User-Centric Visualisation of the Top Activations in a Deep Neural Network for Explainability

Michael T. R. Johns

853369

Project Dissertation submitted to Swansea University in Partial Fulfilment
for the Degree of Master of Science.

Enhancing Human Collaborations and Interactions with Data and Intelligence-driven Systems

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

September 30, 2021

Declaration

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed Michael T. R. Johns (candidate)

Date 30/09/2021

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed Michael T. R. Johns (candidate)

Date 30/09/2021

Statement 2

I hereby give my consent for my thesis, if accepted, to be made available for photocopying and inter-library loan, and for the title and summary to be made available to outside organisations.

Signed Michael T. R. Johns (candidate)

Date 30/09/2021

I would like to thank my project supervisors for their support:

Prof. Matt Jones

Prof. Mark Jones

Dr Leighton Evans

I would also like to thank Ordnance Survey for their support throughout this thesis:

Isabel Sargent

Jeremy Morley

James Clarke

Contents

List of Figures

Table of Tables

1. Introduction

This project will design and implement a program to visualise top activation images from a neural network to enable human in the loop intuitive explainability of a neural network's decisions.

1.1 Project Description

Using a pre-trained model to investigate and implement a tool that can aid in giving meaning to how a black box machine learning neural network model has made its decisions during the training of landscape mapping data. It was Implementing a Graphical User Interface (GUI) system that can visualise steps taken by the neural network and how it has come to its decisions. Displaying the visualisation of these steps at several points using the test data on the existing trained model and presenting it to a user to show the meaning of decisions taken by the machine learning model. The investigation of high dimension datasets and experiment with how best to display in a GUI to enable user-friendly insight and meaning to how a neural network model makes its decisions. Increasing meaning to how neural networks come to decisions at the node level could allow further optimisations and insights into landscape datasets.

1.1.1 Aims

- An interactive visualisation tool to display which key images were used to activate nodes.
- To allow humans to see which key images have affected the machine learning models decisions.
- To gain insight and explanations of decisions made during the machine learning process.
- Implementation of a user-centric GUI tool to allow more explainability into model node decisions.
- Utilise an ethical approach to UX and system design throughout the project with human-centric design principles.

1.1.1.4 GUI Tool Aims

- Visualisation of key images at several nodes in the machine learning process.
- Experiment with how best to visualise the meaning behind decisions.
- User-centric design to enable clear insight into model node decisions.
- Task design and descriptions input gained from OS experts.
    - User labels what they see.
    - User labels what they think the machine learning algorithm might see.
- Working prototypes at the end of each sprint for user evaluation

1.2 Motivation

Beyond the model definitions and the quantitative analyses, there is a need for qualitative comparisons of the solutions learned by various deep architectures[1]. The explanation of why a neural network makes its decisions could help streamline model layers to improve the speed and understanding of decisions made.

1.2.1 Ordnance Survey

Ordnance Survey is the national mapping agency for the United Kingdom. Ordnance Survey provides both the public and business areas with accurate location and map data. Ordnance Survey are key stakeholders in this project with interest in human-centred design to enable explainable machine learning models. The stakeholders involved in this project from Ordnance Survey are Isabel Sargent, Jeremy Morley and James Clarke.

1.3 Software Tools

The software tools selected for use during this project were decided by stakeholder experts and the author to be of most use to meet the requirements.

1.3.1 wxWidgets

wxWidgets is a library tool that allows cross-platform development from one codebase. wxWidgets uses a window management system that is object-oriented and able to be altered using derived C++ classes. wxWidgets is under an open-source license which enables its use for no cost commercially.

1.3.2 C++

The C++ programming language was chosen to utilise the speed of loading images and to be compatible with wxWidgets. The author of the project is most proficient in C++, and this allowed development to continue smoothly.

2. Literature Review

This section will review the background literature relating to the project. There will be several topics covered. Visualisation techniques relating to deep neural networks and the use of user-centric design to enable explainable machine learning models and a brief overview of explainable machine learning methods, trust in AI and ethical issues surrounded AI.

2.1 Explainable Machine Learning

Machine learning algorithms, such as artificial neural networks (ANNs), have been referred to as ‘black box’ algorithms[2], [3]. Although machine learning algorithms such as ANN perform well on some tasks, such as classification, there is always the question of how these ‘black box’ algorithms reach an outcome and the need for trust-building in these algorithms[4]. Human comprehensibility of algorithms of a ‘black box’ nature may help in understanding how to improve the output of the algorithm [1], [2], [3].

In 1993, Towell and Shavlik[2] used backwards propagation in an attempt to translate this decision-making process into a human-comprehensible fashion[2]. This is done at a global level within the algorithm and uses knowledge of the information the network holds in the form of feature names and values[2].

In 2006, Hinton et al. [5] introduced a method to learn features and hierarchies within a deep neural network, one layer at a time. Higher layers were given tied weightings to allow the model to learn the lower levels. As each layer was learned, the next layer to be learned was given untied weights to show that adapting the higher layer weights would result in the overall generative model[5]. The results outperformed all algorithms at the current time using the MNIST dataset[6]. The downfall to this method is only a single whole layer was given as output during Hinton et al. [5] results.

In 2009, Kavukcuoglu et al. [7] introduced a method to visualise features using an unsupervised method that can learn feature choices from the combination of multiple layers and their pooling layers. A topographic map (Figure 1) is produced from feature combinations using the natural image dataset[8].

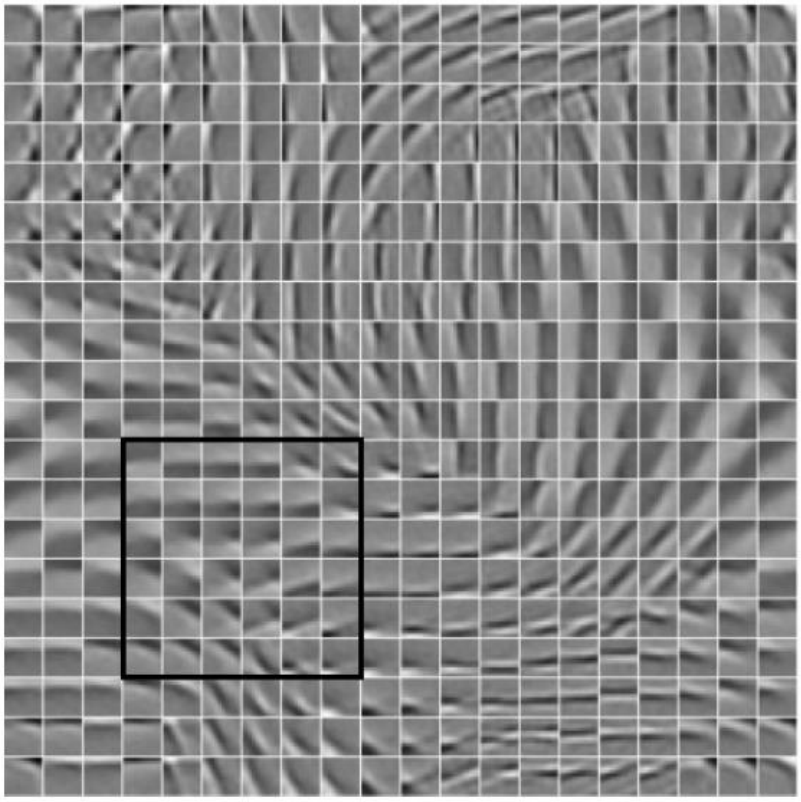


Figure 1: Topographic feature map by Kavukcuoglu et al. [7]*. The map was learned from combining multiple layers and their pooling layers to produce a topographic map of features using the natural image patch*[8] *dataset*

Erhan et al.[1] suggested more work was needed around the area of visualising higher-layer features of a deep network, expanding on Kavukcuoglu et al.[7] workaround feature visualisation and explainability. Vision datasets were the focus for Erhan et al.[1], stating that qualitative comparisons of outputs from a deep network are a must for human comprehensibility. Several techniques were compared using vision datasets and a deep belief network. A filter-based method called activation maximisation was introduced to attempt to explain node activations and what might be learned at each node[1]. Feature sampling from a node is also presented by Erhan et al.[1] in the attempt to show what each node was learning in a human-readable visualisation. Each unit of any layer of the network was taken and compared with several visualisation methods using the MNIST dataset[6] and a natural image patch dataset[8] (Figure 2). The visualisations output showed which features were being learned across two datasets and how different features were learned using the different methods[1]. Erhan et al.[1] confirmed their intuitions that lower-level features can correspond to combinations in the resulting higher-level layers of a deep neural network (Figure 2).

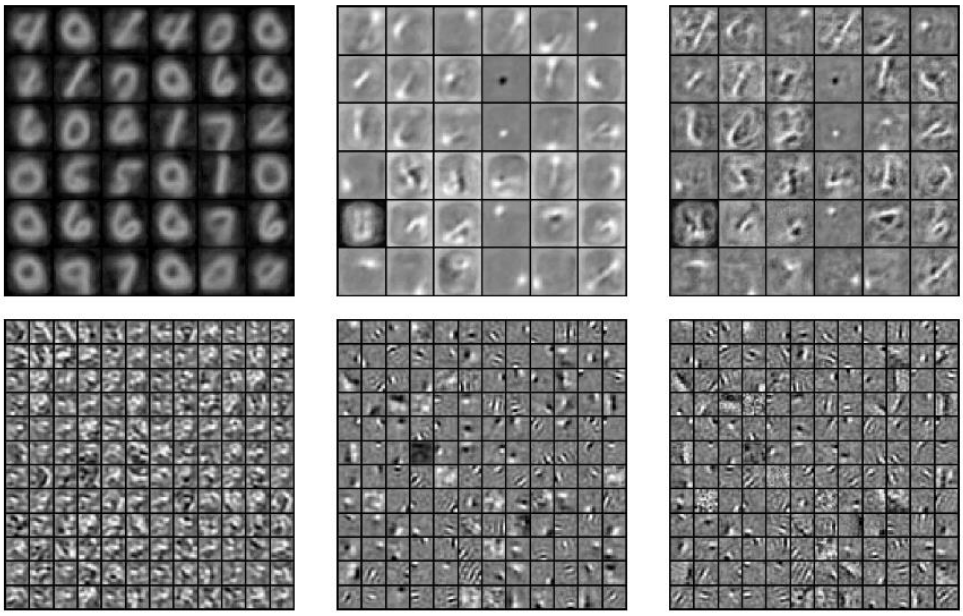


Figure 2: The visualisation by Erhan et al. [1] *of a deep belief networks layer two and trained on two different datasets, Top: 36 units visualisation the MNIST*[6] *dataset. Bottom: 144 units visualisation of the natural image patch*[8] *dataset. Left: Sampling with added clamping. Middle: Linear combination of the previous layer filters, showing meaningful activations on higher layers, have been learned from previous layers in the network. Right: Activation maximisation per unit, white is positive activation, black is negative, and grey is zero activation.*

In 2011, Zeiler et al. [9] presented an unsupervised hierarchical model that shows distinct features of an image across a four-layer neural network. The dataset was comprised of 3060 images which were converted to grey-scale and resized to 150 x 150. The single largest absolute activation is taken and transformed into input space[9]. Layer 1 displayed differing gabors[10], layer 2 expands using layer one features to create edges and features[9]. Layer 3 builds from layer two and works by clustering several layer two features together. Finally, in layer four, near-complete structure reconstruction is visible for objects in the final images (See Figure 3).

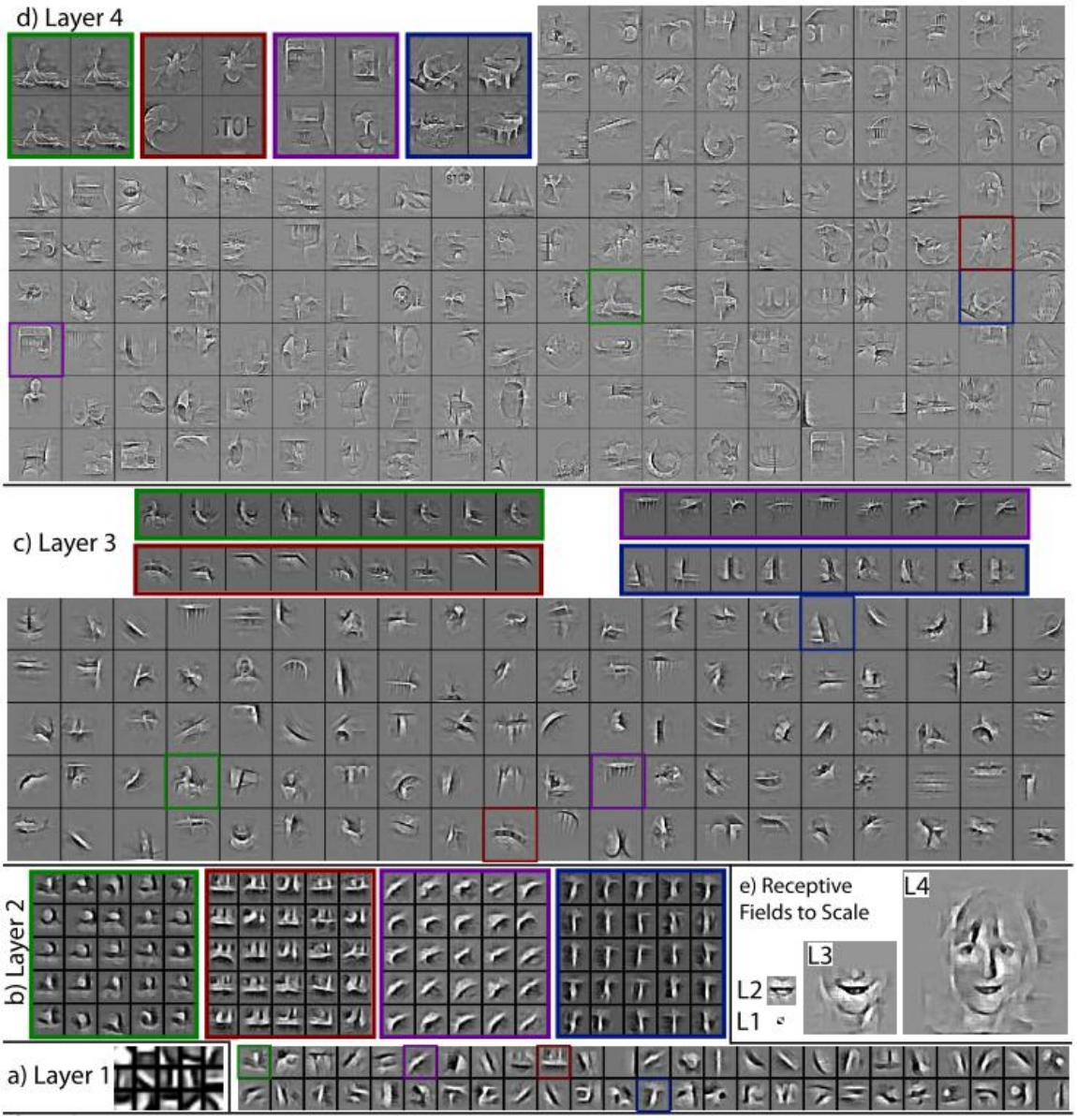


Figure 3: Zeiler et al. [9] *visualisation method of a four-layer unsupervised neural network and feature reconstruction across each layer. a) Layer 1 shows the gabors*[10] *of differing frequency, b) Layer 2 shows cluster combinations of features from layer 1 to create edges and features, c) Layer 3 further clusters the features from layer 2 to create object features, d) Layer 4 further combines features in layer 3 to show near-completed objects from the original input images.*

Zeiler et al. [9] made use of the visualisation output by adjusting custom pooling switches to change the variable input patterns. The custom pooling switches allows the method to work on datasets the model was not initially trained for with a comparable performance rating[9]. This pooling method allows each layer to be trained using the original input image, rather than only relying on the inputs of the previous layer results.

In 2013, Vondrick et al. [11] explained a method to visualise features of a HOG[12] object detector to determine where the algorithm fails to detect an object correctly. The focus of the visualisation was to enable human-readable HOG feature visualisation that is intuitive to humans. Inverting the HOG[12] feature map was used to display an object in a more human intuitive output. Vondrick et al. [11] performed a user study that required a training course and qualification to be passed by all participants to ensure expertise in the area required. Each participant was shown three images and was asked to classify the image into one of twenty categories. There was also an option to select no confidence in their answer. The outcome of the study showed Vondrick et al. [11] method performs better than glyph representation, except for glyphs performing better for images with bicycles present.

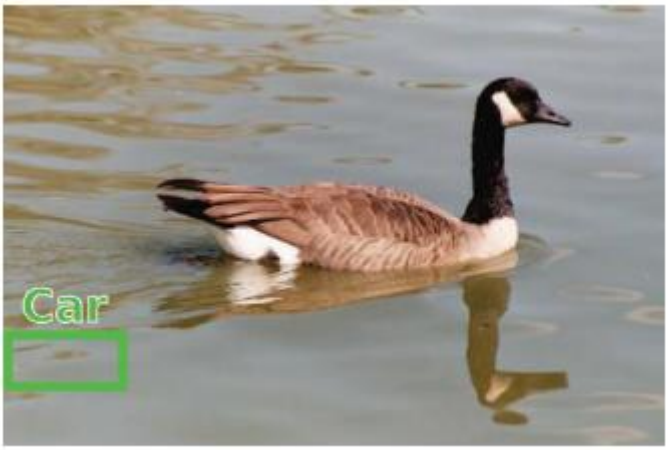


Figure 4 [11]*: A PASCAL image showing a high score detection of a car within the image.*

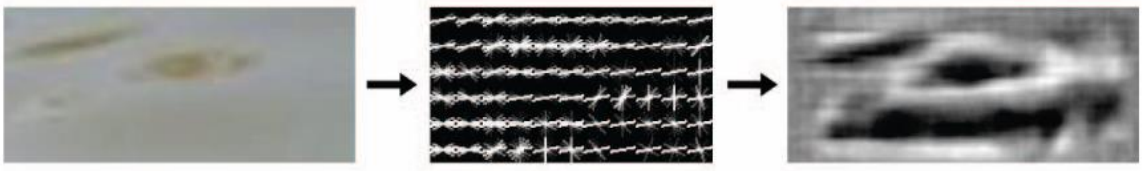


Figure 5: Vondrick et al. [11] *visualisation of features shows that the HOG*[12] *features do present as a car is present in the image, in a more human intuitive fashion than HOG features alone. Left: Image car feature from figure 4, Middle: HOG features of left car feature, Right Vondrick et al.* [11] *visualisation showing a car outline is presented from the HOG features in an intuitive human fashion.*

2.2 Visualising Lower Layers in Neural Networks

Following on from Hinton et al. [5] introduction of a method to enable the learning of high-density neural networks that have many layers, further research into the area emerged.

The visualisation of neural networks beyond the higher layer features was investigated by Zeiler and Fergus[13] in 2014. The visualisation method was built on Zeiler et al. [9] previous work that introduced an adaptive deconvolutional network (deconvnet[9]) for mid and high-level feature learning[9]. The deconvnet is used to attach to a convolutional neural network (CNN) to allow an approximated visualisation of the image pixels at a unit level in any layer within the network[13].

The visualisation is performed by setting only an individual CNN activation weight and setting all other activations in the layer to zero. An un-pooling step is then performed using the locations of the maxima within each pooling region to maintain a structure with the previous layer. The model then uses relu non-linearities to ensure feature maps are positive and a filtering step, similar to backward propagation, to generate an accurate visualisation of the feature map. Zeiler and Fergus[13] explain the limitation of only a single activation in each layer being shown using their visualisation and deconvnet algorithm.

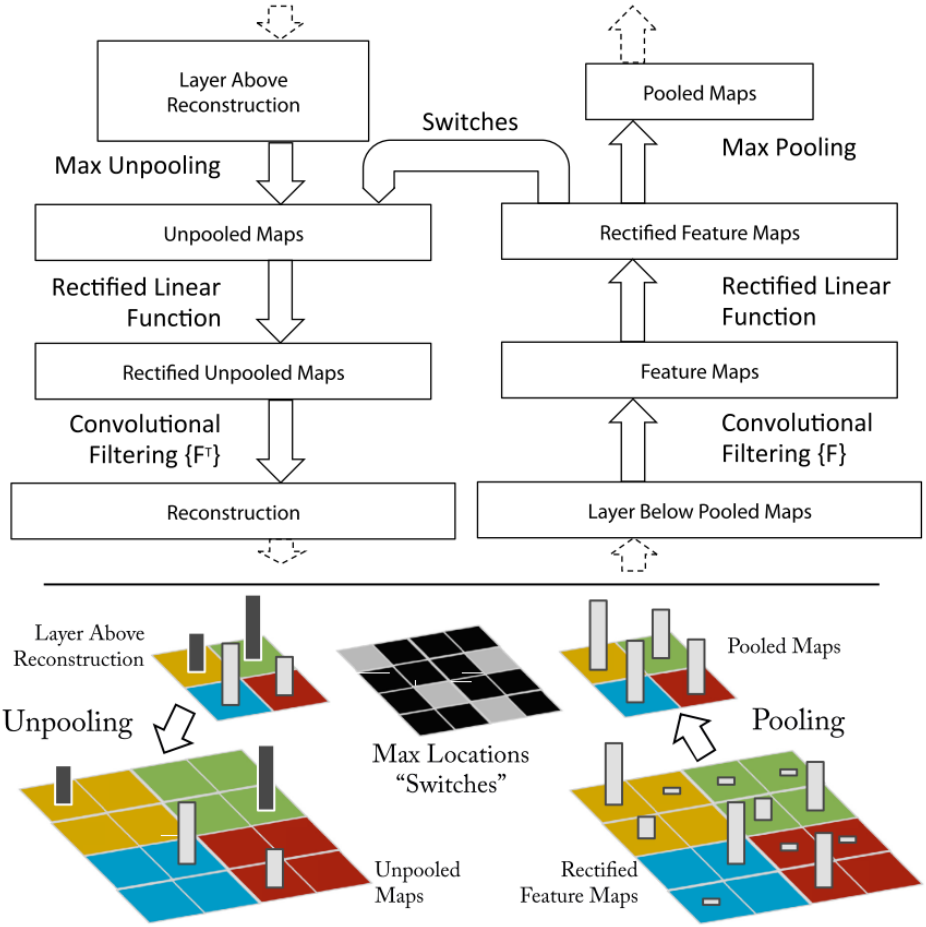


Figure 6: Zeiler et al.[9] *deconvnet layer attached to a CNN layer that shows the process of reconstructing the approximation of the features from the layer below. Switches are used throughout un-pooling and pooling layers to show negative(black) and positive(white) activations.*

The visualisation of activations across several layers allowed Zeiler and Fergus to select features and improve on Krizhevsky et al. [14] 2012 ImageNet classification result, showing local relationship sensitivity in the classification model[13]. Figure 7 shows the comparison and effects of removing or adjusting layers after insight gained from the visualisation output.

Error %	Train Top-1	Val Top-1	Val Top-5
Our replication of Krizhevsky *et al.* [18], 1 convnet	35.1	40.5	18.1
Removed layers 3,4	41.8	45.4	22.1
Removed layer 7	27.4	40.0	18.4
Removed layers 6,7	27.4	44.8	22.4
Removed layer 3,4,6,7	71.1	71.3	50.1
Adjust layers 6,7: 2048 units	40.3	41.7	18.8
Adjust layers 6,7: 8192 units	26.8	40.0	18.1
Our Model (as per Fig. 3)	33.1	38.4	16.5
Adjust layers 6,7: 2048 units	38.2	40.2	17.6
Adjust layers 6,7: 8192 units	22.0	38.8	17.0
Adjust layers 3,4,5: 512,1024,512 maps	18.8	**37.5**	**16.0**
Adjust layers 6,7: 8192 units and Layers 3,4,5: 512,1024,512 maps	**10.0**	38.3	16.9

Figure 7: ImageNet 2012 classification error rate comparison between Krizhevsky et al.[14] *and Zeiler and Fergus*[13]*(depicted as "Our" in the table). The table shows the adjustments and outputs made by Krizhevsky at the top half of the table and Zeiler and Fergus at the bottom half of the table. The results show a significant decrease in error rates using Zeiler and Fergus' method.*

In 2017, Zintgraf et al.[15] expanded on Erhan et al.[1] and Simonyan et al.[16] work, introducing a visualisation method to analyse how CNN's make decisions during image classification. Image patches were created from the pixels of interest and their neighbouring pixels. Zintgraf et al.[15] then use the image patches to display the paths through multiple layers within the CNN, Similarly to Zeiler and Fergus's work[17].

Zintgraf et al.[15] method achieved showing hidden layers in a deep CNN and the features each layer could be learning from a higher level (See Figure 8) to deeper level layers (See Figure 9). The higher-level layers show that multiple features are being learned simultaneously, and the lower-level layers are showing more specialisation of features such as eyes in the image.

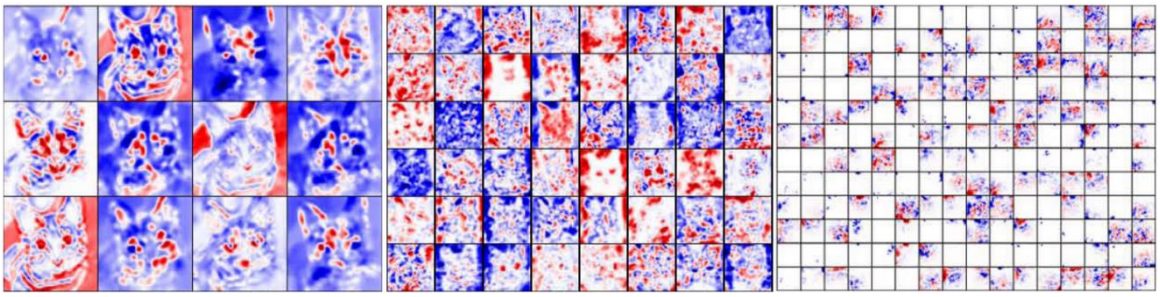


Figure 8: Zintgraf et al.[15] *visualisation of the higher-level feature maps of a deep CNN. The red pixels in each image show the high contribution areas for the decision of classification in each node of each of the three layers. The blue pixels indicate lower contribution areas than the red pixels.*

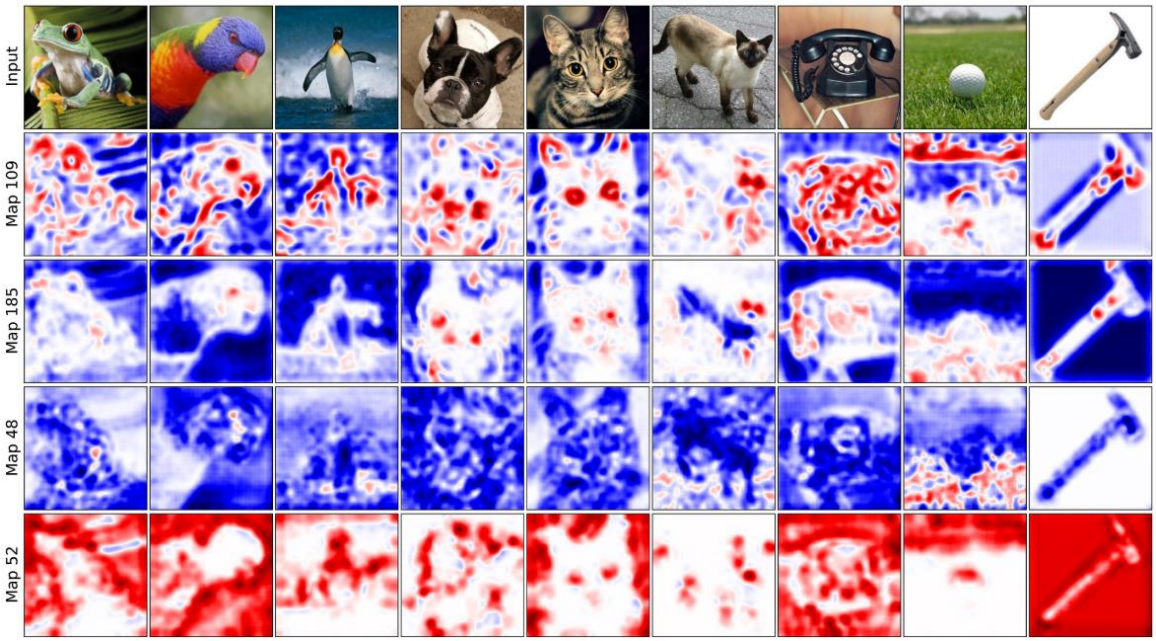


Figure 9: Zintgraf et al.[15] *shows a lower-level feature map than the previous figure 8, showing red pixels as high contribution and blue pixels as lower contributions. Map 185 shows this layer is focussing on the eyes in the image, and map 52 shows the background of an image is the focus of the layer.*

In 2018, Narayanan et al.[4] investigated the interpretability of machine learning algorithms to humans. The interpretability of networks was suggested as a method of how much humans can interpret at once, e.g., a 5-node decision tree versus a 5000-node decision tree. A study performed with 100 participants resulted in showing that an increase in complexity (number of lines, new concepts, repeated variables) increased the response time and lowered the user satisfaction in the result[4]. Tasks that were asked for accuracy in the shortest amount of time possible were performed with little effect on

overall accuracy from the users, suggesting more complex tasks trigger more concentration than simple tasks for humans[4]. Narayanan et al.[4] suggest that knowledge of the largest factors that affect a humans interpretability of machine learning explanation can help direct which features need to be focussed on when explaining machine learning algorithms.

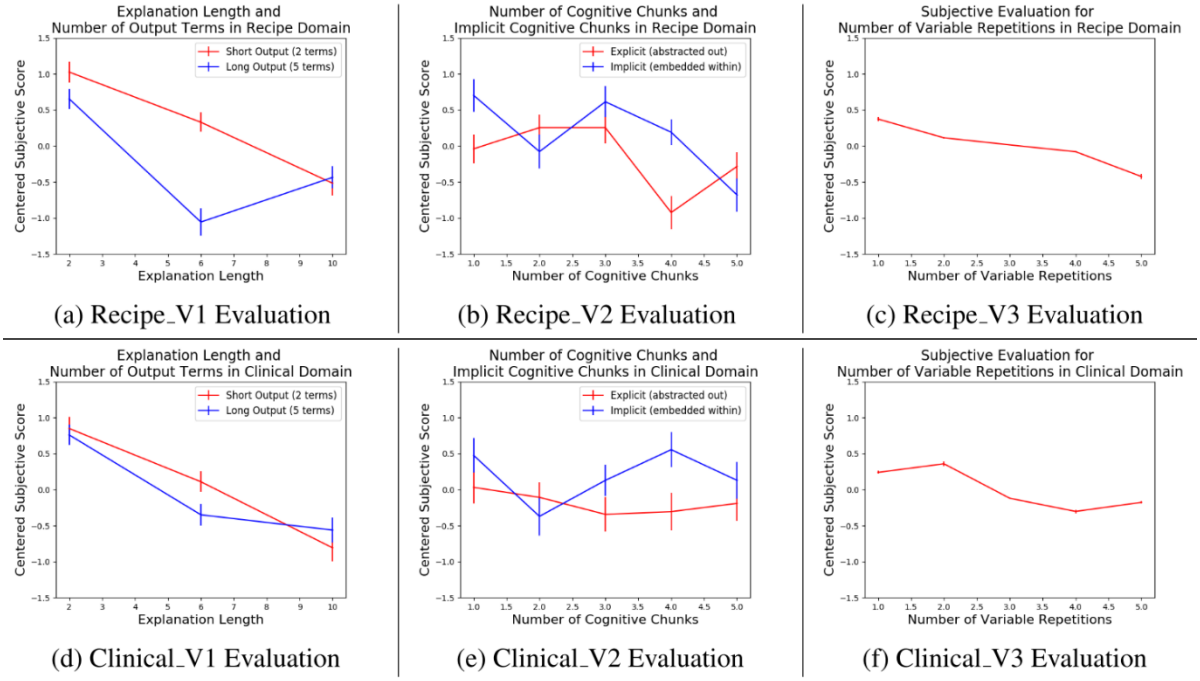


Figure 10: Narayana et al.[4] *results showing survey results of 100 participants answers in explanation and satisfaction with an explanation.*

In 2019, Stylianou et al.[18] showed the contribution of CNN layers through a similar method as Zintgraf et al.[15] by calculating the pixel location contribution of each layer for an image. The focus of the output was on the higher pooling layers of a similarity network and not the same as Zintgraf et al.[15] multiple layers throughout the CNN (See Figure 11).

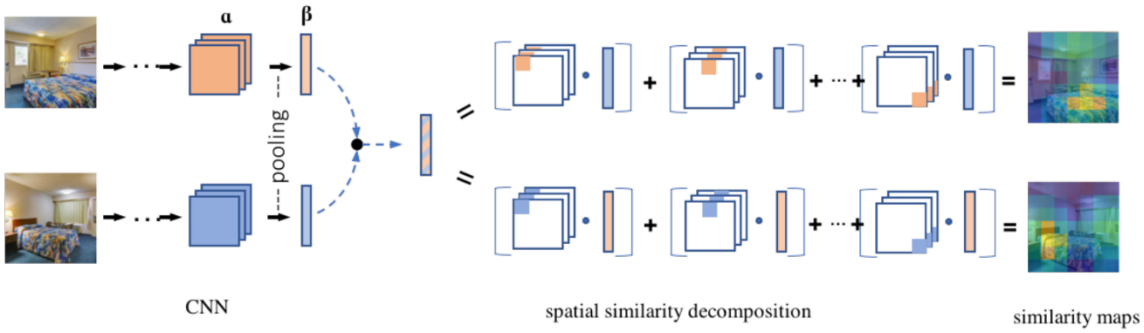


Figure 11: Stylianou et al.[18] *using a similarity network and a CNN combined through a pooling layer to display a higher-level similarity map of activation of features displayed as a heatmap overlayed on the image.*

In 2020, Zurowietx and Nattkemper presented an interactive visualisation application to show activation of an image based on which pixel the cursor was hovering over at the time. The application could be used to load any individual file that represented the CNN output image datasets. Once the dataset was loaded, the image could be overlaid in the background and a transparency value set to the heatmap overlayed on the image. Object detection and edge detection can be seen using the heatmaps generated from layer activations (See Figure 12). The application included a zoom slider to indicate which layer in the dataset would be currently focussed on[19].

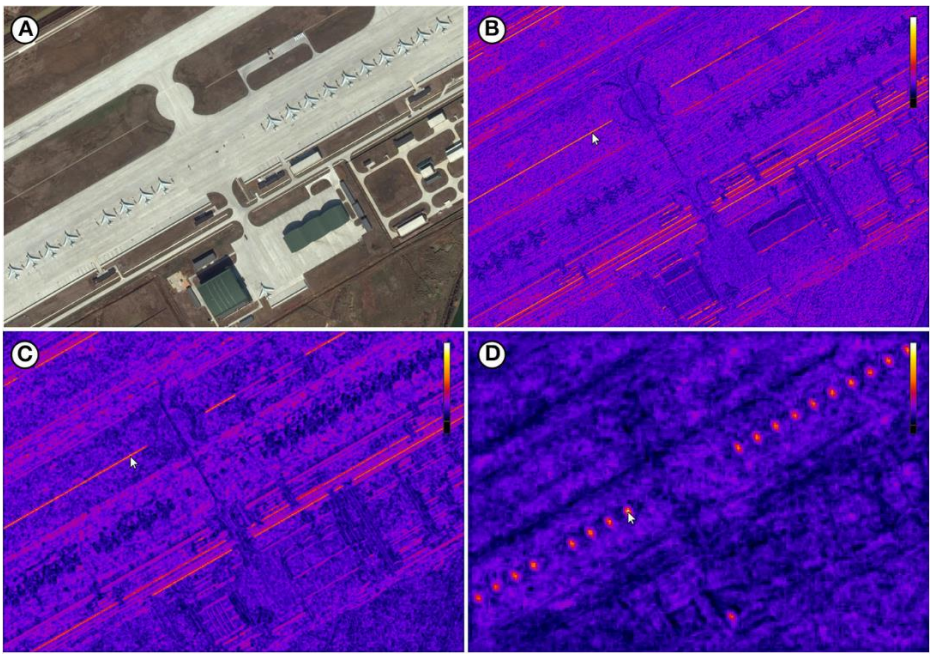


Figure 12: Zurowietx and Nattkemper[19] *shows several heatmaps of the input image, A) Input image - B) Low-level edge detection shown as brighter pink for higher activation, C) One layer higher than (B) and showing higher focussed edge detection - D) higher-level layer showing object detection in the individual planes.*

The Tensorflow Playground[20] introduced a way to explain machine learning aspects, showing layers, nodes and their expected output (See Figure 13). The visualisation is utilised more to explain machine learning principles rather than explain what steps and outputs are taken in a particular dataset and machine learning model.

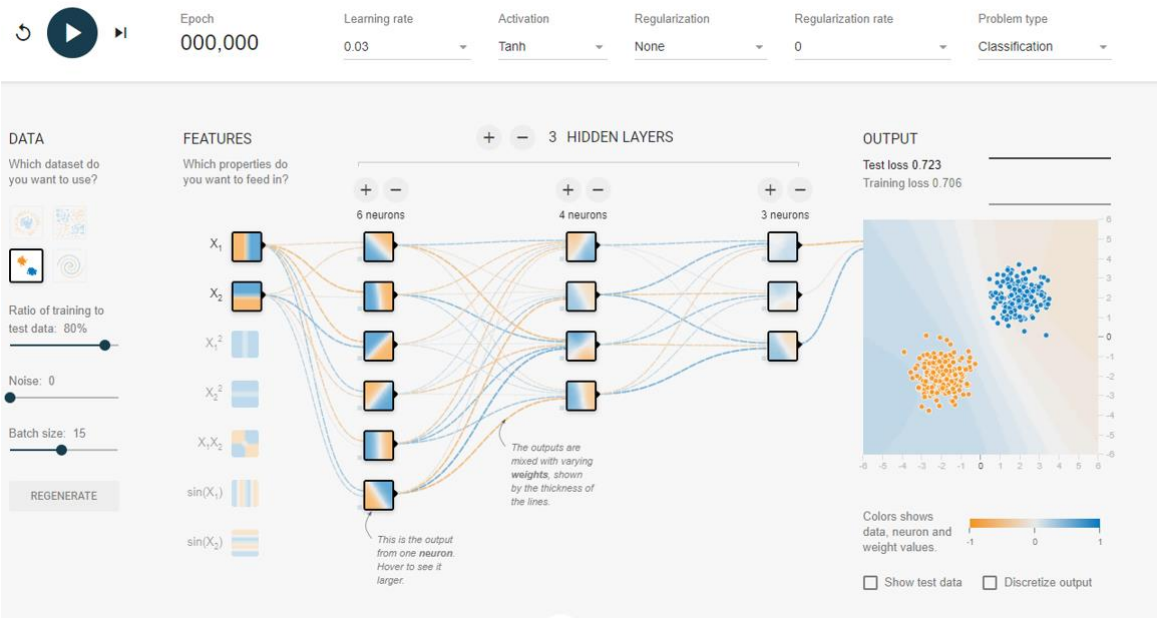


Figure 13: Tensorflow playground[20] *explainability dashboard. Each of the values can be changed to represent a neural network layout and problem type. The visualisation cannot take in a dataset and only shows clustering through a method to explain how machine learning layers work.*

2.3 User-Centric Design

User-Centric designs (UCD) aim to figure out what their users require within each stage of the design process. The whole UCD method is an iterative approach where designers use the user's input at each design step. The aim is to create accessible and usable products[21].

UCD can also get referred to as user-driven design (UDD)[22]. UCD and UDD are a framework of actions that uses usability goals at each design stage[23]. The principles of UCD, which got developed by Rubin, are[23]:

- Early focus on users and tasks
    - Structured and precise information-gathering, which needs to be consistent throughout all areas.
    - Experts train designers before conducting data collection sessions.
- Empirical Measurement and testing of product usage
    - Focus on ease of learning and ease of use
    - Testing of prototypes with actual users
- Iterative Design
    - Product designed, modified and tested repeatedly.
    - Allow for the complete overhaul and rethinking of design by early testing of conceptual models and design ideas.

Conclusively, UCD design needs to get based on the understanding of a user, their requirements, their experiences and what they expect to see within the application. When UCD gets used, this usually leads to increased end-user productivity and satisfaction[4], [24].

This process is vital for our project as we need to make sure that our end user will be happy with the designed outcome. As the user will be using the application regularly, we need to ensure that we are keeping them involved in every step to make sure we produce something that creates an app that will aid their workflows and increase their productivity. If the application does not benefit them in any way, then it would be hard to justify the apps benefits and use.

2.4 UI/UX Theory

The user interface (UI) is the most important of any application, as this is the part that the users will be interacting with. When the UI gets done well, users don't even notice it[25]. However, when the UI gets done poorly, the users can't see past it and will stop using the application[25]. Three of the main researchers who provided the foundations to software design are Ben Shniderman[26], Jakon Nielsen[27] and Bruce Tognazzini[28].

The main design principles for UI are[25]:

- Place users in control of the interface
- To make it comfortable to interact with
- Reduce cognitive load
- make user interfaces consistent

So, where UI is about how the user interface looks and acts, the user experience (UX) is more about having a deep understanding of the users[29]. UX research aims to find information about what the users need, value, abilities, and limitations[29].

There are multiple ways that UX research can get conducted. These include project management, user research, usability evaluation, information architecture (IA), user interface design, interaction design (IxD), visual design and web analytics[29]. Studies have suggested the UX is more of a mindset rather than a specific method[30].

Therefore, as our app is a UCD, we need to ensure that we are doing everything we can to make sure the user will enjoy the app. The best way to do that is by having a well thought out UI, which will be achieved by using appropriate UX research pre-developing and during development.

2.5 Literature Summary

The literature shows that visualisations around machine learning and neural networks visual how a network result is output or how a network layer is making a decision[1], [2], [5], [7], [9]–[11]. Recently some work has been completed on explaining deeper layers of a neural networks decisions and visualising these in a user-centric way[9] and visualisations considered from an interactive user perspective rather than an informative system only[31].

3 Software Methodology

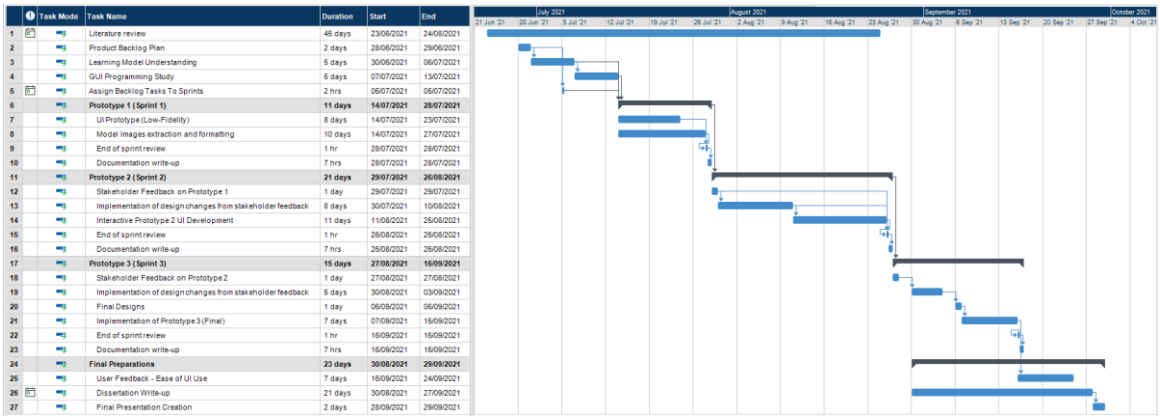
Scrum was used as the software methodology for this project. Agile software methodologies work well to ensure a viable prototype/working product is achieved at the end of each sprint. In a single developer team environment, it was important to include the stakeholders at every meeting to discuss project requirements and expectations for the next sprint. The initial plan was to include stakeholders in a weekly meeting to discuss progress and input to the human-centric user interface aspects of the application.

- Task Reflection
    - Daily Meetings - Developer only
        - 15-minute meeting
            - Progress since yesterday?
            - Suggested improvements on yesterday?
            - Task planning for today?
    - Weekly Meeting - Developer and Stakeholders
        - Once per week, developer only
            - Progress since the last meeting?
            - Suggested improvements to implement?
            - Feedback on design suggestions and implementation of application
                - Ease of use, etc.
            - Time plan Review
        - Sprints lasted between 1-4 weeks
            - End of sprint meeting – Developer only
                - Same questions as 15-minute daily meetings
                - Manage Product Backlog
                - Time plan reflection
            - Start of sprint meeting
                - Plan next sprint using stakeholder meeting feedback and product backlog
                - Keep the schedule on track and make any needed changes to the schedule

- Weekly Meeting – Developer and Academic supervisor
  - Supervisor meeting
    - Discuss progress
    - Review time plan
    - Ensure project is on track
    - Discuss next steps and targets

3.1 Schedule

The schedule was planned using Scrum agile methods and a 1-4 weekly sprint. Parallel writing and development were implemented as part of the whole project span.



	Task Mode	Task Name	Duration	Start	End
1		Literature review	45 days	23/06/2021	24/08/2021
2		Product Backlog Plan	2 days	28/06/2021	29/06/2021
3		Learning Model Understanding	5 days	30/06/2021	06/07/2021
4		GUI Programming Study	5 days	07/07/2021	13/07/2021
5		Assign Backlog Tasks To Sprints	2 hrs	05/07/2021	05/07/2021
6		**Prototype 1 (Sprint 1)**	**11 days**	**14/07/2021**	**28/07/2021**
7		UI Prototype (Low-Fidelity)	8 days	14/07/2021	23/07/2021
8		Model Images extraction and formatting	10 days	14/07/2021	27/07/2021
9		End of sprint review	1 hr	28/07/2021	28/07/2021
10		Documentation write-up	7 hrs	28/07/2021	28/07/2021
11		**Prototype 2 (Sprint 2)**	**21 days**	**29/07/2021**	**26/08/2021**
12		Stakeholder Feedback on Prototype 1	1 day	29/07/2021	29/07/2021
13		Implementation of design changes from stakeholder feedback	8 days	30/07/2021	10/08/2021
14		Interactive Prototype 2 UI Development	11 days	11/08/2021	25/08/2021
15		End of sprint review	1 hr	26/08/2021	26/08/2021
16		Documentation write-up	7 hrs	26/08/2021	26/08/2021
17		**Prototype 3 (Sprint 3)**	**15 days**	**27/08/2021**	**16/09/2021**
18		Stakeholder Feedback on Prototype 2	1 day	27/08/2021	27/08/2021
19		Implementation of design changes from stakeholder feedback	5 days	30/08/2021	03/09/2021
20		Final Designs	1 day	06/09/2021	06/09/2021
21		Implementation of Prototype 3 (Final)	7 days	07/09/2021	15/09/2021
22		End of sprint review	1 hr	16/09/2021	16/09/2021
23		Documentation write-up	7 hrs	16/09/2021	16/09/2021
24		**Final Preparations**	**23 days**	**30/08/2021**	**29/09/2021**
25		User Feedback - Ease of UI Use	7 days	16/09/2021	24/09/2021
26		Dissertation Write-up	21 days	30/08/2021	27/09/2021
27		Final Presentation Creation	2 days	28/09/2021	29/09/2021

Figure 14: Project schedule in agile Scrum method to monitor and keep the progress of the project on track throughout the period.

Table 1: Zoomed in Task List for project Schedule

		Task Mode	Task Name	Duration	Start	End
1			Literature review	45 days	23/06/2021	24/08/2021
2			Product Backlog Plan	2 days	28/06/2021	29/06/2021
3			Learning Model Understanding	5 days	30/06/2021	06/07/2021
4			GUI Programming Study	5 days	07/07/2021	13/07/2021
5			Assign Backlog Tasks To Sprints	2 hrs	05/07/2021	05/07/2021
6			**Prototype 1 (Sprint 1)**	**11 days**	**14/07/2021**	**28/07/2021**
7			UI Prototype (Low-Fidelity)	8 days	14/07/2021	23/07/2021
8			Model Images extraction and formatting	10 days	14/07/2021	27/07/2021
9			End of sprint review	1 hr	28/07/2021	28/07/2021
10			Documentation write-up	7 hrs	28/07/2021	28/07/2021
11			**Prototype 2 (Sprint 2)**	**21 days**	**29/07/2021**	**26/08/2021**
12			Stakeholder Feedback on Prototype 1	1 day	29/07/2021	29/07/2021
13			Implementation of design changes from stakeholder feedback	8 days	30/07/2021	10/08/2021
14			Interactive Prototype 2 UI Development	11 days	11/08/2021	25/08/2021
15			End of sprint review	1 hr	26/08/2021	26/08/2021
16			Documentation write-up	7 hrs	26/08/2021	26/08/2021
17			**Prototype 3 (Sprint 3)**	**15 days**	**27/08/2021**	**16/09/2021**
18			Stakeholder Feedback on Prototype 2	1 day	27/08/2021	27/08/2021
19			Implementation of design changes from stakeholder feedback	5 days	30/08/2021	03/09/2021
20			Final Designs	1 day	06/09/2021	06/09/2021
21			Implementation of Prototype 3 (Final)	7 days	07/09/2021	15/09/2021
22			End of sprint review	1 hr	16/09/2021	16/09/2021
23			Documentation write-up	7 hrs	16/09/2021	16/09/2021
24			**Final Preparations**	**23 days**	**30/08/2021**	**29/09/2021**
25			User Feedback - Ease of UI Use	7 days	16/09/2021	24/09/2021
26			Dissertation Write-up	21 days	30/08/2021	27/09/2021
27			Final Presentation Creation	2 days	28/09/2021	29/09/2021

3.2 Risks

Risks are a part of life, and there can always be some that are unexpected. Planning for these risks and creating mitigation strategies are vital ways to overcome these risks should they arise. Risks can vary, with some risks having more of an impact on the project than others. Examples of these could be: -

- Technology Risk – Disruption of the project by system failures or outages
- Skills Risk – Tasks being performed by inexperienced people
- Schedule risk – Project or task takes longer than planned
- Scope creep – Key stakeholders and developers changing the initial requirements

Some risks will have a larger impact than others, and a risk assessment is normally undertaken to assess these impacts and is further expanded on throughout agile software methodology projects as more risks become apparent. Having a team of experienced people often offers an insight into potential risks earlier in the development cycle than an inexperienced team, as the risks will have likely been identified previously. To help identify risks, an FMEA approach could be used that rates the likelihood of the risk occurring with a low risk being categorised as one and a high risk categorised as a 10; an RPN score is then used to calculate the overall chance of a risk occurring. This is calculated by multiplying together the severity, likelihood and detection together.

- S = Severity of a risk
- L = How likely the risk is to occur
- D = How easy it is to detect the risk

Table 2: Top of Risks Table - Risks to the project, showing in FMEA, with Severity, Likelihood and Detection, which results in the RPN rank of the risk. Mitigation strategies are given for each risk to pre-plan alleviation of risk impact.

Risk	S	L	D	RPN	Mitigation strategies
Requirements unclear after complete literature review	9	3	9	243	• Revisit research to understand and further clarify the scope and requirements of the project. • Meet with the project supervisor to discuss requirements and possible improvements
The product does not meet the requirements of the expected aims	7	3	3	63	• Ensure requirements meet aims and project supervisor expectations
Loss of data	8	7	3	168	• Backup the data in multiple places o Version history with GitHub o Dropbox account utilized for backup of folder structure o Google Drive utilized as last resort backup (updated weekly)
Failure to stay in scope	8	4	2	64	• Ensure project aims are delivered and timeframe allows any additions after the initial aim is complete (e.g. Misaligned edges could be implemented in future, but only once initial aims are fully tested and complete)
Poor project planning after Literature review complete	8	5	3	120	• Examine timeframe at the end of each sprint, milestone and in daily meetings (in line with chosen methodology, Scrum) • Consider resources and timeframes to calculate if the timeline still feasible o If no longer feasible, consider trimming of project result to fit within the timeframe

Table 3: Bottom of Risks Table - Risks to the project, showing in FMEA, with Severity, Likelihood and Detection, which results in the RPN rank of the risk. Mitigation strategies are given for each risk to pre-plan alleviation of risk impact.

Inability to learn a technology within the timeframe of the project	7	4	5	140	• Contact project supervisor to see if any help can be provided to speed up learning of new technology (e.g., explain a step that is currently causing a roadblock) • Consider switching to technologies and/or algorithms that are already familiar
Failure to follow the project's methodology	7	4	1	28	• Revisit literature review and refamiliarize with the methodology chosen o Revisit project plan if required
Ill health of the resource	5	5	1	25	• Revisit time plan when recovered o Possible need to trim project deliverables to fit within the timeframe
Miscommunication between Supervisor and Student	4	3	5	60	• Arrange weekly meetings to ensure the goal of the project is kept on track and expected outcome is correctly envisioned
Employment conflicts with the amount of work required for project completion	6	5	1	30	• Possibility of requirements trimming • Prioritize workload between project and employment o Meet with supervisor to discuss options ▪ Both work supervisor and project supervisor
Coronavirus pandemic outbreak causing further lockdowns	2	5	9	90	• Project supervisor meetings can be performed online via video streaming (Zoom, Microsoft Teams, etc.) • Development affected if ill health of resource occurs
Unable to integrate wxWidgets effectively	10	3	3	90	• Study library early in the project to determine suitability • Consider alternative library or software
Failure to understand obtained Python code	6	3	6	108	• Contact the code developer via email to organise a meeting to discuss the code • Create the code from fresh using C++

4. Design

The design process of this application was approached using the EPSRC values[32]. The EPSRC values were combined with a human-centred design approach to allow human input into the tool being created. Human-centred design approaches taken in this project aims at making software and AI work together, enabling humans to gain insight and explainability into how machine learning models come to a decision.

4.1 Responsible Innovation and Ethics

This project used the EPSRC framework for Responsible Research and Innovation (RRI)[32]. The aim of using this framework was to include all involved stakeholders in discussions and design decisions to increase trust in the application and its processes. Weekly meetings were performed with all stakeholders: Ordnance Survey experts, Academic expert supervisors and the author. Each meeting allowed for time to discuss the progress of the project and any ethical issues that may appear, intending to build trust and transparency for the project. Design choices and decisions were also discussed, and changes were made at each stage in the project.

4.2 Initial Planning

The Initial planning stages involved several meetings with stakeholders (Ordnance Survey, Academic Supervisors and the Author). During these meetings, the dataset and machine learning model was discussed, and data pre-processing steps were performed. The machine learning classification neural network was provided by an Ordnance Survey stakeholder as the dataset that would be required to build a UI tool and visualisation around.

4.2.1 Data Pre-processing

The dataset abstracted from the model included a four to eight-layer neural network that could be abstracted in a hierarchical method. The neural network was compiled using Microsoft VSCode[33][34] and utilised the dataset of Toy Shape images from Microsoft's ml-basics GitHub repo[33]. The images output to a file structure for access by the visualisation application to be built as part of this project.

4.2.2 First Designs

In this stage, the project requirements and aims were set out. First designs were drawn by hand on post-it notes and later added to a brief electronic version (See Figure 15). A hierarchy system for selecting the network layers was chosen, which led to the selection of the nodes that would then display the filters with the top activations for that node. This stage spanned across three meetings, and initial thoughts and ideas were changed with regard to stakeholder feedback.

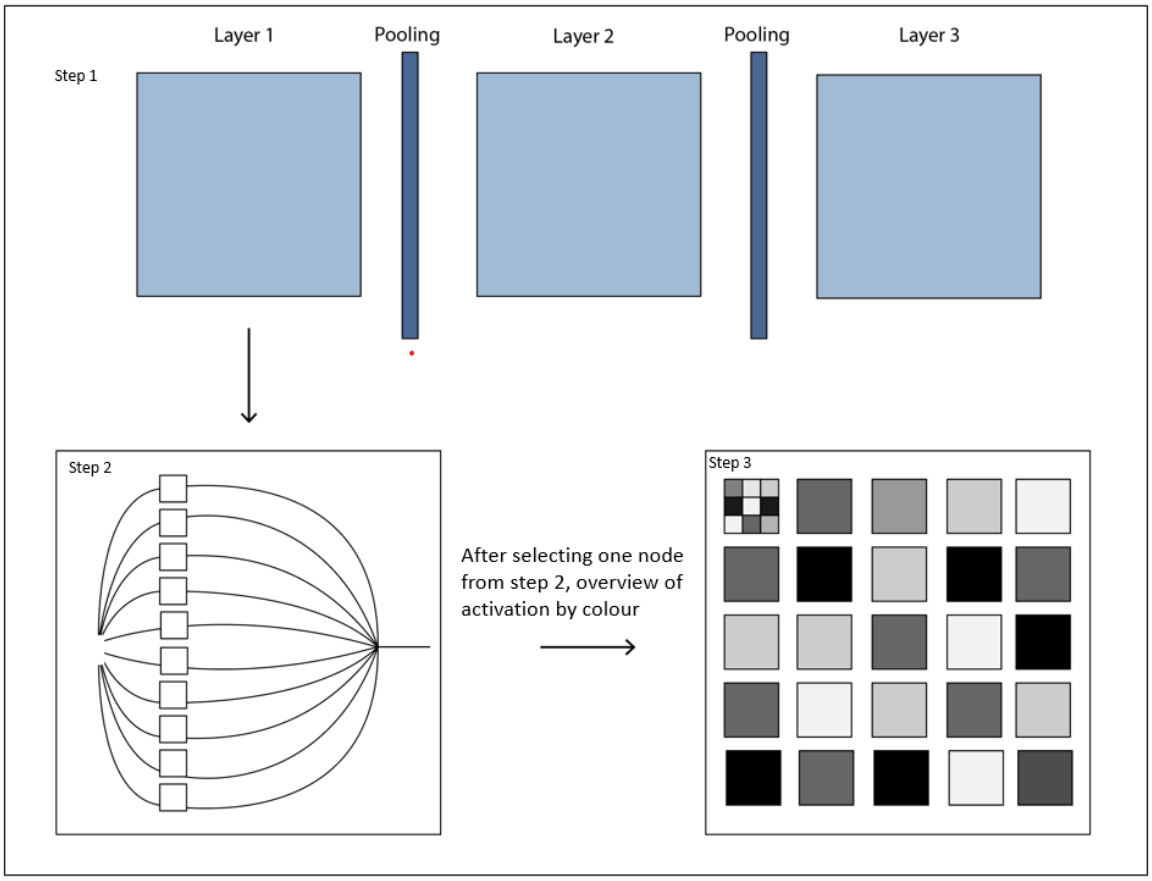


Figure 15: Initial planning image that shows a separation of the network layers (Step 1), nodes (Step 2) and filter activation (Step 3) for each node were key ideas for visualisation navigation within the application from the beginning stages of the project.

4.3 Interactive Prototypes

The next phase in the project progression was to build several interactive prototype designs. There were several designs shown in stakeholder meetings, and decisions were made to facilitate ease of use and transparency to the user.

4.3.1 Interactive Prototype 1

The initial prototype design was utilising static windows and a set layout that would always display all information to the user. This was designed using principles of information visualisation set out in Munzner's 2014 book[31], Visualization Analysis and Design[31], showing information shown on screen without the need of switching between screens can aid in understanding and explainability[31]. Input from the supervisory team was core to the initial designs and the literature review showing the containing elements required. The Tensorflow Playground[20] tool was referenced in the design, which resulted in determining line based linking would cause difficulty in linking through-thickness of lines to show weighting; Munzner[31] also suggests thickness as a measure(area) is difficult for humans to process.

Multiple design suggestions were discussed throughout stakeholder meetings; some of these designs are shown in the Figures below.

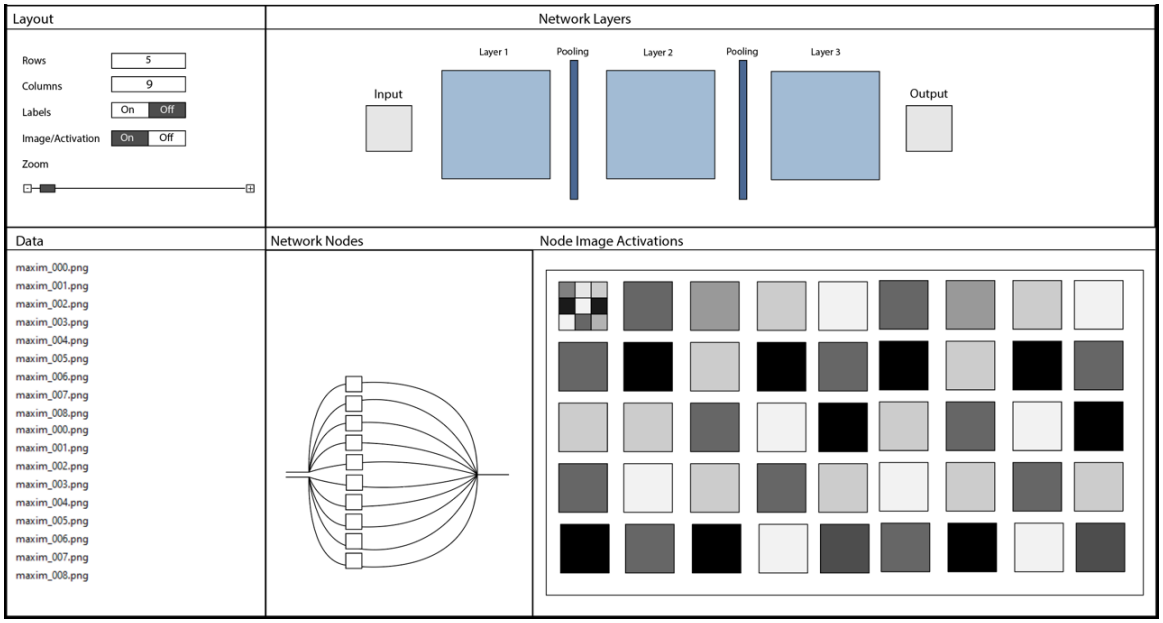


Figure 16: Interactive Design 1 – Showing a suggested layout that was discussed during stakeholder meetings. Network nodes would be later removed and added in the network layers node to allow more space for the node image activation window.

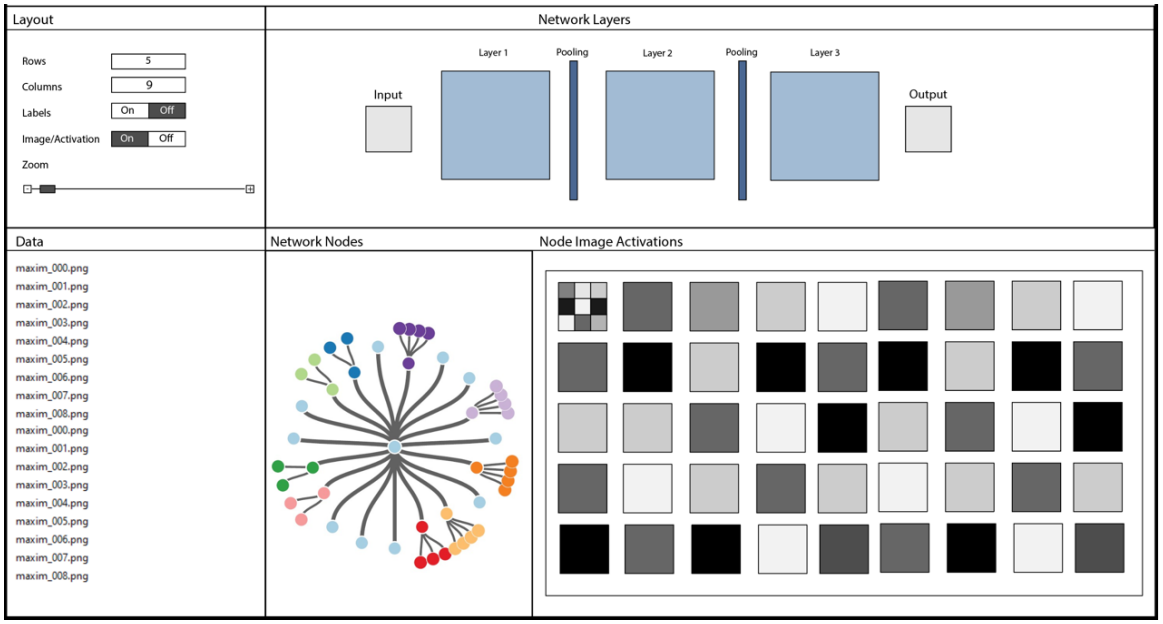


Figure 17: Interactive Design 2 – This design suggested a tree-like structure for the network nodes panel. The stakeholder meeting deemed that while the tree structure is visually appealing, it could lead to confusion as to which node is currently being selected in larger networks.

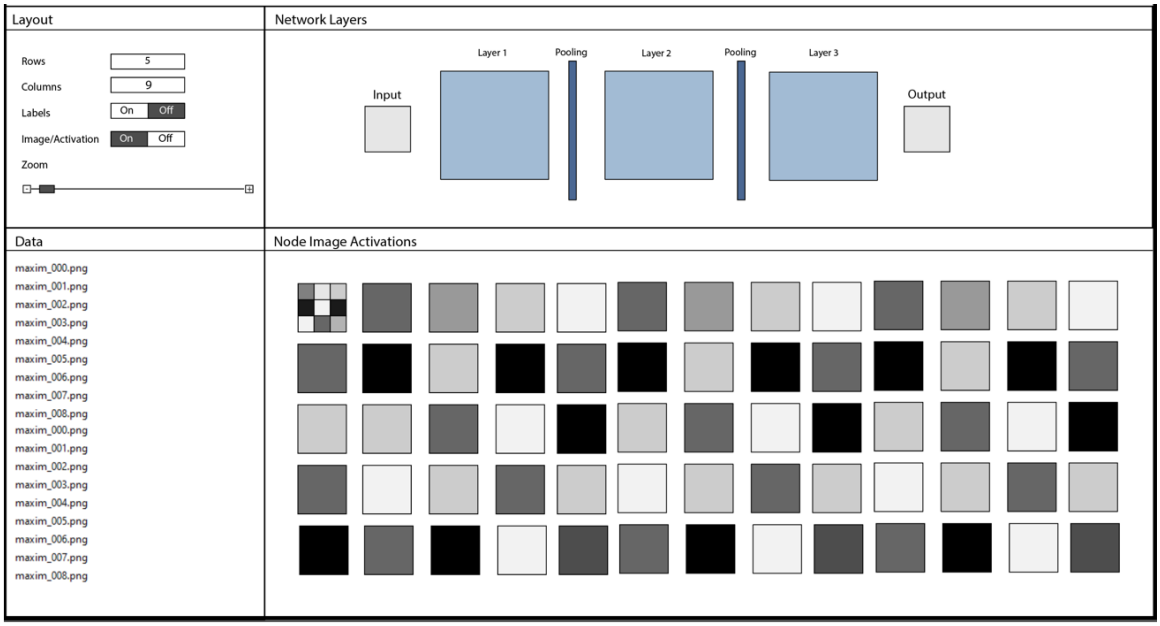


Figure 18: Interactive Design 3 – This design shows the removal of the network nodes panel, which was incorporated into the Network Layers panel: Once a layer has been selected, the panel will change to a network nodes output panel view (See Figure 19).

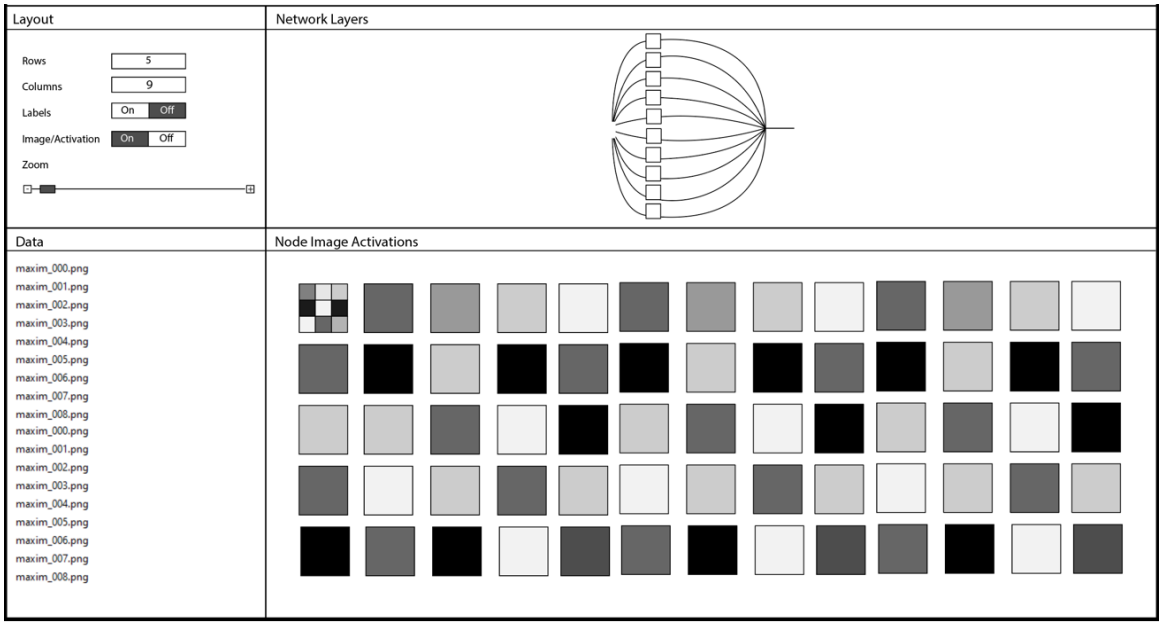


Figure 19: Interactive Design 3a – This design shows the network nodes panel: Once a layer has been selected, the panel will change to a network nodes output panel view you see in this image.

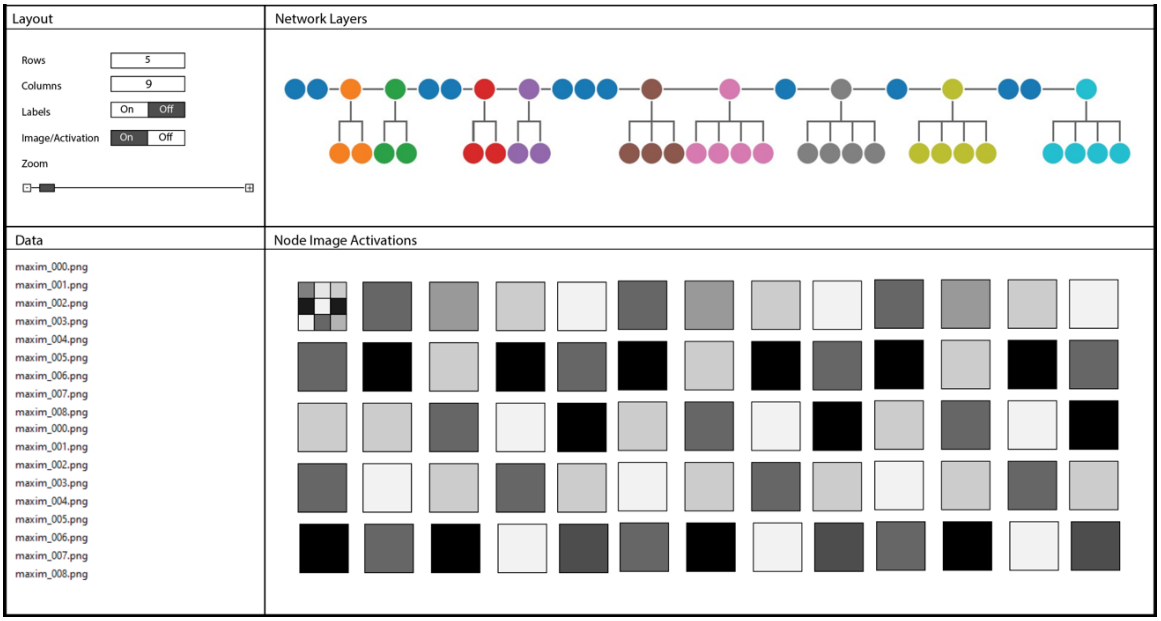


Figure 20: Interactive Design 3b – This design suggested a different layout to the node view. This view was debated in the stakeholder meetings and was deemed to be visually appealing but would make it difficult to show wherein the network a user currently was, especially for larger networks.

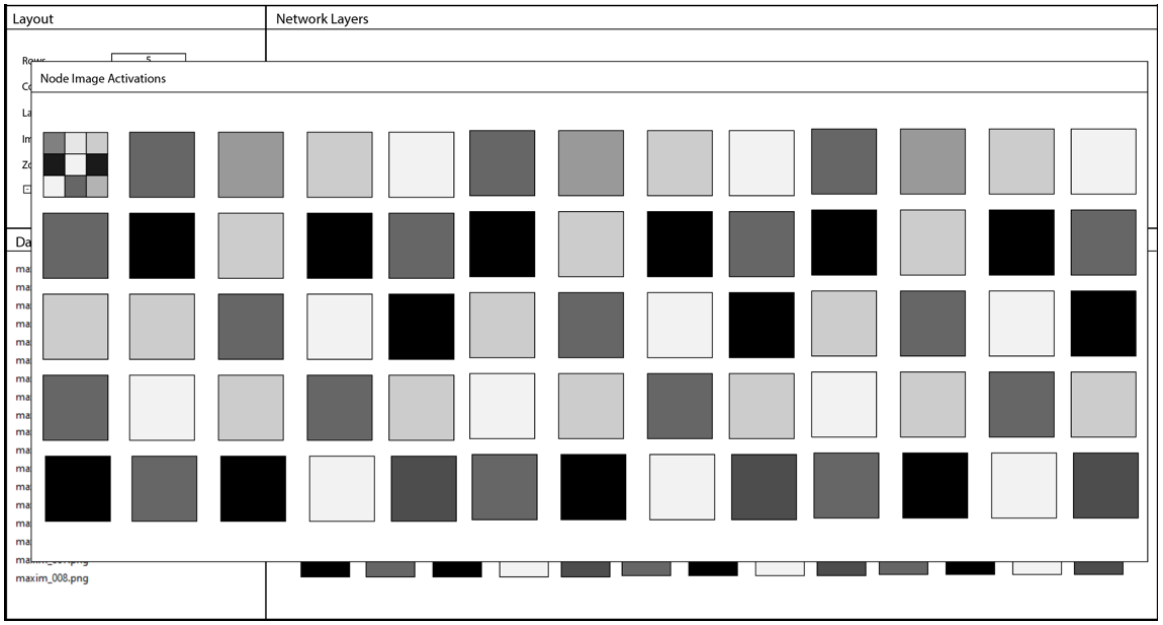


Figure 21: Interactive Design 4 - A pop-up panel was suggested to enable the viewing of large network nodes and higher than top 9 activations for node images.

4.4 Implementation Prototype Sections - wxWidgets

Once the initial draft design stages were discussed, the coding of the application began. The sectioning of each panel was implemented first, then the panels were adjustable to hide any of the panels and automatically filling the space with the remaining panels. During the stakeholder meetings and the literature review, it was decided minimising and maximising each panel using the panel title as a button would be the most effective implementation[31]. The choice to minimise instead of full movement per panel was decided to ensure ease of user interaction with the program with little to no explanation on where important window panels are at any time.

Figure 22: wxWidgets implementation of initial panel sectioning for the visualisation of the program.

14.5 Final Implementation - wxWidgets

The final implementation for visualisation was coded using VSCode[34] and wxWidgets[35]. The implementation visualisation was simplified for ease of use of the current dataset.

This implementation shows the following:

- Layout:
    - This panel allows changing of several variables; most notable is the columns, rows and zoom level for the ‘Node Image Activations’ panel.
        - The zoom level allows great flexibility in comparison with other methods of visualisation used in VSCode[34] and Jupyter notebooks[36].
- Layers:
    - Each layer in the network is displayed as a tab within the notebook panel ‘Layers’ to enable a hierarchical method of encapsulating the network nodes for each layer in the next panel
- Data:
    - The data tree enables selecting of a file path to load the output of a neural network model and the top activations for a certain node.

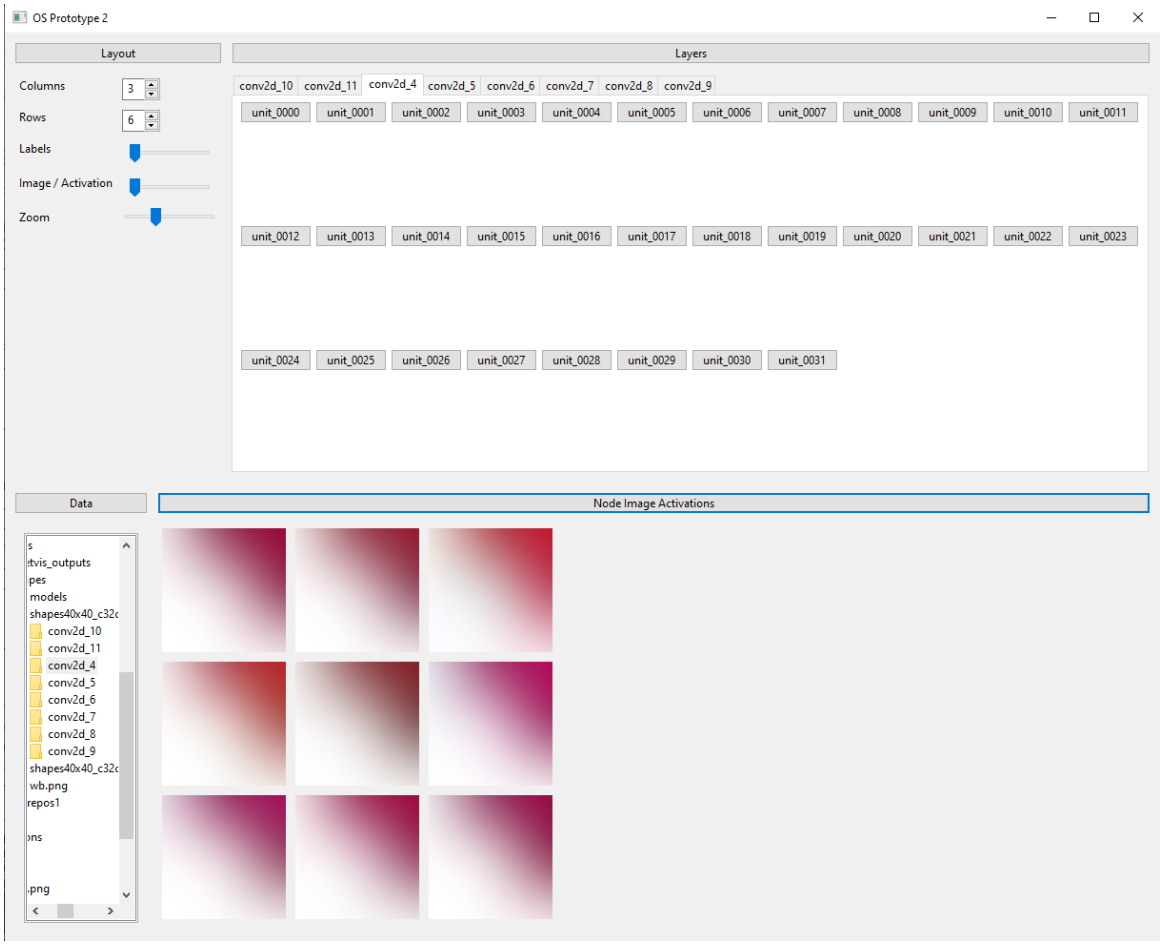


Figure 23: Final Implementation program output - Showing top 9 activations for a node within the network, showing all activations could be detecting diagonal edges or triangle objects at that node.

5. Evaluation

The project followed a qualitative approach through a formative evaluation method. Throughout the project, stakeholders were involved in several decisions to determine the best design choices to enable a human-centred software solution to be developed. User-driven design[21]–[23] was utilised as a part of the decision process with experts in each weekly meeting.

As discussed so far in the Design section, several issues with initial designs were detected through both the literature review and stakeholder feedback. Possible ethical and responsible research innovation issues were discussed in the initial meetings.

Initial planning discussions took place to determine the initial project proposal requirements of the stakeholders from Ordnance Survey and academic requirements. It was determined that an interactive application to display the top nine activation images of a layer was required to aid the explanation of a CNN decision-making process[1], [9], [13]. Discussions continued regarding the software choices for the project, landing on the decision to use wxWidgets as a cross-platform and free open source software solution[35]. The format of the data output from the CNN was discussed, and a couple of meetings (around one hour in length each) were attended by Ordnance Survey expert stakeholders and the author to further extract needs for the applications function utilising UCD and UDD[21]–[23], [26]. The final draft of the proposal was signed off by all stakeholders after edits were performed by the author, making requirements clear for the project and visualisation application. A schedule was then created by the author for the project, which was signed off by the project supervisory team and stakeholders.

Once initial planning stages were complete, and the application requirements were known, the author set out several initial design ideas drawn from the literature on UI/UX theory and machine learning explainability and visualisation sections[26], [31]. Figure 15 shows the separation of network layers that enabled the author to illustrate the process for node image activations to the stakeholders in a step-by-step process.

During stakeholder meetings before the interactive prototype stages would start, several application requirements were determined. The layout controls for the UI: the ability to adjust columns, rows, labels and zoom level of images. The need for a data path to be included for layers of the network (a variation from the layer zoom bar with no label

presented by Zurowietx and Nattkemper[19]) to be traversable from their labels and file structure to match the provided dataset.

The author created several interactive wireframe designs to show during a stakeholder meeting after discussion of required features previously, shown in figures 16-20. Key feedback from the supervisory team and Ordnance Survey on these designs further shaped the application output.

Visualisation methods that looked visually appealing were decided to not be the best selection to ensure the user is aware of the position they are at in the neural network (e.g., Tree visualisation vs hierarchical implementation). Figure 16 and 17's 'Network Nodes' panel was determined to be redundant and could be included as a deeper step in the 'Network Layers' panel to free up more screen space for the 'Node Image Activations' panel. The tree visualisation of the 'Network Nodes' panel was deemed nice to look at but confusing to understand/follow, so a grid-style button layout was adopted for simplicity reasons (See Figure 23). Similarly, in Figure 20, the 'Network Layers' top-down tree was switched to a tab with a page-based layout (See Figure 23) for usability.

The interactive wireframe visualisation helped determine which panels were a higher priority for the application, resulting in the 'Node Image Activations' panel being deemed the highest focus for the application, followed by the 'Network Layers' panel. Minimising of all panels other than 'Node Image Activations' was implemented from stakeholder user feedback of the wireframe prototype.

During the implementation of the wxWidgets prototype, the time being spent on colour theory and spacing was determined to be less useful to the application by the author and stakeholders and could be implemented with ease after completion as a further iteration[31]. However, this stage was useful for panel movement prototyping and visualisation to show the intended function of panels to stakeholders in meetings before work on the final prototype commenced (See Figure 22).

5.1 Results

The output of several layers is given in the below images. The visualisation results show how the application can be used to determine what each node in a network is attempting to detect. The detection is varied depending on the node, and it is up to the

human in the loop to decide if there is an underlying correlation between the top nine node activation images displayed in the application[1], [4], [9], [11], [13].

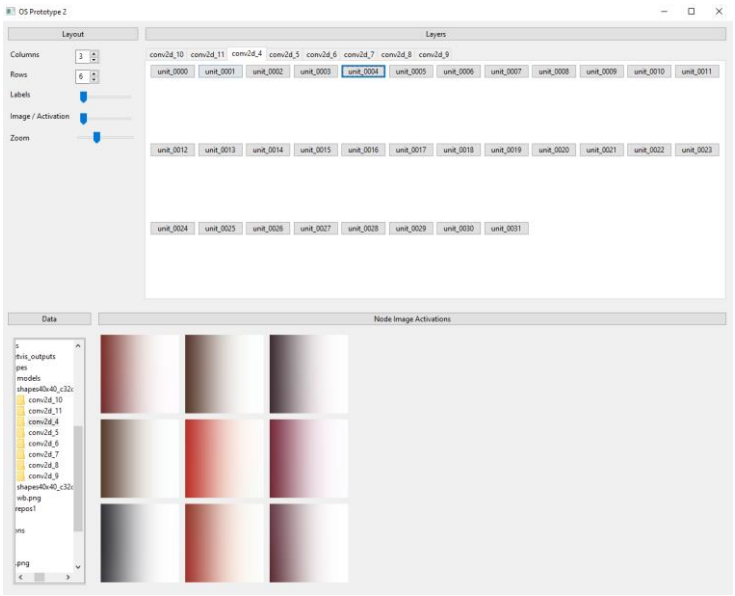


Figure 24: Final Implementation program output: Example 2 - Showing top 9 activations for layer conv2d_4 and node unit_0004 within the neural network output images, showing all activations could be detecting vertical edges or left-hand rectangle objects at the node.

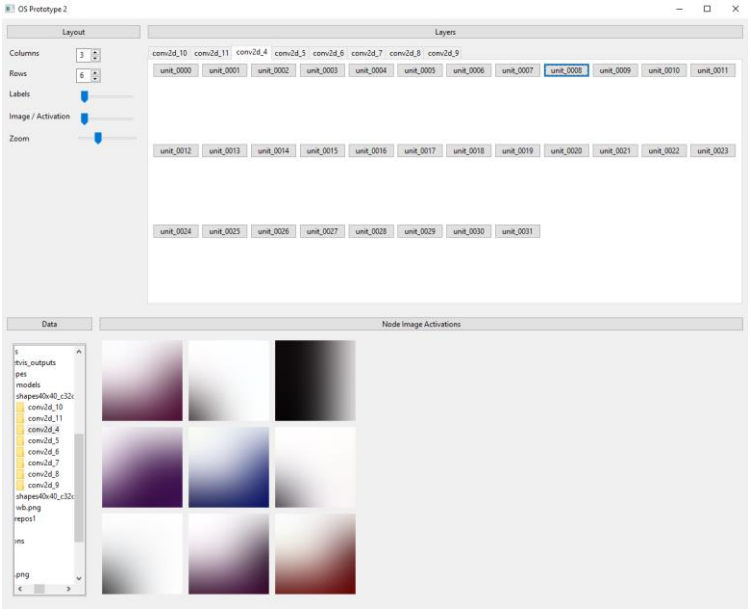


Figure 25: Final Implementation program output: Example 3 - Showing top 9 activations for layer conv2d_4 and node unit_0008 within the neural network output images, showing all activations could be detecting curves at this node. This output is not as clear to determine the detection method at the current node as in previous figures.

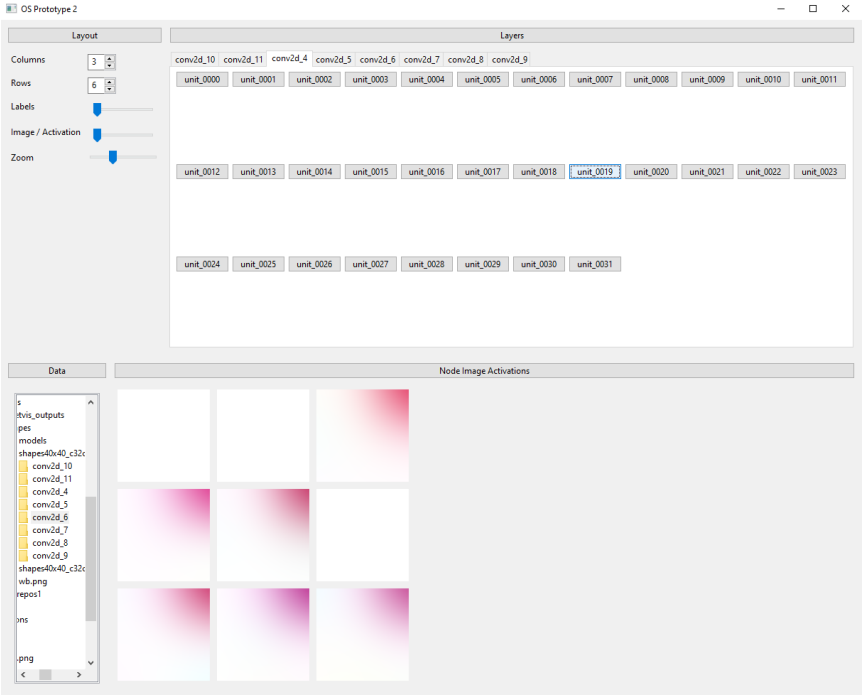


Figure 26: Final Implementation program output: Example 4 - Showing top 9 activations for layer conv2d_4 and node unit_0019 within the neural network output images, showing all activations could be detecting circular quadrants in the top left of the filter.

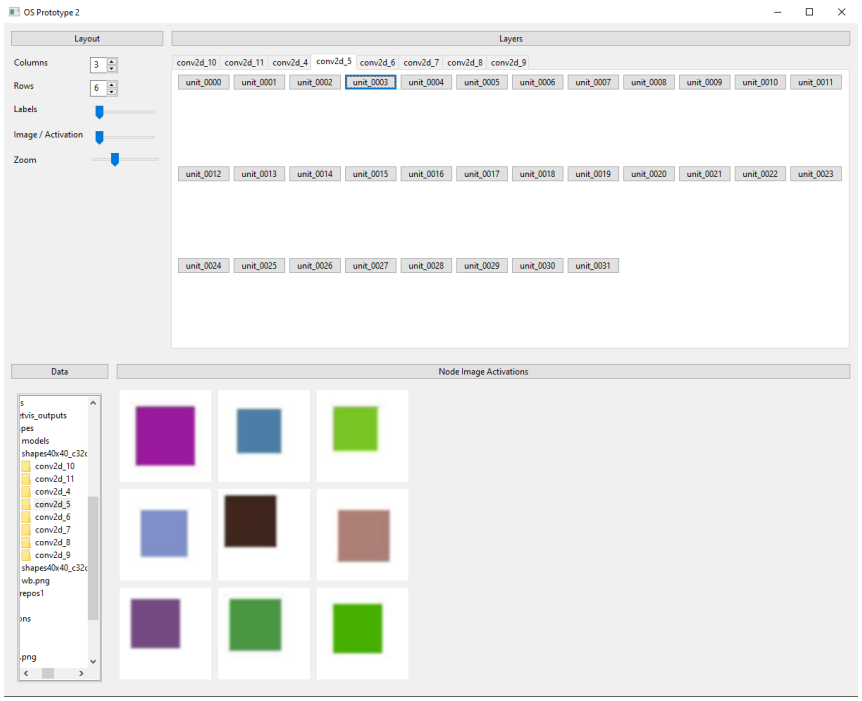


Figure 27:Final Implementation program output: Example 5 - Showing top 9 activations for layer conv2d_5 and node unit_0003 within the neural network output images, showing all activations could be detecting full objects (squares in this case) as the next layer is traversed in the neural network.

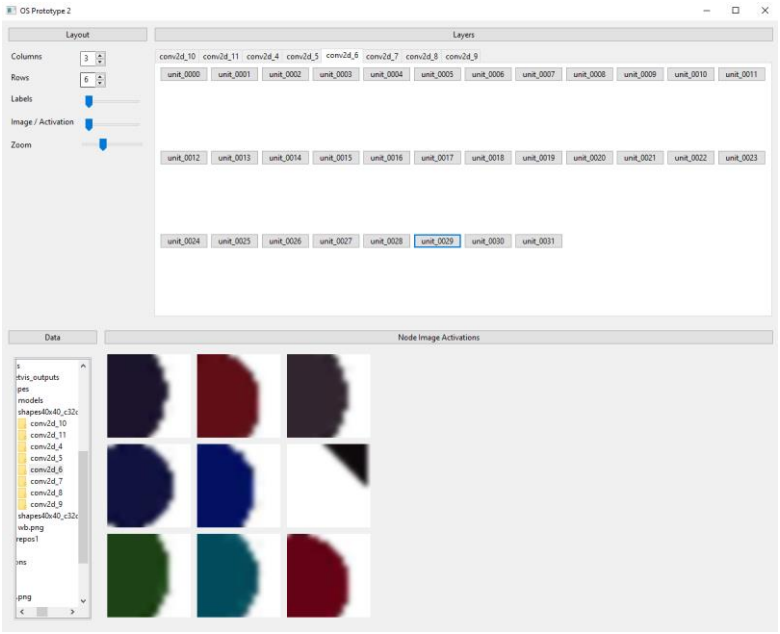


Figure 28: Final Implementation program output: Example 6 - Showing top 9 activations for layer conv2d_6 and node unit_0029 within the neural network output images, showing all activations could be parts of the objects detected in the previous layer (conv2d_5, See Figure 27).

5.2 Future Work

In future iterations of the application, some additional features are planned for implementation.

The main improvement for this project would be the use of real datasets of greater in size than the toy shapes dataset (such as ImageNet[14]) to test how the application works in displaying a higher density of layers and nodes and how this affects usability and explainability.

The ability to select nodes and image activations could be implemented with the appearance of a note box to allow the user to save notes on nodes and images with their thoughts. These notes could then be automatically (using image classification to match labels with features) generated or manually tagged with keywords that would enable to user to filter which nodes were detecting certain features. The filtering of nodes to match tags could enable a user to make more informed decisions on layers and node settings while building their neural network models.

The user interface styling and colour scheme could be updated to allow different colour themes and graphical icons with links could replace the Layers and their nodes. The ability to open several layers and Node Image Activations panels could be implemented for users to utilise side by side comparison of multiple nodes at the same time.

In future iterations, more user feedback will be used. User surveys and focus groups made up of general users would be beneficial to increase the usability of the application across user skill levels.

6. Reflection

This project was a very interesting area of study, and I was really looking forward to starting. A few issues arose with my health and the chosen library (wxWidgets) implementation method. If I were to redo this project from the beginning, I would possibly investigate a faster implementation method for the time requirements of the project. A web-based application where user interface design choices could be implemented quickly to test and create survey-based feedback could have been a benefit to this project from the beginning.

I have learned from this project that user feedback from survey-based participation requires a large amount of time to ensure enough numbers have participated and an intuitive survey can be created. Time planning in a one-person team was difficult to abide by when issues arose and will be taken into consideration in future project timelines.

I also learned the value of stakeholder input to user design. User input from experts is an invaluable design tool and should be utilised as often as possible throughout the design process. Things such as expanding windows and multiple view panels can be easy to overlook during the development process. The process of determining requirements was determined in a short period and allowed confidence in the usefulness of the implemented features within the application.

I have thoroughly enjoyed this project and working with all supervisors and Ordnance Survey stakeholders.

7. Conclusion

During the literature review stage, it was seen that most works had been directed at explaining the final output of a neural network results instead of the process of how these decisions were reached. High-level layers were explained, and individual levels were also explained, but there was a lack of explaining links between layers and the decisions between them.

An application was completed to allow navigation throughout several neural networks' layers and their nodes. Each node was able to show the top nine activations for that node to the user in a side-by-side adjustable layout form. The application was implemented to aid users in the explanations of how a neural network is making its decisions across multiple layers and nodes. During the design of the application, a human first user-centric design process was undertaken. The author and all stakeholders met weekly to discuss design changes and options. These design changes were implemented by the author to enable ease of use of the application for users.

8. References

[1] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, "Visualizing higher-layer features of a deep network," *2009VisualizingHF*, no. 1341, pp. 1–13, 2009.[Online]. Available: https://www.researchgate.net/profile/Aaron-Courville/publication/265022827_Visualizing_Higher-Layer_Features_of_a_Deep_Network/links/53ff82b00cf24c81027da530/Visualizing-Higher-Layer-Features-of-a-Deep-Network.pdf.

[2] G. G. Towell, "Extracting Refined Rules from Knowledge-Based Neural Networks," 1993.

[3] Y. Bengio, "Learning deep architectures for AI," *Found. Trends Mach. Learn.*, vol. 2, no. 1, pp. 1–27, 2009, doi: 10.1561/2200000006.

[4] M. Narayanan, E. Chen, J. He, … B. K. preprint arXiv, and undefined 2018, "How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation," *arxiv.org*, 2018, [Online]. Available: https://arxiv.org/abs/1802.00682.

[5] H. E., OsinderoSimon, and TehYee-Whye, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006, doi: 10.1162/NECO.2006.18.7.1527.

[6] "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges." http://yann.lecun.com/exdb/mnist/.

[7] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," pp. 1605–1612, Mar. 2009, doi: 10.1109/CVPR.2009.5206545.

[8] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nat. 1996 3816583*, vol. 381, no. 6583, pp. 607–609, 1996, doi: 10.1038/381607a0.

[9] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2018–2025, 2011, doi: 10.1109/ICCV.2011.6126474.

[10] A. G. Ramakrishnan, S. Kumar Raja, and H. V. Raghu Ram, "Neural network-based segmentation of textures using Gabor features," *Neural Networks Signal Process. - Proc. IEEE Work.*, vol. 2002-January, pp. 365–374, 2002, doi: 10.1109/NNSP.2002.1030048.

[11] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "HOGgles: Visualizing object detection features," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1–8, 2013, doi: 10.1109/ICCV.2013.8.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *Proc. - 2005 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, vol. I, pp. 886–893, 2005, doi: 10.1109/CVPR.2005.177.

[13] M. D. M. Zeiler, R. F.-E. conference on computer Vision, U. 2014, and R. Fergus, "Visualizing and understanding convolutional networks," *Springer*, vol. ECCV 2014, no. Part I, pp. 818–833, 2014, [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53.

[14] KrizhevskyAlex, SutskeverIlya, and H. E., "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[15] L. M. Zintgraf, T. S. Cohen, and M. Welling, “A New Method to Visualize Deep Neural Networks,” Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.02518.

[16] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps,” *undefined*, 2014, [Online]. Available: http://code.google.com/p/cuda-convnet/.

[17] M. D. Zeiler and R. Fergus, “LNCS 8689 - Visualizing and Understanding Convolutional Networks,” 2014.

[18] A. Stylianou, R. Souvenir, R. P.-2019 I. Winter, and U. 2019, “Visualizing deep similarity networks,” *ieeexplore.ieee.org*, 2019, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8659098/.

[19] F. Riguzzi, M. E. Papka, T. Ertl, M. Zurowietz, and T. W. Nattkemper, “An Interactive Visualization for Feature Localization in Deep Neural Networks,” *Artif. Intell*, vol. 3, p. 49, 2020, doi: 10.3389/frai.2020.00049.

[20] “A Neural Network Playground.” https://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=0&networkShape=4,2&seed=0.04620&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false.

[21] “What is User Centered Design? | Interaction Design Foundation (IxDF).” https://www.interaction-design.org/literature/topics/user-centered-design.

[22] E. Lucchi and A. C. Delera, “Enhancing the historic public social housing through a user-centered design-driven approach,” *Buildings*, vol. 10, no. 9, Sep. 2020, doi: 10.3390/BUILDINGS10090159.

[23] “Notes on User Centered Design Process (UCD).” https://www.w3.org/WAI/redesign/ucd.

[24] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey, “A Survey of User-Centered Design Practice,” *Proc. SIGCHI Conf. Hum. factors Comput. Syst. Chang. our world, Chang. ourselves - CHI ’02*, 2002, doi: 10.1145/503376.

[25] “The 4 Golden Rules of UI Design | Adobe XD Ideas.” https://xd.adobe.com/ideas/process/ui-design/4-golden-rules-ui-design/

[26] “Ben Shneiderman.” https://www.cs.umd.edu/users/ben/goldenrules.html.

[27] “10 Usability Heuristics for User Interface Design.” https://www.nngroup.com/articles/ten-usability-heuristics/.

[28] “First Principles of Interaction Design (Revised & Expanded) | askTog.” https://asktog.com/atc/principles-of-interaction-design/.

[29] “User Experience Basics | Usability.gov.” https://www.usability.gov/what-and-why/user-experience.html.

[30] C. M. Gray, “‘It’s more of a mindset than a method’: UX practitioners’ conception of design methods,” *Conf. Hum. Factors Comput. Syst. - Proc.*, pp. 4044–4055, May 2016, doi:

10.1145/2858036.2858410.

[31] T. Munzner and E. (Graphic artist) Maguire, “Visualization Analysis and Design,” *Cs.Ubc.Ca*, vol. 16, p. 404, 2014, [Online]. Available: https://www.cs.ubc.ca/~tmm/talks/minicourse14/vad17stat545-4x4.pdf%0Ahttps://books.google.com.co/books?hl=en&lr=&id=dznSBQAAQBAJ&oi=fnd&pg=PP1&dq=Visualization+Analysis+and+Design&ots=HfNtFwMbMq&sig=5Voltt5TZziDkT5sVf-JoQDjP3w%0Ahttps://www.crcpress.com/V.

[32] “Research - EPSRC website.” https://epsrc.ukri.org/research/.

[33] “GitHub - MicrosoftDocs/ml-basics: Exercise notebooks for Machine Learning modules on Microsoft Learn.” https://github.com/MicrosoftDocs/ml-basics.

[34] “Visual Studio Code - Code Editing. Redefined.” https://code.visualstudio.com/.

[35] “wxWidgets: Documentation.” https://docs.wxwidgets.org/3.0/.

[36] “Working with Jupyter Notebooks in Visual Studio Code.” https://code.visualstudio.com/docs/datascience/jupyter-notebooks.

A.1 Appendix – Ethical Issues

This project does not include any outside user data or user interaction. Only the author and stakeholders were involved in decision making based on the referenced literature review.