CS407 Individual Report

Matthew Wight (2008398)

Individual Contribution

My main technical contributions were the research, development and integration of several of the individual features included in the AutoOSINT app, namely the car number plate analyser, the road and street sign reader, the JPEG ghost detector and the confidence ranking system featuring the PIGEON transformer geolocation model. Additionally, I implemented the interactive map powered by OpenStreetMap, which displays pointers to the GPS locations predicted by the automated features.

It was also decided early on in development that I would focus heavily on maintaining a a high quality of documentation within the project, including the the Readme file in the main project repository and contributions to the written reports submitted by the group, giving context to the individual chapters and sections to ensure flow and readability.

In addition to this, I provided support, guidance and encouragement throughout the project by maintaining consistent communication both online and in-person through meetings. I made sure to give feedback on code and design ideas as well as reminding other team members of deadlines helping with difficult technical tasks.

Team Management

From the beginning of the project, our group opted to use a flat hierarchy, with every group member's opinion holding equal weight during planning phases. This helped to minimise any pressure felt by group members when making crucial decisions. Despite this, each group member maintained a substantial amount of control over the direction of project due to the small size of the group.

Each team member offered a unique skillset, which made delegation of roles and tasks relatively simple as members were able to choose a task to implement and another to assist with at the start of each sprint. Allowing each member to choose their tasks helped in maintaining a high morale and good level of motivation throughout development as it made the process overall more enjoyable and interesting.

However, due to the small team size, there were some areas in which none of the team had a great deal of expertise. In particular, nobody in our team was confident and experienced with web development tools. We aimed to tailor the project's technical requirements in order to mitigate such weaknesses—for example, we chose to create a desktop app to avoid wasting valuable development and research time troubleshooting web-related issues. We were also able to focus development primarily around languages and frameworks which the team were already familiar with, namely Python and Django. The extensive selection of libraries offered by Python were also massively helpful to minimise the project's complexity and development time wherever possible.

Team Effectiveness

The team's effectiveness as a unit was overall good and communication was regular, both over online channels and through in-person meetings. All group members were highly motivated to complete their assigned tasks due to the flat hierarchy and ability of each member to choose their tasks and responsibilities before each sprint.

However, we faced some issues with communication during especially busy periods in the academic year (around coursework deadlines). During these times, communication was less frequent and development progress was much slower. The team mitigated this issue by discussing its impact on the project during the initial planning phase and factoring it into the development timeline. By ensuring that the timeline was realistic, we were able to complete almost all tasks on or before their planned completion date.

Consideration of Team Roles

As discussed in the previous section, the hierarchy of the team in terms of high-level decision making was largely flat. However, Some specific roles were assigned to each group member in order to set out clear expectations and responsibilities, and to ensure smooth and efficient workflows. Planned team roles were discussed before their assigning so that team members could choose roles that would best fit their skillset, and to ensure that no potential roles were overlooked. Selecting roles based on skills and preferences also helped to motivate the team to maintain their responsibilities.

I was assigned the role of Admin Contact, responsible for communication with the module organiser, because of my previous experience working with them. In addition, I took on the role of Document Manager. This was a larger role involving maintaining and proof-reading reports, presentation slides, Readme files and code comments. Additionally, I provided a report template and section headers in Overleaf to help provide guidelines for the structure and style of the group report.

Justification of Working Methods

Development Methodology

We chose to adopt the 'Scrumban' development methodology for this project, combining a classic Scrum methodology with a Kanban board to help keep track of the task backlog and in-progress tasks[(1)(2)]. The incremental approach was a good fit given our requirements, as the aim of the AutoOSINT app is to combine a range of OSINT tools into a single semi-automated program—thus, each increment can focus on a small subset of the tools.

One of the main advantages of using Scrum was the increased transparency within the team, allowing all team members to see all of the changes made to the project and helping to increase trust. It also promoted individual responsibility for each team member over their chosen tasks, making everyone more motivated.

Another crucial aspect of Scrum methodology is its adaptability. Because work is organised into deliverable-oriented sprints, plans can be adjusted at many different points in the development process. This was especially beneficial for our project for two reasons: firstly, because the team was inexperienced in using some of the OSINT tools planned to be automated, meaning the design and implementation was likely to change from the original plan after research had been conducted.

Additionally, some of the tools featured in the AutoOSINT app (such as the PIGEON visual transformer geolocalisation model) are cutting-edge and were only published or updated part way through the development process. Including these features greatly improved the overall utility of the product but required changes to the original plan which were only possible due to the methodology chosen.

Increments and Sprints

Development time was split into two increments, each consisting of three sprints. By planning the tasks to be completed during each sprint, we were able to maintain focus on producing the agreed-upon deliverables. This reduced the risk of scope creep, which was a risk that had been considered due to the expansive number of possible features and OSINT tools that could be included in the project.

Lessons Learnt

Team Development

From the beginning of the project, the team achieved a good level of harmony because the other three group members were all active members of the same society. This helped greatly in the initial stages of organising the team and making team members acquainted with each other. However, it presented a unique challenge for me, being a newcomer to the friendship group.

I made an effort to maintain a high level of activity in the chosen communication channels, as well as attending social meetups and events with the rest of the team, in order to develop a friendly relationship conducive to strong communication, teamwork and efficiency. I learned to consider the key intervention measures at each stage of group development in order to have the greatest influence on the project and to make the team more effective as a whole[(3)(4)].

Project Scope

Ultimately, some areas of the project scope were likely too broad: while good results were achieved on a large number of inputs, it might have been more interesting to focus on a much smaller subset of images (such as just mountain scenes) in order to attain more precise results. However, we were still able to complete all of the main objectives given in the original specification despite losing a quarter of the development team.

In the future, if I were to be part of a smaller development team, I would aim to produce a smaller and more specialised app rather than attempting to rival state-of-the-art quality on a large scale.

Communication and Methodology

One weakness of our chosen working methodology was that each member being able to choose their own tasks led to a slightly uneven distribution of workload. Some of the tasks in the backlog were easier to complete than others, so team members might in some cases be incentivised to pick the easiest tasks.

Another issue was with communication: we were unable to contact one of our group members outside of term time, which led to development being slowed down significantly. This was a difficult issue to address with a completely flat hierarchy and no in-person meetings outside of term time. For some groups (such as those with salaried team members), an incentive/disincentive scheme might help to deal with members being inconsistent at some times in development.

The loss of one of our group members at the beginning of Term 2 had a significant impact on the project which was difficult to mitigate or plan for. This was especially impactful due to the already small size of the team. Fortunately, our agile development methodology proved to be robust enough to allow us to adjust the development timeline and team roles to make up for this.

Although this change presented substantial challenges to our group, I found that it helped me to learn more about leadership and project management due to the increased pressure on the project. I aimed to be engaged, supportive and encouraging within the group in order to keep motivation as high as possible despite the setback.

Skills and Knowledge Learnt

Software Engineering Principles

As previously discussed, the team as a whole lacked experience with web development. While the implementation and overall architecture of the AutoOSINT app was designed with the aim of minimising the amount of web development knowledge required, there were still aspects of the project (such as the integration of the Python scripts into the Django REST-based app using HTML and JavaScript elements) that helped the team to become more comfortable with front-end and full-stack development. Now, many of these skills can be directly applied to web app development problems in the future.

Furthermore, our software engineering skills improved as a whole thanks to the extensive hands-on experience designing, implementing and integrating various features. This project has been the largest and most complex of any academic project that any of our group members have undertaken at undergraduate level. The scale of the project helped to improve our practical understanding of project management and software engineering principles—for example, the benefits of using a Kanban board became clear, as keeping track of task completion and backlog would have been difficult without it.

Computer Vision Techniques

Developing features including the road sign analyser and number plate extractor gave me extensive experience with foundational computer vision tools. I utilised a wide variety of vision techniques including morphological operations, image sharpening and blurring and image metrology (measuring dimensions in images). In addition, I conducted a large amount of research into machine learning models for computer vision such as vision transformers and CLIP. It was especially interesting to see more applications of findings from papers I had read for previous projects. For example, Attention is All You Need[(5)], which focused on machine translation and natural language processing, is a core part of vision transformers, which are a type of image classifiers shown to outperform convolutional networks in many cases[(6)(7)].

Transformers and ViTs

One of the most exciting areas I conducted research into was transformers and vision transformers (ViTs). I briefly encountered transformers during my third year individual project as a tool for natural language processing. Transformer neural networks use an attention component which is able to embed context into inputs, allowing the model to distinguish between more important versus superfluous input features. Transformers have been shown to substantially outperform recurrent and convolutional networks in many machine learning tasks

Furthermore, newly published research has shown that transformers can also be used to perform classification on images—these models are called vision transformers (ViTs). A recent model called CLIP combines a ViT with a text embedding method such as Word2Vec, which can embed semantic and syntactic information into text vectors, and is able to perform zero-shot classification for a huge variety of tasks without re-training.

I implemented a CLIP model into the confidence scoring module of our app and found it to be very effective and accurate in many different cases. During the project's development, a new CLIP model was published which is designed specifically to perform image geolocation. The model's predictive accuracy is very impressive (better than any human expert) when given images in a style similar to Google Street View, although I found many cases where its performance was poor (especially rural scenes). I would be very interested to use CLIP more extensively for image geolocation in a future project.

Depth and Pose Estimation

While it was ultimately not included in the final product, I spent a substantial amount of development time researching, experimenting with and testing various depth and pose estimation techniques. This research and testing has given me new knowledge about several areas of computer vision, especially techniques applicable to mobile robot navigation[(8)] (which regularly utilises depth estimation).

For depth estimation, I considered three approaches. The first involves using the Thin Lens Equation[(9)]:

$$\frac{\text{real object size}}{\text{distance to object}} = \frac{\text{object height on sensor}}{\text{focal length}}$$

The idea behind this approach is to use a reference object with a known height (or to infer height of a scene object using a reference image) in order to estimate the distance between the camera and a landmark in the scene. This would allow us to estimate the exact geographic location of the point of view, given the location of the object and the camera's bearing are already known[(10)]. We opted not to use this method because it required prior knowledge of the exact specifications of the camera used to take the photo, as well as the exact height and location of the reference object in the image. Additionally, testing showed this method to be ineffective with large reference objects such as buildings due to view obstruction and perspective warping.

Another approach was homography-based plane detection, aiming to compare the input image with a reference image taken from a similar position and use the transformation between the two to estimate location[(11)]. This approach had similar problems to the first, and the two image's perspectives were often too ambiguous to deduce any meaningful positional data from. Calculation of vanishing points can also be used to help find a solution, but this relies on flat ground and many straight lines being present in the scene[(12)].

The final method tested was a deep learning approach. One deep learning model I tested was the Depth Anything model, which combines a visual transformer with a convolutional neural network and has been trained on 62 million images including indoor and outdoor scenes[(13)]. While such models were very effective at predicting relative depth in an image, the absolute depth predictions were poor—especially for outdoor scenes with objects placed a long distance from the camera. Although there is ongoing research into this area, I was unable to find a suitable pre-trained model for our purposes (or a large enough dataset to train a custom model).

Changes to Working Methods

Throughout the lifespan of the project, there were a number of alterations made to both the timeline and the working practices used. The main change was the frequency of meetings—initially we had planned to meet as a team every week and also meet with our project supervisor every week. Soon into the start of development, we found that this was not an effective use of time, as some weeks would see little to no progress being made due to team members focusing on coursework for other modules.

Regular group meetings were instead scheduled every two weeks, instead relying on online communication channels and the shared GitHub repository and Kanban board to ensure transparency and regular communication. Where pair programming was implemented, pairs would also conduct more frequent in-person meetings to discuss technical details of a specific feature. We found that these changes improved productivity and made meetings more focused and purposeful.

Towards the end of the project, we also implemented regular check-ins over text, stating clearly each day what progress had been achieved yesterday and the plans for that day. This was an alternative to daily in-person standup meetings. We found this to be an effective way to keep all team members informed of the project's progress as the deadline neared and the amount of time dedicated to finishing the app and the report increased. It also helped to encourage team members to split the workload evenly, as it became more clear when some members were working slower than others.

Throughout the project, there were several changes made to the requirements and development timeline. For example, we decided to reschedule the confidence ranking system to a later sprint in iteration 2 because it directly uses the predictions made by the other features, so its performance would be difficult to test if it was implemented before all of the other features. To ensure progress on the project was consistent, we moved forward the deadline for completion of the road sign analysis module. Our agile development methodology made this change very simple.

Another change made was the decision not to include generative AI detection in the app. This decision was made after discussing with the project supervisor, who advised that AI detection would likely be a very complicated feature to research and implement, and that development time would be better spent on implementing geolocation features instead.

Excluded Planned Features

Two of the features we had planned to include in the app—clue input and shop sign recognition—are not available to use in the final app. This is because both features were assigned to the group member who left the group. Had this occurred earlier in the development timeline, we may have decided to delegate these tasks among the remaining group members or discuss their exclusion more extensively in the final report. However, we only learned that these features had not been completed as expected very close to the project deadline, so instead we chose to remove them from the project, briefly discussing these changes in the project timeline and management sections of the group report.

Fortunately, due to the modular approach we took to designing the system architecture, the effects on the app's functionalities were minimal, and we were still able to achieve promising results in testing. I found that this modular approach was very helpful in making development as efficient as possible, especially with the variety of different features and changing requirements. I would strongly consider using the same approach in future projects.

In hindsight, the impact of this challenge on the project shows that a more robust review system might have been beneficial by ensuring that at least two members were fully aware of the progress towards completion of each task. However, this would have been difficult to implement due to the small team size and difficulties with contacting the group member who left during the last few months of the project.

References

[1] E. Hassanein and S. Hassanien, "Cost efficient scrum process methodology to improve agile software development," *International Journal of Computer Science and Information Security*, vol. 18, 2020.

[2] Z. Khan, "Adaptive agile development process : Using scrumban to improve software development process," 2014.

[3] D. Seet, "Critical team development and intervention tips for project managers," 12 2008.

[4] I. Saunders, "Leadership and teamwork," 2022.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 06 2021.

[7] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 1–1, 2022.

[8] E. Vincent and R. Laganiere, "Detecting planar homographies in an image pair," *CiteSeer X (The Pennsylvania State University)*, 11 2002.

[9] A. H. Firester, "The thin lens equation for optical parametric image conversion," *Optical and quantum electronics*, vol. 1, pp. 138–142, 08 1969.

[10] L. d. Guzman, "Scene distance estimation using monocular depth estimation," 01 2024.

[11] D. Conrad and G. N. DeSouza, "Homography-based ground plane detection for mobile robot navigation," *MOspace Institutional Repository (University of Missouri)*, 05 2010.

[12] A. Criminisi, "Single view metrology," *International Journal of Computer Vision*, vol. 40, pp. 123–148, 2000.

[13] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," 01 2024.