● Test if url used properly ● Test if output of application is accurate ● Uses dummy model
Integration testing	● Test entire application along with model ● Ensure that application works as intended and is well integrated with the model
Final evaluation	● Test entire application along with model ● Checks if application provides accurate results for videos tested

6.2. Test methods

6.2.1 Global Average Precision (GAP)

Using the test dataset we evaluate using the Global Average Precision (GAP) where for each query image, we predict one class and a confidence score which is then averaged over the list. This will be used to test the models trained with the small dataset following which the optimal parameters will be used in generating a model trained using the large dataset which will intern be tested in a similar manner.

6.2.2 Unit Testing

When testing the application, unit testing could be performed to ensure it works as required. By testing if the individual components of the application behave as intended.

6.2.3 Manual Testing

Once all stages are complete, application testing we will test the entire application using a set of videos and manually test them to see if they have the correct results.

6.3. Sample Test Cases

Stage	Tests
Model Optimization	● Uses Test data set of the small dataset to generate GAP
Final Model Testing	● Uses Test data set of the large dataset to generate GAP
Application testing	● Uses valid URL to check if it works as intended ● Uses invalid URL to test for error popups ● Uses upload video to test if video uploaded to application accurately ● Test upload button that tests upload to youtube functionality
Integration testing	● Uses URL and video to ensure application runs without errors
Final evaluation	● Uses selected URLs to generate timestamped landmarks and then manually check if they are accurate.

7. References:

Adil Nygaard, U. N. (2018). *Final Project. Google Landmark Recognition Challenge.* stanford university.

Agnieszka Mikołajczyk, M. G. (2018). Data augmentation for improving deep learning in image classification problem. *2018 International Interdisciplinary PhD Workshop (IIPhDW)*.

Alexander Buslaev, A. P. (2018). Albumentations: fast and flexible image. *arxiv*, 1809.

Ali, A. (2014, 05 24). *Getting Started with Hashing in SQL Server*. Retrieved from databasejournal: https://www.databasejournal.com/features/mssql/getting-started-with-hashing-in-sql-server.html

Araujo, A. (2019, 04). *DELF: DEep Local Features*. Retrieved from github: https://github.com/tensorflow/models/tree/master/research/delf

Bolei Zhou, A. K. (2014). Object Detectors Emerge in Deep Scene CNNs. *arXiv*.

Bolei Zhou, A. K. (2016). *Learning Deep Features for Discriminative Localization.* Computer Science and Artificial Intelligence Laboratory, MIT.

Brown, C. (2018). *Here Are the Top 10 Most Popular Types of Videos on YouTube*. Retrieved from mag.octoly.: https://mag.octoly.com/here-are-the-top-10-most-popular-types-of-videos-on-youtube-4ea1e1a192ac

Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. *CVPR*, 1251-1258.

David Crandall, L. B. (2009). Mapping the World's Photos. *18th international conference on World wide web* (pp. 761-770). Madrid, Spain: WWW .

Dumitru Erhan, Y. B. (2009). *Visualizing Higher-Layer Features of a Deep Network.* Montreal: Universite de Montreal.

Fabian Pedregosa, G. V. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* , 12.

Fergus, M. D. (2014). *Visualizing and Understanding.* New York: Dept. of Computer Science,New York University.

Geoffrey Hinton, O. V. (2014). Distilling the Knowledge in a Neural Network. *NIPS 2014 Deep Learning Workshop* (p. 1503). arXiv.

Gilley, C. (2015, 03 12). *The Pros and Cons of Agile Product Development.* . Retrieved from community.uservoice: http://community.uservoice.com/blog/the-pros-and-cons-of-agile-product-development/

google. (2019). *google-landmarks-dataset.* Retrieved from kaggle: https://www.kaggle.com/google/google-landmarks-dataset

Henderson, P. (2018). *Video Marketing in Travel - 10 Key Insights.* Retrieved from markgrowth: https://blog.markgrowth.com/video-marketing-in-travel-10-key-insights-a64fdcffa465

Hussam Qassim, A. V. (2018). Compressed residual-VGG16 CNN model for big data places image recognition. *8th Annual Computing and Communication Workshop and Conference (CCWC).* Las Vegas, NV, USA: IEEE.

Hyeonwoo Noh, A. A. (2017). Large-Scale Image Retrieval With Attentive Deep Local Features. *The IEEE International Conference on Computer Vision (ICCV)* (pp. 3456-3465). IEEE.

Image augmentation. (2019). Retrieved from peltarion: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/image-augmentation

Kaiming He, X. Z. (2016). Deep Residual Learning for Image Recognition. *CVPR*, 770-778.

KanjarDeV.Masilamani. (2013). Image Sharpness Measure for Blurred Images in Frequency Domain. *Procedia Engineering*, 149-158.

Kerzner, H. (2017). *Agile/Scrum Project Management. Project Management Case Studies (5th Edition).*

Luis Perez, J. W. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *stanford.edu*.

Marcelino, P. (2018, 10 23). *Transfer learning from pre-trained models*. Retrieved from Towards Data Science: https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751

Mark Everingham, L. V. (2010). The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* , 303-338.

Mingjing Li, B. W.-Y. (2010). *US Patent No. US 7,647,331 B2.*

Nikolai I. Korsunov, D. A. (2015). Recognition Method of Near-Duplicate Images. *The 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.* Warsaw, Poland: IEEE.

Nitish Srivastava, G. H. (2014). Dropout: A Simple Way to Prevent Neural Networks from. *Journal of Machine Learning Research*, 1929-1958.

Olga Russakovsky, J. D.-F. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 211-252. Retrieved from http://www.image-net.org/about-overview

Qassim, H., Verma, A., & Feinzimer, D. (2018). Compressed residual-VGG16 CNN model for big data places image recognition. *IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC).* Las Vegas, NV, USA: IEEE.

Raghav Bansal, G. R. (2018). Blur image detection using Laplacian operator and Open-CV. *5th International Conference* (pp. 399-403). ICEEE.

Rahul Raguram, J. T.-M. (2012). *Improved Geometric Verification for Large.* Chapel Hill, NC, USA.: Department of Computer Science University of North Carolina.

ravel video view statistics - Think with Google. (2018). Retrieved from thinkwithgoogle: https://www.thinkwithgoogle.com/data/travel-video-view-statistics/

Rosebrock, A. (2015, 09 07). *Blur detection with OpenCV*. Retrieved from pyimagesearch: https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/

Rosebrock, A. (2017, 11 27). *Image hashing with OpenCV and Python*. Retrieved from pyimagesearch: https://www.pyimagesearch.com/2017/11/27/image-hashing-opencv-python/

Schmüdde, D. (2019, 03 14). *How to Version Control Jupyter Notebooks*. Retrieved from nextjournal: https://nextjournal.com/schmudde/how-to-version-control-jupyter

Staab, R. A. (2009). Exploiting Flickr Tags and Groups for Finding Landmark Photos. *European Conference on Information Retrieval*, 654-661.

Till Sieberth, R. W. (2016). Automatic detection of blurred images in UAV image sets. *ISPRS Journal of Photogrammetry and Remote Sensing*, 1-16.

Tsung-Yi Lin, M. M. (2014). Microsoft COCO: Common Objects in Context. *arXiv*, 1405.

Weyand, A. A. (2018, 05 1). *Google-Landmarks: A New Dataset and Challenge for Landmark Recognition*. Retrieved from ai.googleblog: https://ai.googleblog.com/2018/03/google-landmarks-new-dataset-and.html

Weyand, B. C. (2019, 05 03). *Announcing Google-Landmarks-v2: An Improved Dataset for Landmark Recognition & Retrieval*. Retrieved from ai.googleblog: https://ai.googleblog.com/2019/05/announcing-google-landmarks-v2-improved.html

Xiaohui Shen, Z. L. (2012). Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking. *IEEE Conference on Computer Vision and Pattern Recognition.* Providence, RI, USA: IEEE.

Yan-Tao Zheng, M. Z.-S. (2009). Tour the world: Building a web-scale landmark recognition engine. *Conference on Computer Vision and Pattern Recognition.* Miami, FL, USA: IEEE.

Zisserman, K. S. (2015). very deep convolutional networks for large-scale image recognition. *Visual Geometry Group, Department of Engineering Science, University of Oxford*.

8. Numbered annexes

Appendix A:

Risk Register for Final Year Project

Prepared by: Team 8 **Date:** 5/15/2019

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Probability	Impact	Risk Score	Status
RIsk 1	1	Trained model is too slow	When passed a video, the trained model takes too long to process it and determine what landmarks are in it. This will defeat the purpose of the applicaiton as users will not want to wait a long time to get the result.	Technical risk	A model aiming for a high accuracy will take longer to process videos	Duration to process a single frame is too long	1) Take lesser frames (increase the time between frames/filter more frames) 2) Create a seperate model with relaxed accuracy requirements and focus more on its performance	7	5	35	risk has not occured yet
RIsk 2	2	User rejects or dislikes the final product	The general user refuses to use the final product for several reasons	Customer risk	1) Application is too hard to use. 2) User does not find a practical use for the product.	Low number of users after launch	Discuss with users to determine what are the issues with the application and attempt to resolve it through a software update.	4	7	28	risk has not occured yet
RIsk 3	3	Added workload or time requireme nts	Due to additional tasks given that arose from a change of direction, finished work not up to standard or other problems.	Estimates risk	Initial proposal was not planned properly and thus the risk and potential problems were not identified.	Work overdue or lagging behind scheduled deadlines	To reduce the risks, make careful considerations when determining the requirements. If triggers occur, review the overall project schedule to adjust to the additional time requirement	2	9	18	risk has not occured yet
RIsk 4	4	Model does not achieve the desired accuracy	The result of testing the model did not provide us with a satisfactory Global Average Precision Score	Technical risk	1) The training algorithm used is not fully optimized. 2) Insufficient number of images used to train the model. 3) Wrong variables used in the Neural Network Training stage.	Model not correctly identifying landmarks consistently	1) Increase the training time. 2) Find a new dataset with more images. 3) Consider other approaches or algorithms to train the model	3	6	18	risk has not occured yet
RIsk 5	5	Time taken to train the model is too long	The model training algorithm takes too long to train on the given dataset and exceeds the time allocated according to project timeline	Estimates risk	A model aiming for a high accuracy will take longer to train	Estimated completion time of the training exceeds allocated time	1) Reduce the number of epochs, which is the number of iterations to train the dataset. 2) Reduce the number of images in the training dataset. 3) Reduce the number of landmarks we aim to train the model. 4) Look for alternative servers that are more powerful to train the model	5	3	15	risk has not occured yet
Risk 6	6	Design is infeasible	The overall design for the project is infeasible to implement for several reasons such as it being too time consuming or too complicated	Technical risk	Main algorithm of the program was not understood when doing the proposal	Inability to implement certain key functions of the product	Reduce the expectation of the product. For example, reduce the goal for the accuracy of the model.	1	9	9	risk has not occured yet
Risk 7	7	Teammate being uncoopera tive	Teammate refuses to do the assigned tasks and communicate with the rest of the team	Organisationa l risk	1) Lost of interest in project. 2) Conflict between team members.	Teammate not adhering to project schedule or not completing task with an acceptable level of competency.	1) Have a discussion with the member on what issues he or she has in the project. 2) Attempt to resolve conflicts among team members. 3) As a last resort, consider dropping or replacing the team member.	1	5	6	risk has not occured yet

Rationale for Probability and Impact Score:

Risk 1 This risk is rated at an above average probability as all the currently published research papaers only work on images and not videos. Therefore, we are unable to estimate how long it would take to process an entire video. The impact is average as if if does occur, it would require us to reconfigure parameters in our model and possibly retrain the model again. This would cause a delay in the overall timeline.

Risk 2 This risk is rated at an average probability as no such product to our knowledge exists in the market that does what we intend to do. Therefore, we have no indication of how well it would be accepted by general users and whether they will find it practical. It has a high impact score as low acceptance of the product will defeat the purpose of spending resources to build the product.

Risk 3 It has a rather low probability as the requirements of the project have already been determined in detail. Changes occurring in the middle of the project should only occur when problems or bugs in the application is found. However, if such changes (especially in terms of requirements) are required, there would be a significant impact towards achieving the goals of the project as we might not be able to deliver a finished application within the expected timeframe.

Risk 4 It has a rather low probability as according to past research papers that we will be referring to, they have achieved a satisfactory acccuracy which we intend to improve on. This has a medium impact as having a low accuracy model will produce too many false positives and false negatives which will impact the user's experience with the app as they are unable to trust the results generated by the model.

Risk 5 It is rated at an average probability as we are unable to estimate how long the model will take to train at this point of time. However, the impact is rather low as we feel like we have allocated sufficient time for the model training process, with consideration that we might have to change parameters before restarting the training process.

RIsk 6 It has a low probability score as many research papers as well as other sources have successfully implemented it and as of now, even at our level of understanding, we feel like it should not be an issue to implement it. However, if this risk does occur, the impact would be very severe as we would have to restart the entire project with a completely new design and waste all the effort that was done prevously.

Risk 7 It has a very low probability as we feel like we get along with each other well as a team and due to the size of the team (3), conflicts should rarely arise. This would have an average impact as dealing with an uncooperative teammate will consume precious time that could have been used on the project. If the worse case scenario happens, which is when we are forced to replace a member, the new member would not have knowledge of what we have been working on previously and therefore would have to learn from scratch

Documentation of Response Strategy:

Risk 1 We could consider increasing the duration between frames that we take to reduce the total number of frames our model has to process. A more stringent filtering algorithm could also be considered to filter out more frames. However, the user and the Youtuber might have different requirements. A Youtuber might value accuracy over time taken while a user might value the opposite. Therefore, we could also create a separate model with relaxed accuracy requirements and focus more on the performance of the processing, which will create a more acceptable user experience.

Risk 2 We will first gather information from the users through various means such as survey and interview to determine what are the causes of them disliking the product. With such information, we can then critically assess our requirements and perform the necessary changes to better suit the requirements of the user. Once we have fixed the issue, a software update can be pushed to resolve the issues.

Risk 3 If such a case occurs, we should evaluate the importance of the additional feature as well as the amount of delay it would cause to overall project schedule. If we have decided to implement it, we would modify the overall project timeline to account for the additional tasks. We can do this by looking at tasks that we have overallocated time for.

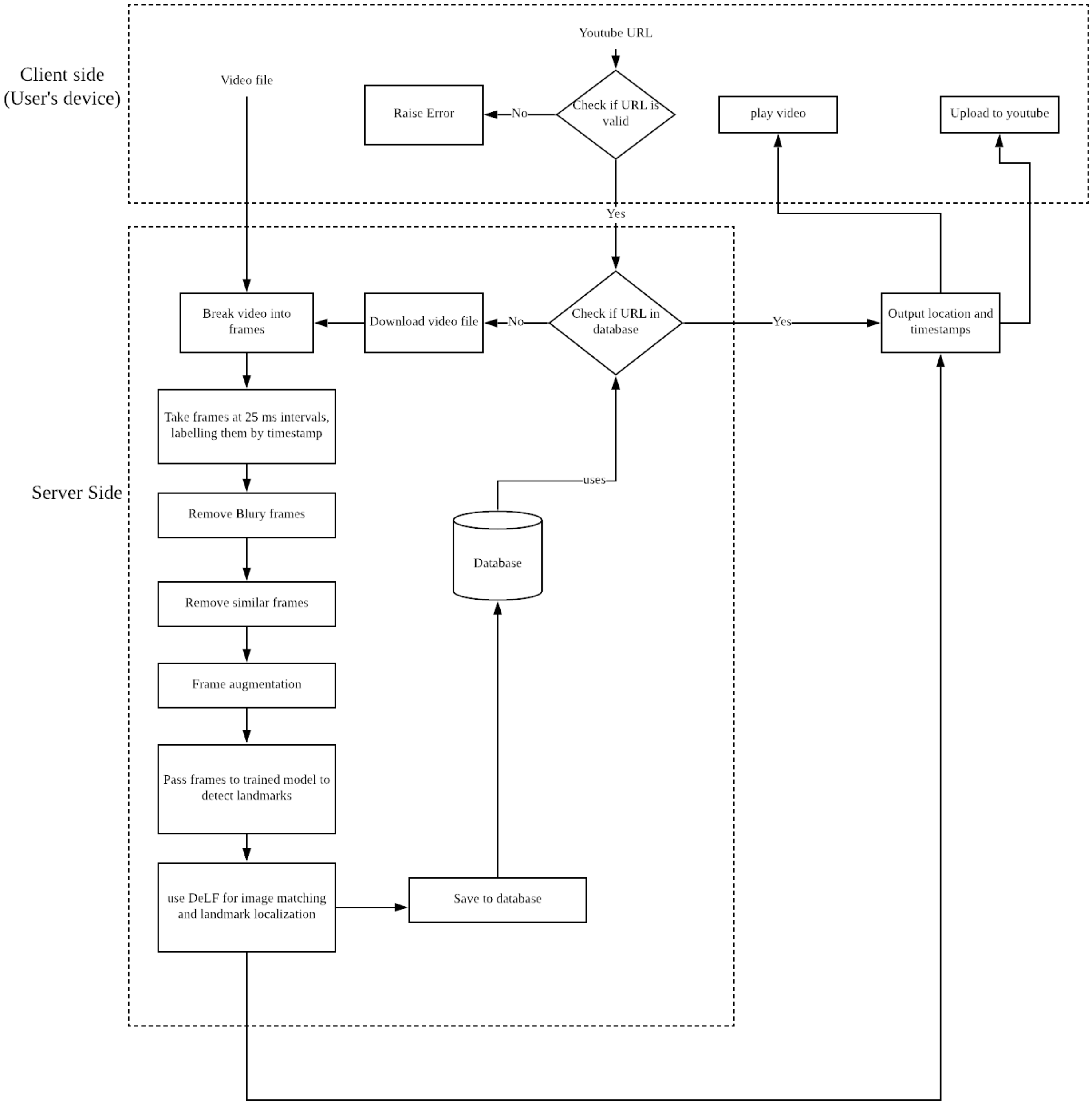
Risk 4 Such a case would occur due to incorrect parameters in the training algorithm. We would first try to change the variables to improve the accuracy of the model. This might negatively reduce its performance. Next, we could also find more data to train the model which will also improve its accuracy. If we are still unable to achieve a desirable level of accuracy, we would consider looking at other approaches to solve this problem.

Risk 5 We would first try to reduce the number of epochs, which is the number of iterations to train the dataset. This will however negatively affect the accuracy. Besides, we could also consider renting more powerful servers that will be able to reduce the training time to fit within our schedule. Finally, we can also consider reducing the number of images or landmarks that we are training our model with.

RIsk 6 If this occurs, it means we have overestimated our capabilities. We would first try to reduce our expectation for the product by maybe reducing the accuracy that we aim to achieve. By doing this, we can be more open to the solutions we can use to solve the problem.

Risk 7 First, we would have a team discussion to determine what are the issues that team members have with the project or fellow team member. If it is a conflict between members, try to have a discussion to arrive with an agreement on how to move forward in the project. Only as a last resort, we should consider the option of dropping or replacing the team member.

Appendix B:



Client side
(User's device)
Youtube URL
Video file
Raise Error
No
Check if URL is valid
play video
Upload to youtube
Yes
Break video into frames
Download video file
No
Check if URL in database
Yes
Output location and timestamps
Take frames at 25 ms intervals, labelling them by timestamp
uses
Server Side
Remove Blury frames
Database
Remove similar frames
Frame augmentation
Pass frames to trained model to detect landmarks
use DeLF for image matching and landmark localization
Save to database