Relevance in the eye of the beholder

Christian Lie

DIVISION OF PRODUCT DEVELOPMENT, DEPARTMENT OF DESIGN SCIENCES,
FACULTY OF ENGINEERING LTH, LUND UNIVERSITY
2019

MASTER THESIS

Relevance in the eye of the beholder

Diagnosing classifications based on visualised layerwise relevance propagation.

Christian Lie

LUND
UNIVERSITY

Relevance in the eye of the beholder

Diagnosing classifications based on visualised layerwise relevance propagation.

Published by

Department of Design Sciences
Faculty of Engineering LTH, Lund University
P.O. Box 118, SE-221 00 Lund, Sweden

Subject: Technical Design (MMKM10)
Division: Product development
Supervisor: Joakim Eriksson
Co-supervisor: Markus Klang
Examiner: Axel Nordin

Abstract

This thesis examines a method for how humans can assess quality of classifications by image based neural network. Through examples visualisations of layerwise relevance propagation (Bach et al. 2015) test participants are tasked with learning to differentiate correctly and incorrectly classified images. To test their learning, they are tasked with evaluating visualisations based on whether they believe the underlying classification to be correct or not. The test was conducted for 20 test participants who after 10 learning examples each were tasked with evaluating the same 72 visualisations. Three different types of visualisations are developed according to different mathematical principles.

The results show that the test was often confusing for the test participants indicating both that better preparation and training of test participants might be appropriate but also that the subject is inaccessible to many. There are also small beneficial effects which are expected to increase with better training and preparation.

Keywords: Neural network, Machine Learning, Visualisation, User testing, Layerwise relevance propagation, LRP

Sammanfattning

Denna uppsats undersöker en metod för hur människor kan utvärdera kvaliten på klassificeringar från ett bildbaserat neuralt nätverk. Testdeltagare ombeds studera visualiseringar av an analysmetod, kallad layerwise relevance propagation (Bach et al. 2015), för att lära sig differentiera korrekt och inkorrekt klassificerade bilder. För att testa vad de lärt sig får de sedan i uppgift att utvärdera visualiseringar baserat på om de tror att den bakomliggande klassificeringen är korrekt eller inte. Testet utfördes för 20 testdeltagare som efter 10 läroexempel fick utvärdera 72 visualiseringar var. Tre olika typer av visualisering utvecklades enligt olika matematiska principer.

Resultaten visar att testet ofta förvirrade testdeltagare vilket indikerar både att bättre förberedelse och träning av testdeltagare kan vara lämpligt men också att området kan vara svårtillgängligt för många. Resultatet påvisar även små positiva effekter vilka förväntas öka med bättre förberedelse och träning.

Nyckelord: Neurala nätverk, Maskininlärning, Visualisering, användartest, Layerwise relevance propagation, LRP

Acknowledgments

I would like to thank Joakim Eriksson for sharing his experience in user testing and guidance which helped keep this thesis on the move. I would also like to thank Markus Klang for his patience and expertise in machine learning.

I would like to thank my fellow thesis students for sharing my defeats and victories, you've kept me sane. I would like to thank everyone who participated in the study for their generosity.

I would like to thank Johanna Wendesten for always being herself and helping me being myself.

All of you made this one-person thesis not only possible but also never feel the least lonely.

Thanks.

Lund, January 2019

Christian Lie

Table of contents

1 Introduction

This chapter will cover the background, goals, hypothesis and motivation of the thesis.

1.1 Background

As image processing neural networks improve they are bound to enter our lives through more and more application areas. In many cases they can be used to automate boring tasks which are trivial to a human actor. Such as classifying if an image shows a cat or a dog or determining where in an image there are cars, pedestrians and road signs. However, they can also be used to carry out image-based tasks which are not trivial to human actors, for example in detecting skin cancer based on images (Esteva et al. 2017).

As neural networks are applied to areas such as skin cancer diagnosis, trust is crucial. How can we know when we are to trust a neural network result and when not to? This is especially problematic as neural networks are mostly a black box technology where there is seldom a clear model for understanding how they operate.

Imagine an example with a neural network for sorting images, classifying them based on some characteristic not obvious to humans. The algorithm has been tested and is known to succeed in classification 80% of the time. However, when used in practice, on images for which the class is not known, there is still an issue. How can one tell which classifications to trust? There is a lot of use in knowing which 80% are correctly classified and which 20% are incorrectly classified.

One way of approaching this issue and somewhat addressing the issue of using a black box technology is to study and visualise the motivations behind each classification. If the visualised motivation is intuitively or analytically understandable this poses a big opportunity for communicating the trustworthiness of a classification.

To study the motivation behind classifications this thesis will be using an analysis method known as Layerwise Relevance Propagation. This method provides a relevance value for each pixel of the input and will be explained in further detail.

This thesis is a study on whether it is possible to separate the correct network classifications from incorrect ones by using visualised motivation.

1.2 Objectives

This thesis considers situations in which user analysis acts as a supplement to image classification algorithms. The main objective is to study the capacity of users to differentiate correct and incorrect classifications by the algorithm based on visualisations approximating motivation.

Secondary objectives include granting insight into how these visualisations affect the user perception of working alongside an algorithm with regards to trust and scepticism. Another secondary objective is to conduct an exploration into how analyses approximating motivation can be expressed.

1.3 Research Questions

The objectives are formalised as a series of research questions.

How effective are Layerwise Relevance Propagation visualisations for identifying wrongly classed images?

How effective are Layerwise Relevance Propagation visualisations for identifying correctly classed images?

By what mechanisms do LRP visualisations aid the user?

Are there differences in depending on how the LRP visualisations are created?

… with regards to earlier research questions?

… with regards to user experience

1.4 Hypothesis

The visualisations are expected to have a positive measurable effect on identifying both correct and incorrect classifications.

The visualisations are expected to help the test participants by mechanism of showing what the algorithm considers relevant pixels in the image and have the test participants compare that to their own intuition for what is relevant.

This can provide insight into whether humans have a strong general intuition for relevance against which the algorithms evaluation of relevance can be compared.

1.5 Thesis overview

The following chapters will detail the selection of the task to study, the creation of the network to study, an overview of the LRP analysis method, and creation of visualisations to communicate that analysis.

This thesis presupposes that for many image tasks humans have a well-trained ability of determining what aspects and areas of an image are relevant. This is often true even for areas in which we do not have expert knowledge. For example, even a person who knows nothing about different species of flowers can still tell them apart. One of the main things being studied is if this somewhat general intuition for what is relevant in an image can be repurposed to evaluate neural network classification.

The idea is that the network visualised motivation for what was relevant in the image can be matched against what the human intuition considers relevant and thus the user can determine the quality of the network classification.

To study the impact of visualisations for user analysis tests will be held. The test will be implemented with minimal preparation of the test participants and using an abstract task. This is to test is this human intuition for relevance is general enough to function even for a real-life task which might be beyond the abilities of a normal human. To observe a beneficial effect under these circumstances is supposed to test a “gold standard” for usability. A tool that is easier to use is far more valuable and easier to implement across tasks and disciplines.

NOTE: Please observe that while most of the report functions in black and white pages 36 through 46 need colours due to its integral role in the visualisations. Pages 70 and onward are also best viewed in colour.

2 Method

This chapter will cover the process of creating the necessary materials and procedures to conduct the study.

The user tests will be conducted similarly to the Rubin & Chisnell (2008, p29) manner of exploratory study. This is due to there not being much in the way of precedent for applying these kinds of visualisations to evaluate classification quality. Because of this several variants of visualisations will be examined to help build a foundation to direct further studies. This also means that the study should have qualitative aspects to help to understand how test participants interact with and interpret the visualisations.

2.1 Task selection

2.1.1 **Task requirements**

There are many restrictions when it comes to selecting suitable tasks, the goal is to have a task somewhat reflective of practical applications where it is hard for both user and algorithm to have absolute knowledge of the true classification of an image.

The first requirement is that the task should be non-trivial for algorithm. Preferably the algorithm should be able to classify 70-90% of test images correctly and produce both false positives and negatives. This requirement is very important but can be somewhat managed during the training of the network. The network training can simply be halted when the network has reached a good level of accuracy. It would be preferable to have the network train until it plateaus in terms of performance as that is how most real-life networks are trained.

The second requirement regards test participant intuition for which parts of the image are relevant. If this is very strong the user should be better suited to identify when the algorithm has attributed relevance to the wrong parts of the image. On the other hand, if the task is too abstract the user will have no framework to place the visualised motivation of the algorithm into. For example, it would be problematic

to show a visualisation of which parts of an EKG (Electrocardiography) motivated a classification since most users have no concept of what might be relevant.

The third requirement is that the images can't necessarily be too big. Larger images can increase the amount of processing power necessary to train the network out of a feasible range. It is also possible to downsize the images if this does not remove so much detail that it hinders the network extracting features or the user identifying them.

The fourth requirement is appropriate time use to maintain feasibility of the study. To large time use could end up limiting the number of test participants.

2.1.2 **Task candidates**

The first step in identifying task candidates was a search for appropriate datasets to train a network around. The primary candidates are listed below. These were selected due to having freely available datasets based on image identification and somewhat matching the task requirements. Meaning that they were all deemed to be somewhat non-trivial, enable some user intuition, contained images of a size that was viable for training and human analysis and would not create issues of time in the tests.

- Age estimation
- Flower classification
- Brain tumour classification

For each of these a definition was created along with a list of pros and cons. It should be noted that there might be many other datasets that fit the requirements. However, once three reasonably fitting ones had been found it was deemed to be time for selecting one and advancing the thesis work.

2.1.2.1 Age estimation

Age estimation would consist of feeding the network an image and having it return an estimate age number as a float value. This is somewhat special in that it involves linear estimation instead of sorting images into different classes. This linear quality means that the question would not be whether the algorithm has classified the image correctly or not but rather how close it is and how that correlates to the test participants confidence in the analysis.

As most people are accustomed to determining the age of other people, test participants can be expected to have good intuition for what image information is relevant.

Image size might be an issue as many facial details the test user might want to discern could be lost in low resolution images.

Time use should not be an issue as no prior training should be required beyond explaining the concept of the study.

The dataset would be the IMDB-wiki dataset by Rothe, Timofte and Van Gool (2015) consisting of 523000 images of cropped faces[1].

2.1.2.2 Flower classification

Flower classification would consist of feeding the network an image and having it classified as one of 102 species.

Image size should not be an issue as flowers often have their differences in relatively speaking larger features such as colour, number of petals, shape of petals. These are not easily lost in lower resolution images.

Test participants can be expected to have some intuition for which parts of the images are relevant as they should be adept at determining which parts of the images are the flowers and which are irrelevant background or foreground.

Time use should not be an issue as not much preparatory training should be required beyond explaining the concept of the study.

The dataset would be from the Oxford Visual Geometry Group, consisting of 8189 images divided among 102 classes (Nilsback & Zisserman, 2008)[2].

2.1.2.3 Brain tumour classification

Tumour classification would consist of feeding the network an image and having it classified as one of three different kinds of brain tumours. Possibly non-trivial

Test participants can be expected to have very little prior intuition for which parts of the images are relevant but might build some as the test progresses. Outliers might be medical professionals, especially radiologists.

It is hard for a layman to determine if image size would be an issue without knowing at what level of detail the tumours are discerned.

Time could very well be an issue as more preparatory training should be required so that the users might build some basic intuition for which parts are relevant.

Dataset from Jun Cheng of the Southern Medical University in Guangzhou consisting of 3064 images divided among 3 classes and 233 patients (2017)[3].

---

[1] The dataset can be found at: https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/

[2] The dataset can be found at: http://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html

[3] The dataset can be found at: https://figshare.com/articles/brain_tumor_dataset/1512427

2.1.3 **Selected task**

The flower classification task was selected for several reasons. The main one is that the average test participant can be expected to have some intuition for what areas are relevant. At the same time, it is more abstract than age estimation which could make the result more generalisable for different real-life applications. At the same time the dataset is of a very manageable size and the images would not suffer too much from being downsized. Time should not be too big a problem.

2.2 Creating the network

2.2.1 **Architecture**

While the following section contains solid academic sources, the author would like to suggest that a novice in neural networks visit the excellent blog of Chris Olah for better introductions to some of the various concepts.

http://colah.github.io

2.2.1.1 Tensors

Tensors are a basic way of structuring information in a neural network. A tensor is a multi-dimensional array of numbers (Goodfellow et al. 2016, 31). In this project the tensors in the network are filled with 32-bit floating point numbers. An example of a three-dimensional tensor is a 16 by 16-pixel RGB image which has 16 columns, each containing 16 rows which in turn contain 3 colour values, this tensor is said to have the shape 16x16x3. A collection of 10 such pictures would be a four-dimensional tensor with the shape 10x16x16x3. To identify an element in such a tensor, four indices are needed, one for each dimension (Goodfellow et al. 2016, 31). On the other hand, a vector would be a one-dimensional tensor.

2.2.1.2 Neurons and activation functions.

The neuron is the basic unit of a neural network. It takes a tensor of numerical inputs and processes them to create a scalar output (often known as activation) (Goodfellow et al. 2016, 165).

The processing can be described as two steps (Figure 2.1). First each of the connected inputs are multiplied by an individual weight and then summed along with the bias term. Then the sum is put through an activation function. As a set of equations where x_n is the input, w_n is the weight corresponding to that input, b is the bias-term, X is the sum, f_a is the activation function and Y is the output of the neuron.

$$x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + \cdots + x_n * w_n + b = X \tag{2.1}$$

$$f_a(X) = Y \tag{2.2}$$

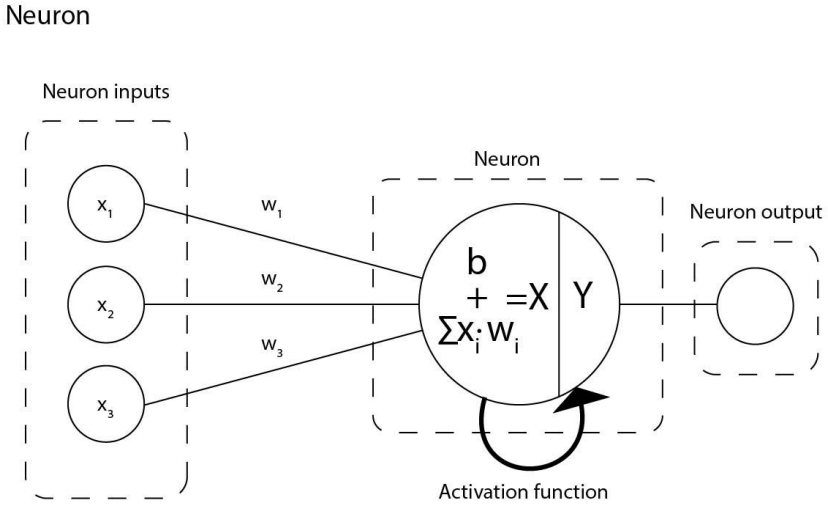


Figure 2.1. Overview of a neuron

2.2.1.3 Feed-forward networks

A feed-forward network organises neurons into layers. Each layer consists of a fixed number of neurons in a fixed configuration. Each layer takes its input as a tensor to be processed and the output is a tensor consisting of the arranged neuron activations[4]. This connects the outputs of the neurons of one layer as the input of the next layer neurons. The input of the first layer is the image to be classified (Figure 2.2). The input of the second layer is the output of the first layer and so it continues inside the network. The final layer is then taken as the output of the network (Goodfellow et al. 2016, 164). The neurons in this layer are often referred to as logits.

[4] The reader might want to visit the following site for a clearer explanation of how feed-forward network process information: http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

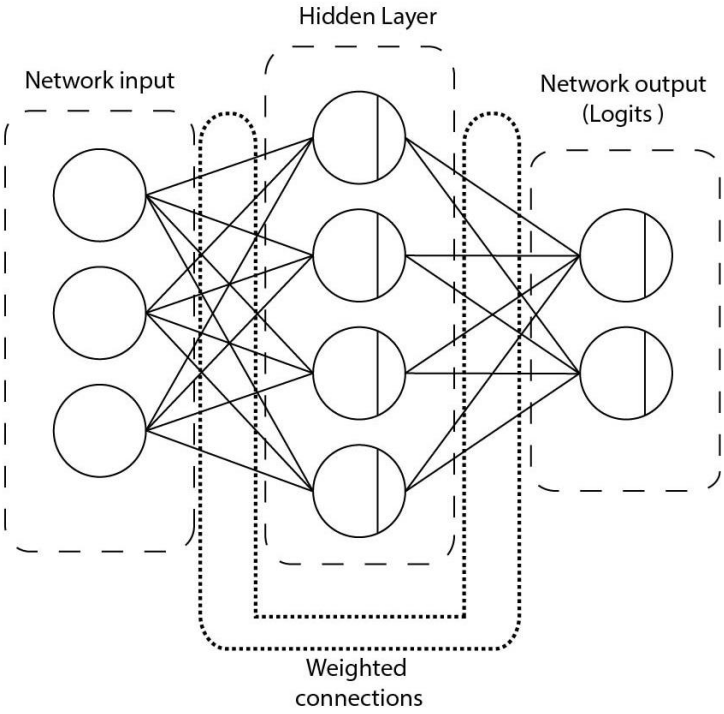


Figure 2.2 Overview of neural network and its basic components

Interpretation of the last layer is often done as follows in classification tasks. Each neuron in the output-layer is seen as representing the score for a given class. When an image is fed to the network the output neuron(logit) with the highest score is set to represent the prediction of the network (Figure 2.3) (Goodfellow et al. 2016, 180).

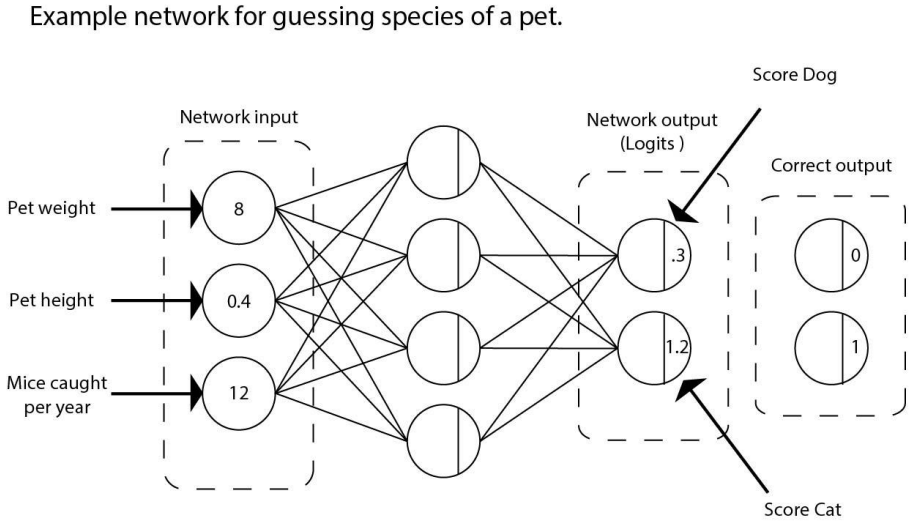


Figure 2.3 Example network

2.2.1.4 Supervised training

Supervised training of the network is conducted using input for which the correct output (label) is known.

The correct output values are often converted as what is known as one-hot encoding. This means that the label for the image, which is often a number designating the class, is converted into a vector with a big activation score for the correct class and no activation score for the incorrect classes (Goodfellow et al. 2016, 141).

After the network has been fed an image the error is computed. This is done by matching the logits against the desired, correct, logit values (Goodfellow et al. 2016, 165). The error for each logit is then calculated (Figure 2.4).

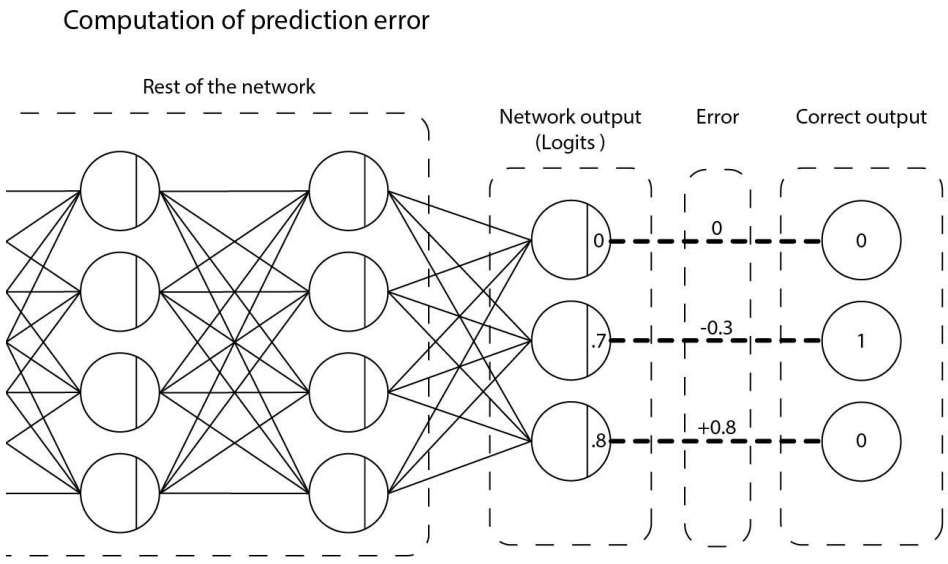


Figure 2.4 Computing errors for each logit

These errors are compiled and are used to optimise the network, when the network learns, it really is minimizing this compiled error (Goodfellow et al. 2016, 274). The weights and bias for each logit neuron is adjusted just slightly depending on the size of the error. The same process is then repeated for the layer below, and so on, until the whole network has been slightly adjusted and, hopefully, improved. The network error is a performance measure for the task and can be compiled for different datasets where the correct output is known.

The process of once running all the images of the training dataset through the network and adjusting the weights is referred to as an epoch. It often takes many epochs for a network to improve to the desired point.

2.2.1.5 Activation functions and dead neurons

As mentioned earlier each neuron has an activation function that transforms the sum of the weighted inputs and the bias into the output. They can be described as one form or another of $y = f(x)$.

The simplest activation function is the linear activation function which is simply $y = f(x) = x$ (Goodfellow et al. 2016, 178) (Figure 2.5 , top).

The most common one in use is the Rectified Linear Unit or ReLU function. This function always returns zero for values under zero and returns the input otherwise, this means $y = \max(0, x)$ (Goodfellow et al. 2016, 189) (Figure 2.5 , middle). However due to how weights in are updated during supervised training, neurons

using this activation function will not update when the output is zero. If at any point a ReLU activated neuron should stop producing positive output it will stop updating as well, never to recover. This is referred to as the neurons "dying". It is the activation function used in the VGG-16 network and therefore it will also be used in this thesis (Simonyan, Zisserman 2014).

To combat the problem of dying ReLU neurons there exists a variant of it called the leaky ReLU. This activations function behaves the same as the ReLU for values equal to or larger than zero and for values smaller than zero it has a much smaller derivative. This means that the weights can be adjusted for negative values while still having a similar shape to the ReLU.

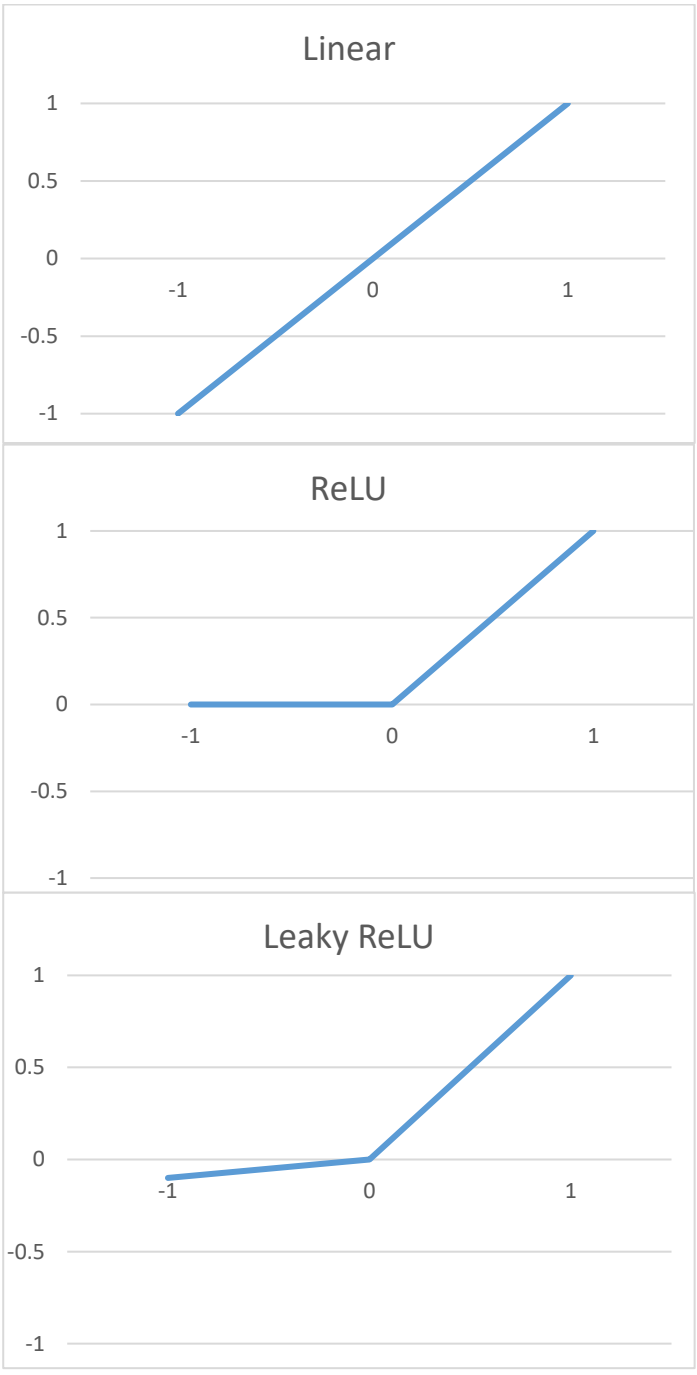


Figure 2.5 Activation functions. Linear(top), Rectified Linear Unit (middle), and Leaky ReLU(bottom).

2.2.1.6 Fully connected layers

The fully connected layer is the simplest kind of layer. Each neuron takes the full input tensor to the layer as its input (Just as in Figure 2.3) and the all the neuron outputs are arranged into the layer output (Goodfellow et al. 2016, 170). This means that if the input tensor contains 4000 values each neuron must have 4000 individual weights. If the fully connected layer contains 2000 neurons that means that a total of $4000 \cdot 2000 = 8000000$ weights will be adjusted as the network trains. This might be feasible for a few layers and for smaller input tensors but for large tensors, such as a 224x224x3 image, other solutions may be necessary.

2.2.1.7 Convolutional layers

Convolutional layers are based on having many copies of the same neuron each processing a subset of the input tensor for the layer (Goodfellow et al. 2016, 330). The idea is that we are often looking for similar patterns all across the input tensor (for example detecting edges in an image) and thus it is more effective to reuse the same neuron rather than having hundreds of copies that might end up training towards very similar goals (Goodfellow et al. 2016, 330)[5]. Applying the same neuron to an input tensor can be likened to applying a filter to it.

Each of the subsets are of identical size and divide the input tensor according to some dimensions. For example, if we have a small 8 by 8 RGB image, in other words an 8x8x3 tensor (Figure 2.6, top), we might select the subsets along different dimensions. We might want each copy of the neuron to take a separate row of pixels as its input, in other words each neuron would have an 8x1x3 subset tensor as its input (Figure 2.6, middle). This would mean we would have a one-layer output value per row of the image, creating an 8-value vertical vector. Or we might want each copy to take a separate colour channel as its input, in other words an 8x8x1 subset tensor as its input (Figure 2.6, bottom). This would allow us to search each of the colour channels for a similar pattern. Both of these examples are one-dimensional convolutions as they slice the layer input tensor along one axis.

[5] The reader might want to visit the following site for a clearer explanation: http://colah.github.io/posts/2014-07-Conv-Nets-Modular/

Examples of subsets for convolutions along one dimension.

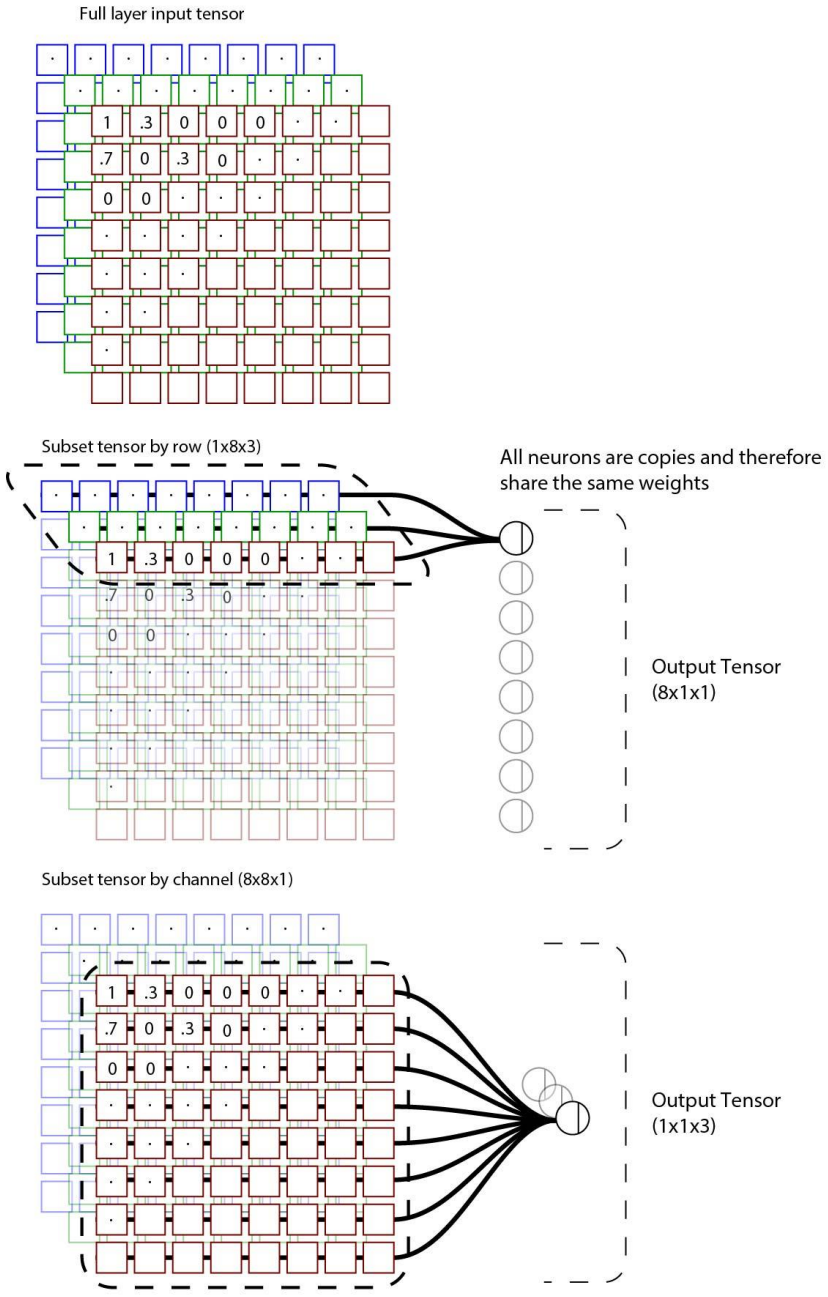


Figure 2.6 Selecting subsets for convolution along one dimension.

It is also possible to create two-dimensional convolutions by slicing the layer input tensor along two dimensions. In this network (and many other image related networks) the subsets are selected as squares from the layer input tensor (Goodfellow et al. 2016, 328)For example we might take 2x2x3 subset tensors as the input (Figure 2.7, top)As mention earlier convolutional layers utilise copies of the same neuron but it is also possible to have sets of copies of different types of neurons to look for more than one pattern in the input tensor. Each of types are fed the same subset tensors but conduct different calculations and their activations are compiled into a common layer output tensor (Figure 2.7, bottom). This can be likened to looking for several different patterns in the same input data and is very useful. In the output tensor of a convolutional layer the last dimension signifies the number of different neurons in the layer, for example an output tensor of 16x16x4 means that there are four different types of neurons, each existing as 16x16 copies.

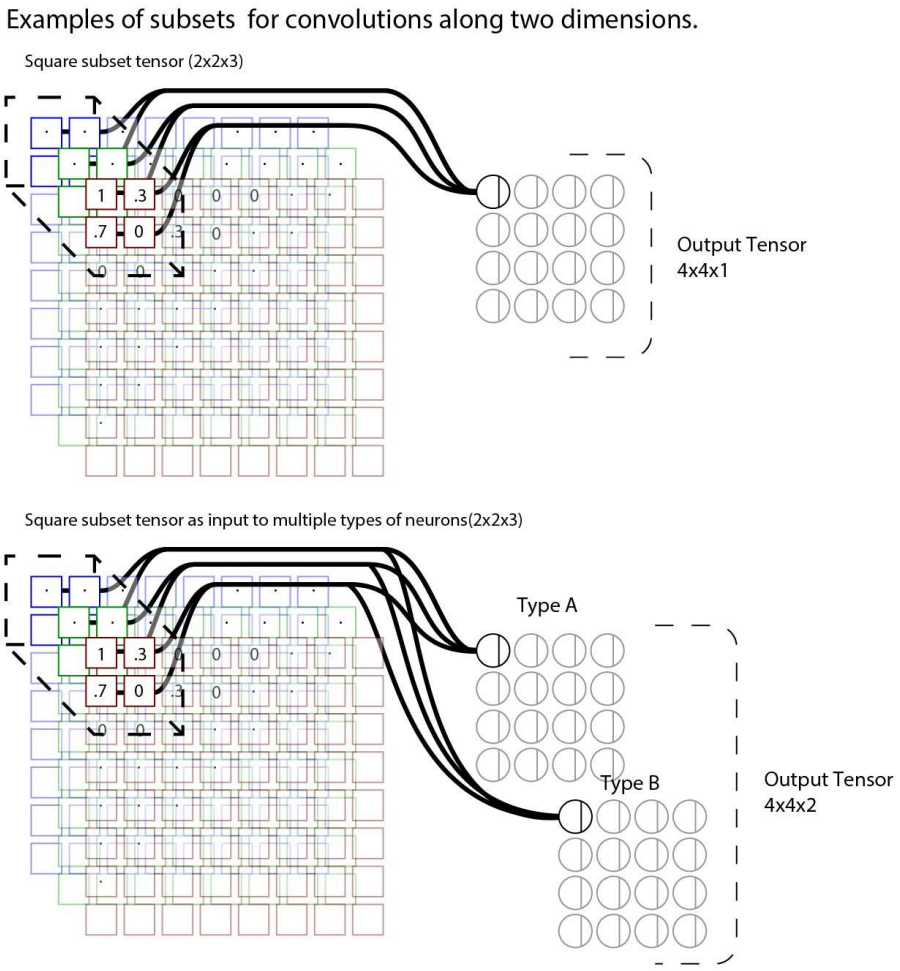


Figure 2.7 Selecting subsets for convolution along two dimensions.

All the examples this far have been non-overlapping but there is nothing which limits the different subsets of the input tensor from overlapping (Goodfellow et al. 2016, 343). Whether there is an overlap or not depends on how far each subset of the layer input tensor is shifted from its neighbours (Figure 2.8). If the subsets are of size 3x3 and are only shifted one cell from each other there will be an overlap of two cell. The amount that each subset is shifted from its neighbours is called the stride.

Stride can be applied in all dimensions. The previous example with subsets by rows had no movement through the second and third dimensions and a stride length of one in the first dimension, this can be summarised as a stride of (1,0,0). The previous example with subsets by colour channel had no movement through the first and second dimensions and a stride length of one in the third dimension, this can be

summarised as a stride of (0,0,1). The previous example with square subsets had no movement through the third dimension and a stride length of one in the first and second dimensions, this can be summarised as a stride of (1,1,0).

If all values of the layer input tensor are to be treated somewhat similarly there may be a problem. For example, for a two-dimensional layer input tensor using 3x3 subsets it might be desired to have all values be in the centre of the subset once. How can this be done for the values at the edges and corners of the layer input tensor? This problem is handled by "padding" the layer input tensor where necessary with dummy numbers (Figure 2.8)(Goodfellow et al. 2016, 343). In this study all padding will be done using zeroes. Padding with a stride length of (1, 1, x) has the benefit of enabling the layer output tensor to be equal in shape to the input vector along the convolutional dimensions (the first two in this case) (Goodfellow et al. 2016, 345).

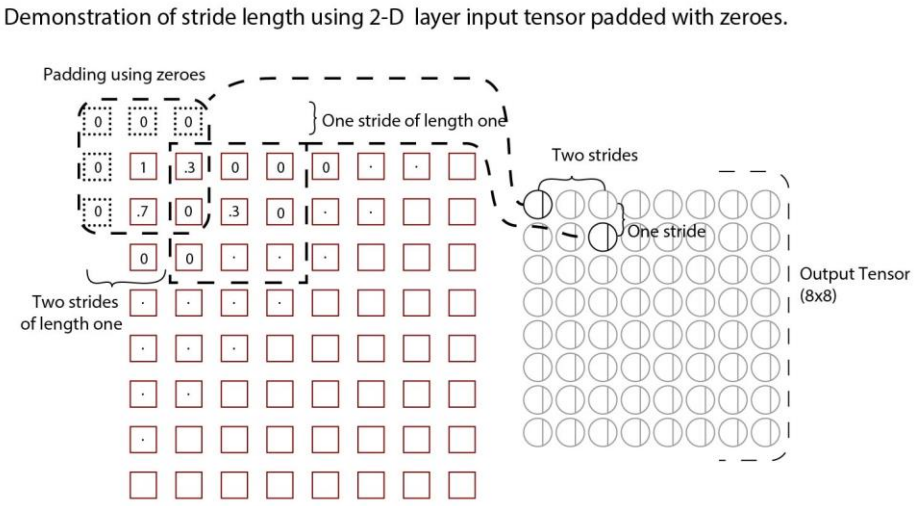


Figure 2.8 Example of stride length and padding

When using convolutional layers, it is easy for the tensor sizes to grow quite a bit. Consider an input RGB image as a 224x224x3 tensor which is run through 12 different types of neurons creating a 224x224x12 tensor.

2.2.1.8 Max-pool layers

A max-pool layer divides its layer input tensor into subsets and for each one it selects the maximum value to add to the layer output tensor (Goodfellow et al 2016, 335). This allows us to downsize large tensors and maintaining the most extreme values.

Max-pooling is often used on subsets sliced in 3 dimensions. For example, in the VGG-16 architecture the slices are 2x2x1 with strides of (2,2,1).

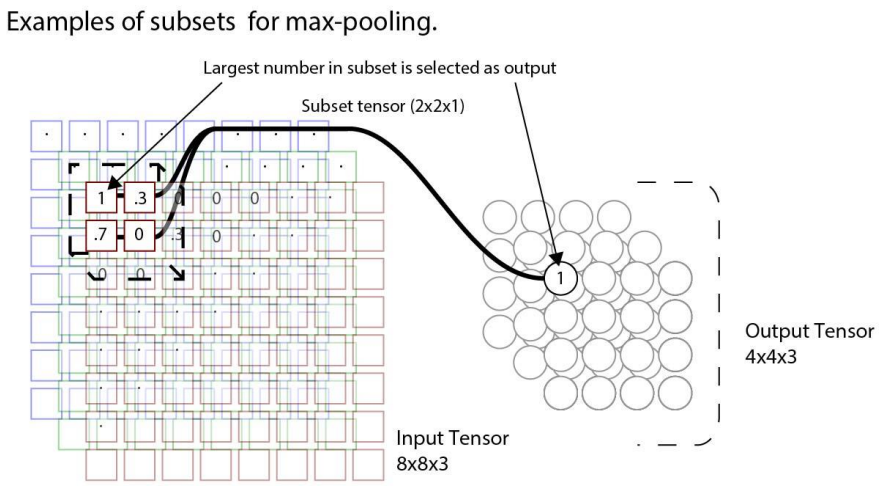


Figure 2.9 Example max-pooling

2.2.1.9 VGG-16 Architecture

The VGG-16 architecture was developed by Karen Simonyan and Andrew Zisserman for image classification. It features a total of 16 layers.

Input images are 224x224x3 and normalized for the mean pixel colouration of the network it was trained on, in this case the ILSVRC2012 dataset (ImageNet Large Scale Visual Recognition Challenge 2012). The pixel values are given as floating-point values in the range 0-255.

All VGG-16 neurons use the Re-LU activation function.

All VGG-16 convolutional layers use 3x3xN subsets where N is the full length of the third tensor dimension. All VGG-16 convolutional layers use a stride of (1,1,0). All VGG-16 convolutional layers use padding with zeroes.

All VGG-16 max-pool layers use 2x2x1 subsets and a stride of (2,2,1).

The full VGG-16 architecture will now be summarised as a table.

Table 2.1 The VGG-16 Architecture.

Layer name	*Input Tensor*	*Output Tensor*
Convolutional 1	224x224x3	224x224x64
Convolutional 2	224x224x64	224x224x64
Max-pool	224x224x64	112x112x64
Convolutional 3	112x112x64	112x112x128
Convolutional 4	112x112x128	112x112x128
Max-pool	112x112x128	56x56x128
Convolutional 5	56x56x128	56x56x256
Convolutional 6	56x56x256	56x56x256
Convolutional 7	56x56x256	56x56x256
Max-pool	56x56x256	28x28x256
Convolutional 8	28x28x256	28x28x512
Convolutional 9	28x28x512	28x28x512
Convolutional 10	28x28x512	28x28x512
Max-pool	28x28x512	14x14x512
Convolutional 11	14x14x512	14x14x512
Convolutional 12	14x14x512	14x14x512
Convolutional 13	14x14x512	14x14x512
Max-pool	14x14x512	7x7x512
Fully-connected 1	7x7x512	1x1x4096
Fully-connected 2	1x1x4096	1x1x4096
Fully-connected 3 (Logits)	1x1x4096	1x1x1000

Please note that the network was slightly modified to have an output layer of 102 neurons, one for each species of flower.

This network architecture was used due to it being well studied, due to the availability of pre-trained weights and because it features a mix of convolutional and fully-connected layers which is used in many other image-classifying networks. For example, Krizhevsky et.al (2012).

2.2.2 **Train, Test and validation data**

Previous mentions of input data and labels have been with regards to teaching and optimizing the network but there are other uses for the dataset.

The first is to be able to evaluate the dataset after training. A useful trained network should be able to perform its task well for images it has not yet seen (Goodfellow et al. 2016, 102). Otherwise it can only predict the classes of images that we already know the label for. To do this a subset of the dataset is separated before training and not utilized for training examples. This is the separation between the training dataset and the test dataset.

The second is to determine when it is time to stop training the network. One might think that the more training the better, but this is not the case (Goodfellow et al. 2016, 108). In the beginning the network is under specialised, it has not trained enough to perform the task, neither for the training dataset nor a general dataset (This is known as underfitting). In the middle the network is learning general descriptions which apply both for the training dataset and the general dataset. Eventually the network will learn more and more precise description from the training dataset in favour of deprioritising the more general descriptions (This is known as overfitting). For example, it might eventually just learn the correct exact pixel values for the top row, forgetting the more general analysis learned first.

To combat this issue the performance on a general dataset is approximated continuously during training. After going through the training dataset once and adjusting the weights the algorithm is asked to provide predictions (without adjusting the weights afterwards) for a separate validation dataset (Goodfellow et al. 2016, 118). The error is computed for the validation dataset separately too and is used to evaluate when the network is still underfitted (and learning more general descriptions) and when it is beginning to overfit (becoming to specialised for the training dataset). By using this measure, we can adapt our network architecture and decide when to stop training. Often the error for the training dataset decreases along with the validation dataset for the first iterations and eventually the validation error begins to increase while the trained error keeps on decreasing.

The reason to not use the test data for this is to maintain the test dataset as representing completely new data. Using the test data for validation can be likened to practicing for a test in which you know the questions.

2.2.3 **Data preparation**

To prepare the images for the network the images and labels need to be pre-processed and separated into train/test/validation datasets.

First the images are cropped to be square by removing rows evenly from the bottom and top. Then they were downsized to be 224x224 pixels. As the pretrained VGG-16 network is trained on images normalised for the mean RGB values of the ILSVRC2012 (ImageNet Large Scale Visual Recognition Challenge 2012) dataset the same normalisation is applied to the flower images (Simonyan, Zisserman 2014). This is done by subtracting the mean pixel value of ILSVRC2012 for red,

green and blue respectively. These are (123.68, 116.78, 103.94) for (r, g, b) respectively.

The data is separated into a training dataset, a validation dataset and a test dataset. For each dataset each image and its corresponding label are kept in separate vector sorted in the same order. As the number of images per class of flowers varies from 40 to 258 images the number of images of each class is limited in the training dataset to avoid skewing the learning. If some classes are overrepresented the algorithm can improve its accuracy by simply predicting those. The images were split per class with one fifth (rounded down) put into the testing dataset, one fifth (rounded down) put into the validation dataset. If there were forty or fewer images remaining afterwards all of them were put into the training dataset, if there were more only the first 40 were put into the training dataset.

This led to the test and validation dataset both consisting of 1602 images and labels respectively and the training dataset consisting of 3612 images and labels.

The labels were one-hot encoded prior to training as described earlier.

2.2.4 **Transfer learning**

Training a network can require a lot of time and GPU processing power and if the training dataset is small overfitting is a big risk (Sharif Razavian et al. 2014). To avoid this a method called transfer learning might be used.

Transfer learning entails copying weights from the first n layers of a trained network onto the first n layers of a target network that is to handle another task (Yosinski et al. 2014). The remaining layers of the target network are initialized randomly. When the network is then subjected to training on the dataset of the new task either all weights can be updated or only the weights on the randomly initialized layers.

The idea is that the pre-trained network in the process of learning to its specific task has learned how to extract more general features in its hidden layers (Yosinski et al. 2014). By utilising these pre-trained weights, it is not necessary to train the whole network, only the final layers. In summary, by using transfer learning both training time and the risk of overfitting can be reduced.

In this thesis all weights, except for those in the last layer, were initialized from a version of the VGG-16 network trained on the ILSVRC2012 dataset (ImageNet Large Scale Visual Recognition Challenge 2012)[6]. The weights for the last layer were initialised randomly with a truncated normal distribution.

[6] The network was downloaded from https://github.com/tensorflow/models/tree/master/research/slim

In the subsequent retraining the weights for all the convolutional layers were kept constant. This meant only the weights of the final 3 layers were being retrained.

2.2.5 **The final network**

The network was constructed using a TensorFlow wrapper called Interpret Tensor created by Lapuschkin et al. (2016). This wrapper was used due to it automatically implementing the LRP analysis described later.

The network first iteration had some errors in training due to neurons in the logit layer not functioning properly. By changing the activation function of the last layer to linear it was discovered that issue probably was that the logit neurons were dying. The last layer was instead set to use the leaky ReLU activation function. Upon implementing this change there were no further issues.

The network was trained for 20 epochs and through the validation dataset error it was determined to use the post-epoch 5 network as the final network for the thesis. For the test dataset, the network classified 87% of images correctly (1394 images) and 13% images incorrectly (208 images).

2.3 Layerwise Relevance Propagation

The goal of the relevance propagation is to for each pixel in the input image quantify its contribution to a specific classification. Relevance propagation in this thesis is done by the method detailed by Bach et al. (2015).

At the network level relevance propagation happens in the following fashion: An image is fed to the network which produces a prediction in form of a logit vector. A class is selected among the logits (often the one with the highest score as it is interesting to see what motivated the top prediction) and is assigned a relevance value. This relevance value is distributed among the neurons in the previous layer which in turn distribute their relevance and so on until the image input layer. The relevance values in the input layer is the quantification of the contribution for each pixel towards the examined class. Positive relevance means that the pixel had a positive contribution to the prediction of the examined class. Negative relevance means that the pixel had a negative contribution to the prediction of the examined class. Relevance closer to zero signify that the pixel did not argue neither for nor against the examined class.

The paper by Bach et al. (2015, fig. 25) (Figure 2.10) provides some very compelling demonstrations of intuitive visualisations. They show examples of applying a deep neural network on classifying images of various objects and animals (e.g. cat, chicken, cup, spider) and the algorithm clearly and precisely discerns the

areas where they are located and their contours as more relevant compared to the background. They also show some very intuitive results when examining the MNIST handwritten numbers dataset (Bach et al. 2015, fig. 12) (Figure 2.11). In these images it is very clear what parts of the image have positive and negative relevance for different number classifications.

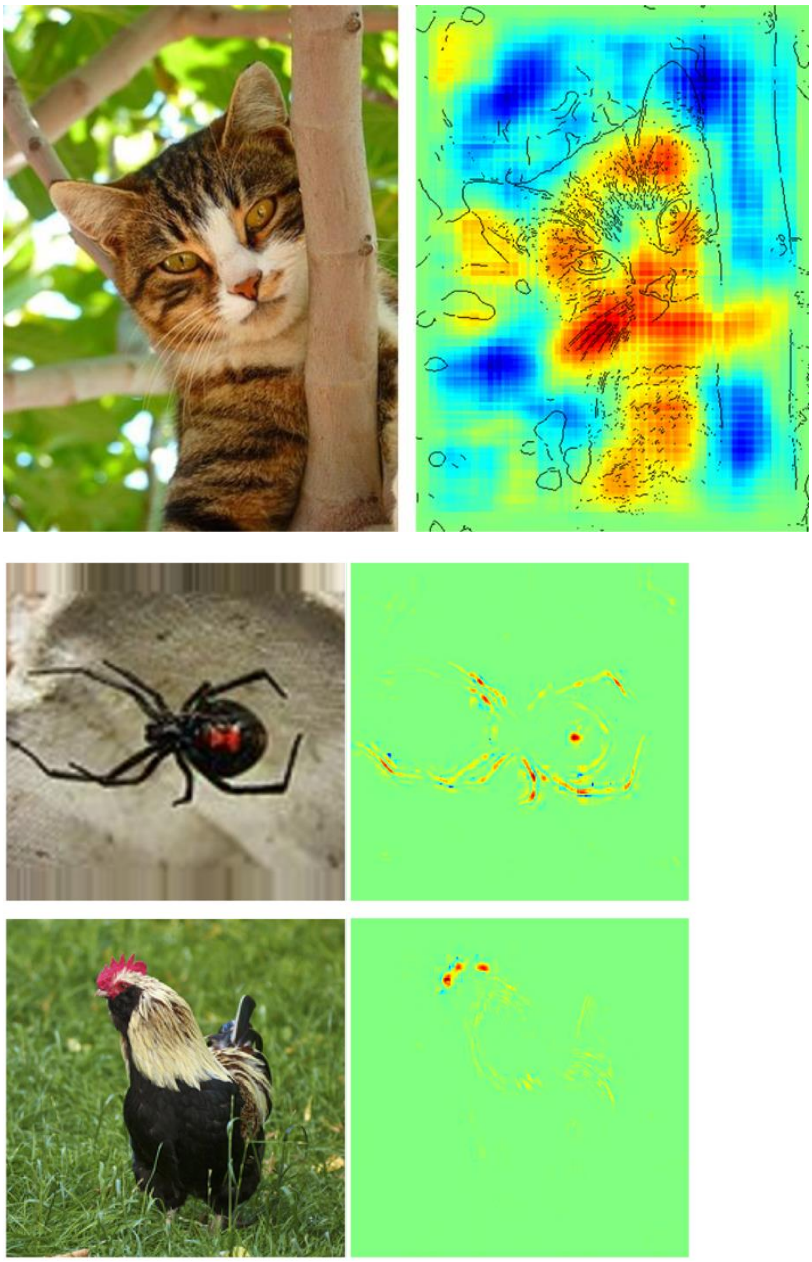


Figure 2.10 Images from Bach et.al using the ILSVRC2012 dataset

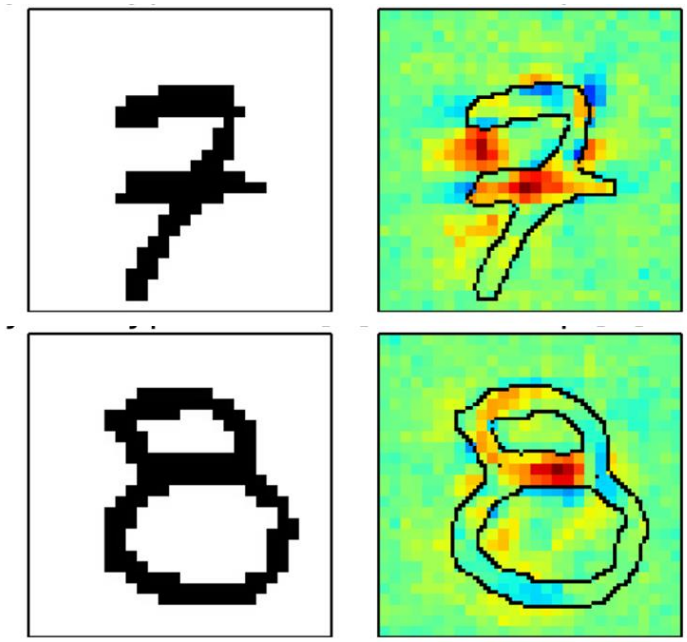


Figure 2.11 Images from Bach et.al using the MNIST handwritten numbers dataset

2.3.1 **Propagating relevance**

The process of propagating relevance is simple on the neuron-level and consists of distributing the relevance assigned to a neuron among all its input neurons. For each input the input value is multiplied its weight to compute the "message" contributed to the sum (Bach et al. 2015). Each message is then assigned positive relevance if it has the same sign as the pre-activation sum (e.g. positive message into positive pre-activation sum) and negative input if it has the opposite sign. The relevance is proportional to the size of the message.

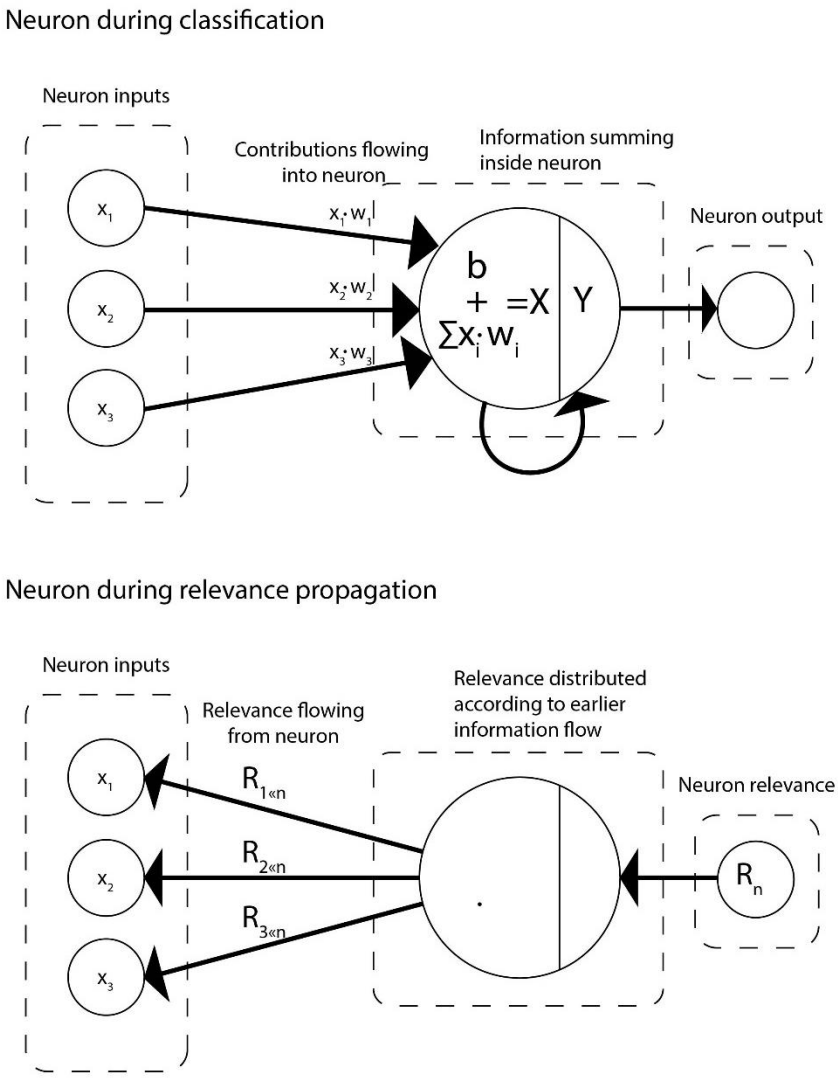


Figure 2.12 Neuron behaviour during classification and relevance propagation

As a set of equations reflecting the diagram in Figure 2.12 where R denotes relevance, n denotes the neuron which relevance is distributed from, Psum denotes the pre-activation sum, *a* denotes the input activations which the relevance is distributed among, 1←n denotes relevance distributed from the neuron to one of the inputs and w denotes the weights:

$$R_n = R_{1\leftarrow n} + R_{2\leftarrow n} + R_{3\leftarrow n} \tag{2.3}$$

$$a_1 * w_1 + a_2 * w_2 + a_3 * w_3 = Psum \tag{2.4}$$

$$R_{1\leftarrow n} = R_n * \frac{a_1 * w_1}{Psum} \tag{2.5}$$

The first relevance is inserted in the logit neuron for the class to be examined ensuring that the LRP result only pertains to that class. This method does not consider the bias term.

2.3.2 **Epsilon term**

When propagating relevance according to the above equations the transmitted relevance can take on unbounded values if the Psum is close to zero (Bach et al. 2015, 21). To combat this a term stabilizing term $0 \leq \varepsilon$ is introduced to equation 2.5. This gives the following set of equations where R denotes relevance, n denotes the neuron which relevance is distributed from, Psum denotes the pre-activation sum, *a* designates the input activations which the relevance is distributed among, 1←n denotes relevance distributed from the neuron to one of the inputs and w denotes the weights:

$$R_{1\leftarrow n} = \begin{cases} R_n * \frac{a_1 * w_1}{Psum + \varepsilon} & Psum \geq 0 \\ R_n * \frac{a_1 * w_1}{Psum - \varepsilon} & Psum < 0 \end{cases} \tag{2.6}$$

Aside from avoiding unbounded values these equations also help produce less grainy heatmaps upon visualisation. See

2.4 Visualisations

2.4.1 **Visualisation goals**

In order to structure the work of creating the visualisations a few goals are defined by this thesis. In addition to bringing some structure to the development these also serve to amplify any positive effects on discerning correct and incorrect network classifications. Furthermore, they provide clear objectives for further research to optimise or dismiss.

The first and most important goal is that the visualisations convey a feeling of reliability or unreliability depending on the quality of the estimation. Preferably all incorrectly classified images should convey that unreliability both analytically and intuitively. Many of the following visualisations goals are designed with the objective of helping this goal.

The visualisations should clearly relate the original image to the LRP-analysis. This can help identifying what in the original image was triggering the network, such as shapes and areas. More generally it is the link between the human intuitive understanding of the image and the algorithm analysis.

The visualisations should be clear. It should be easy to distinguish areas of large relevance (be it positive nor negative) from areas with relevance close to zero.

The visualisations should illustrate the balance between total positive and negative relevance in the analysis.

The visualisations should be proportional. A point with double the relevance of another point should feel doubly relevant that other point.

The visualisations should be precise, reflecting precisely the relevance of each individual pixel.

The visualisations should be mathematically distinct from one another. This may help understand what principles and mechanisms the test participants build their understanding and preferences on.

2.4.2 **Visualisation development**

There were three methods of visualisation created in this project.

Before the LRP-analysis for each image was visualised it was averaged and then normalised by squashing the data. This meant that the RGB values were first averaged for each pixel and then multiplied by a factor C such that $R_{max} \leq 1$ and $R_{min} \geq -1$ and either $R_{max} = 1$ or $R_{min} = -1$. While this distorts the data somewhat (maybe some images just have more relevance than others) it strives to make for clearer visualisations for all images. This normalisation is not due to any specific precedent.

All colours here are presented with RGB channels as (red-value, green-value, blue-value) and go from black (0, 0, 0) to white (1, 1, 1).

All four visualisations were presented in the same way, as to only compare the visualisations and not the presentation. They were presented in sets of three for each image classified by the network (Figure 2.13). On the left was the RGB original to be classified (without normalisation). On the right was the heatmap. In the middle was a hybrid between the original and the heatmap. The hybrid was a simply done by weighing the colour values of the original at 35% and the heatmap at 65%.

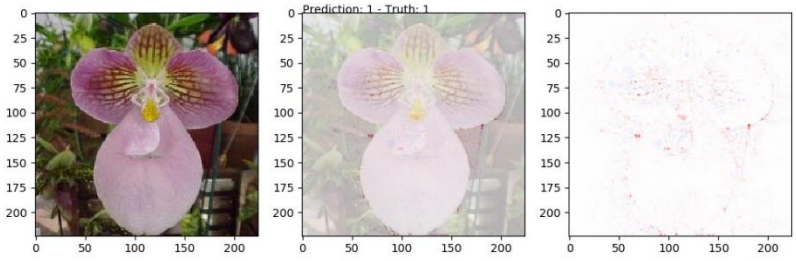


Figure 2.13 Presentation layout

This hybrid was done to help relate the original image and the LRP-analysis to each other and the result felt a lot better than just having the heatmap and the original side by side.

The visualisations were developed in the order they are presented below.

2.4.2.1 The "linear" heatmap

The first visualisation is based around a mainly linear relationship between the relevance of a pixel and its colouration. Most of the basic development was done using this visualisation.

The function for creating the heatmap uses three colours; one greyscale colour(zero-colour), one to signify positive relevance (full red (1, 0, 0)) and one to signify negative relevance (full blue (0, 0 ,1)). The idea was to have intensity of colour as a signifier of intensity of relevance, be it positive or negative. As for the selection of red and blue the reasoning was twofold. The first reason was to have very dissimilar colours, basically orthogonal in appearance instead of having two colours which are more similar, such as yellow and red. Another combination might have been red and green, but this can be ruled out due to issues with colour blindness.

If the relevance for a pixel is 0 it is coloured with the zero-colour. If the relevance for a pixel is +1 it is coloured with red. If the relevance for a pixel is between 0 and +1 it is coloured by interpolation between zero-colour and red. Conversely, if the relevance for a pixel is -1 it is coloured with blue. If the relevance for a pixel is between 0 and -1 it is coloured by interpolation between zero-colour and blue.

As a set of equations where R is the relevance of the pixel, the zero colour is set to (0.5, 0.5, 0.5), and the heatmap colouration of the pixel is (r, g, b):

$$R \geq 0: \qquad R * (1\,,0\,,0) + (1-R)(0.5\,,0.5\,,0.5) = (r\,,g\,,b) \tag{2.7}$$

$$R < 0: \qquad -R * (0\,,0\,,1) + (1+R)(0.5\,,0.5\,,0.5) = (r\,,g\,,b) \tag{2.8}$$

The zero colour was initially set to white (Figure 2.14). It was found this bleached the hybrid image too much and it was also difficult to make out the areas of high and low relevance against the crisp white.

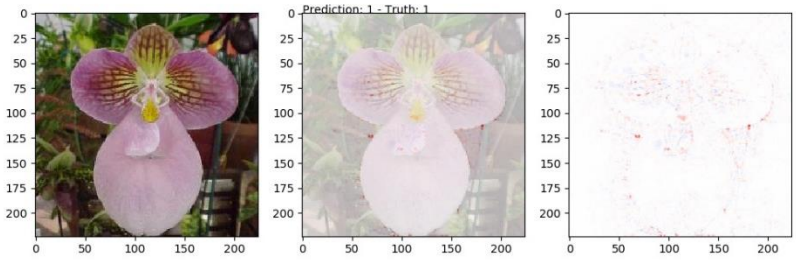


Figure 2.14 Hybrid (centre) using white as the zero-colour

The zero colour was then set to black (Figure 2.15). It was found this darkened the hybrid image too much and it was still difficult to make out the areas of high and low relevance against the black.

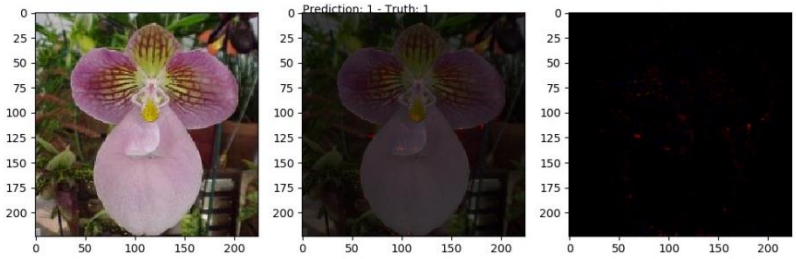


Figure 2.15 Hybrid (centre) using black as the zero-colour

In the end the zero colour was set to grey (0.7, 0.7, 0.7), this made the colours stand out better both in the heatmap by its own but especially in the hybrid images (Figure 2.16). This is probably due to the zero-colour being closer to the average brightness of the original.

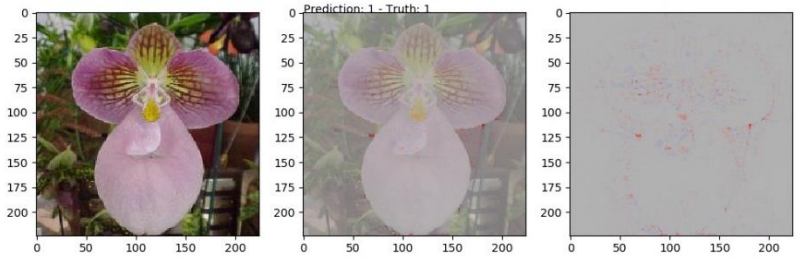


Figure 2.16 Hybrid (middle) using grey as the zero-colour

In the beginning the hybrid consisted of the RGB original but it was found that the red and blue of the heatmap were hard to distinguish with a coloured background. When the hybrid instead was done with a greyscale original and the heatmap the results felt much clearer (Figure 2.17). This is a further advantage of having intensity of colour signifying relevance as the heatmap is more clearly discernible from the greyscale background, especially where relevance is very positive or negative.

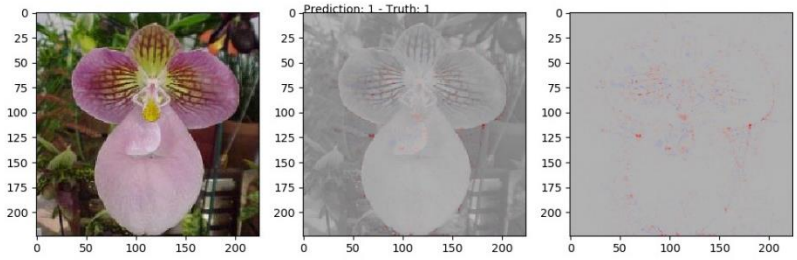


Figure 2.17 Hybrid(middle) using greyscale version of the original image

It was still hard to clearly distinguish the coloured areas, especially when a small number of pixels had much larger relevance than most of the others, then whose few pixels were coloured strongly, and the rest were coloured very weakly. This can be illustrated by creating a histogram (Figure 2.18, top) for all the pixels where each column is coloured as those pixels would have been in the heatmap. It also shows the colouring along the bottom.

To combat this issue of weak colouring, an amplification factor was introduced for the colours that signify positive and negative relevance. First, they were multiplied

by 2 and later by 3 for the testing. The difference is in the number of pixels that were previously grey that now have a distinguishable red or blue tint (Figure 2.18 and Figure 2.19).

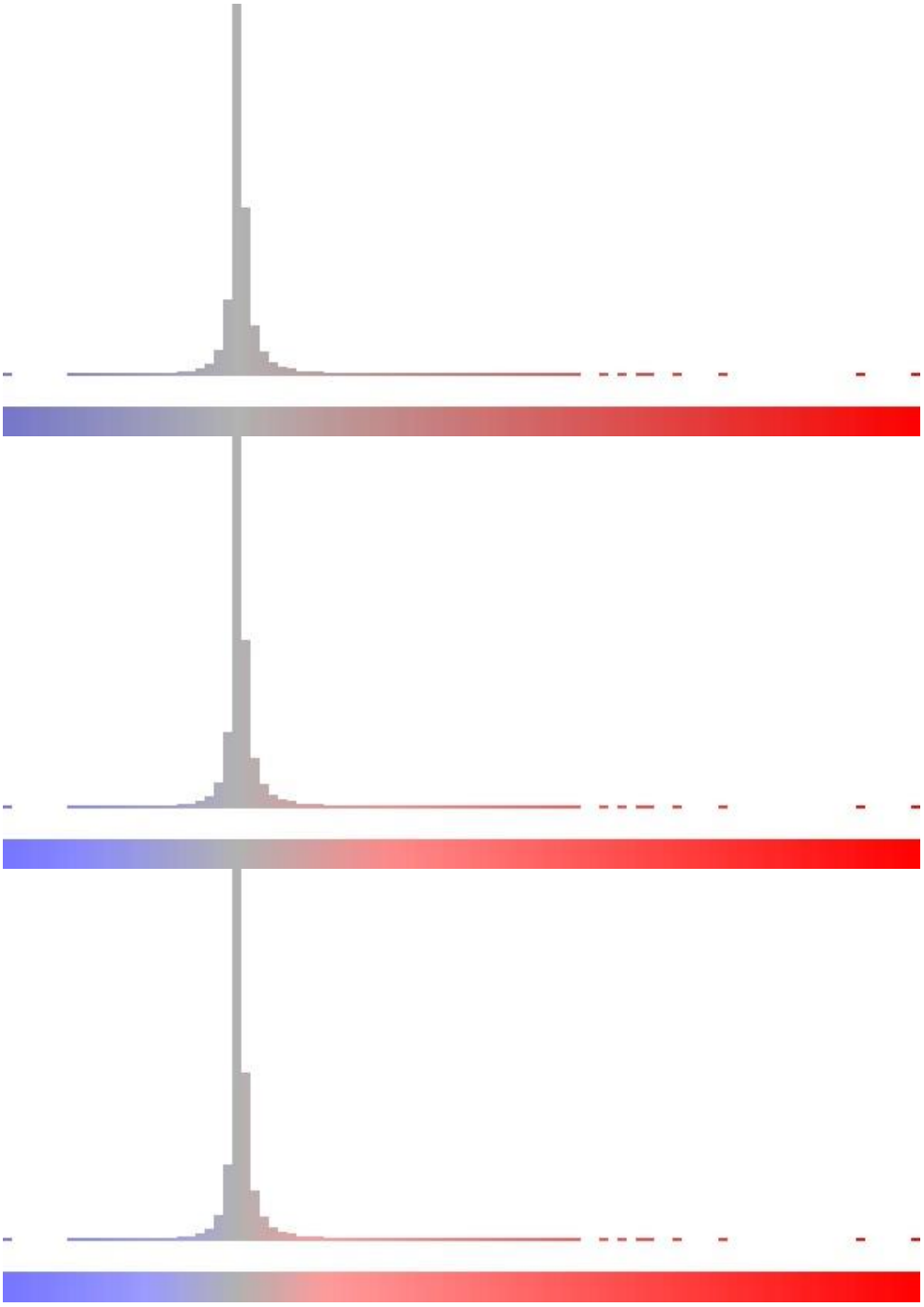


Figure 2.18 Variants on amplification factor. The topmost variant is an amplification factor of 1, the middle one is 2 and the third one is 3. Notice how the values around zero(grey) turn more pinkish and blueish.

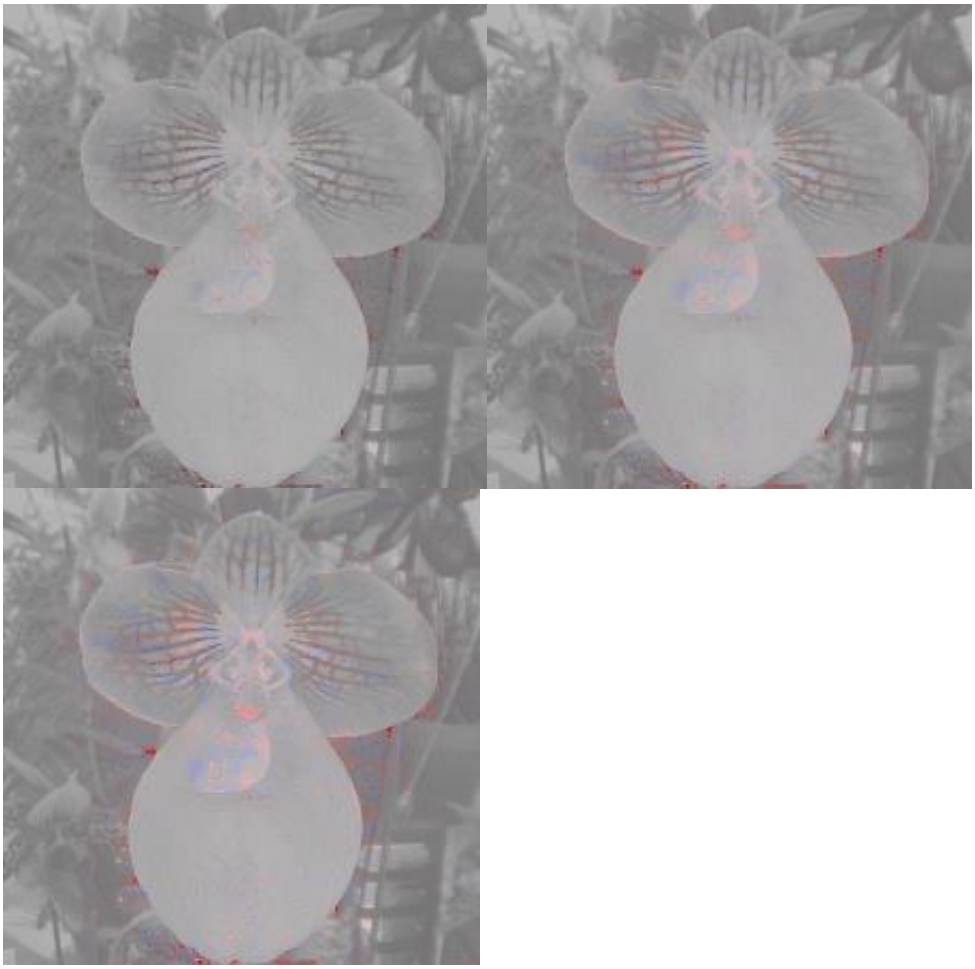


Figure 2.19 Variants on amplification factor. The top left variant is an amplification factor of 1, the top right is 2 and the bottom one is 3. Notice how the values around zero(grey) turn more pinkish and blueish.

As a set of equations where R is the relevance of the pixel, A is the amplification factor, the zero colour is set to (0.7, 0.7, 0.7), and the heatmap colouration of the pixel is (r, g, b):

$$R \geq 0: \qquad A * R * (1\,,0\,,0) + (1-R)(0.7\,,0.7\,,0.7) = (r\,,g\,,b) \tag{2.9}$$

$$R < 0: \qquad A * -R * (0\,,0\,,1) + (1+R)(0.7\,,0.7\,,0.7) = (r\,,g\,,b) \tag{2.10}$$

This visualisation was created to be the simplest representation of the LRP analysis, maintaining proportionality and precision well.

As described here this visualisation initially had issues with the goal of clarity as the colouring was very weak. This has been somewhat rectified by amplifying the signifying colour but for some images the colours are still weak.

2.4.2.2 The "histogram" heatmap

The second visualisation considers filtering the information by comparing only the pixels with large positive and negative relevance. The function for creating the heatmap uses five colours; one greyscale colour (zero-colour (0.7 , 0.7 , 0.7)), one to signify larger positive relevance (full red (0.95 , 0 , 0)), one to signify smaller positive relevance(slightly red(0.8 , 0 , 0)), one to signify smaller negative relevance(slightly blue(0 , 0 , 0.8)), one to signify larger negative relevance(full blue (0 , 0 , 0.95)).

The colours are assigned depending on which percentile the pixel belongs to in terms of its relevance value. The bottom and top 3% are coloured slightly blue and red respectively while the bottom and top 1% of pixels are coloured full blue and full red respectively. Any pixels in between are coloured with the zero-colour.

As a set of equations where I is where the pixel ranks among all the pixels when sorted by relevance from lowest to highest, N is the total number of pixels in the image and the heatmap colouration of the pixel is (r, g, b):

$$I < 0.01N: \quad (0\,,0\,,0.95) = (r\,,g\,,b) \tag{2.11}$$

$$0.01N \leq I < 0.03N: \quad (0\,,0\,,0.8) = (r\,,g\,,b) \tag{2.12}$$

$$0.03N \leq I < 0.97N: \quad (0.7\,,0.7\,,0.7) = (r\,,g\,,b) \tag{2.13}$$

$$0.97N \leq I < 0.99N: \quad (0.8\,,0\,,0) = (r\,,g\,,b) \tag{2.14}$$

$$0.99N \leq I: \quad (0.95\,,0\,,0) = (r\,,g\,,b) \tag{2.15}$$

This visualisation avoided the issue of the "linear" heatmap in that it did not matter if a few pixels had extreme values, this can be clearly illustrated by using the same coloured histogram graph as earlier, comparing it to the amplified linear model (Figure 2.20)

This visualisation was created with the goal of being clearer than the linear one by marking and contrasting areas of large positive and negative relevance to those of medium relevance. It is also very precise as it marks individual pixels clearly. It is not good for illustrating the balance between positive and negative relevance in an image as there will always be the same number of pixels in each of the percentile categories.

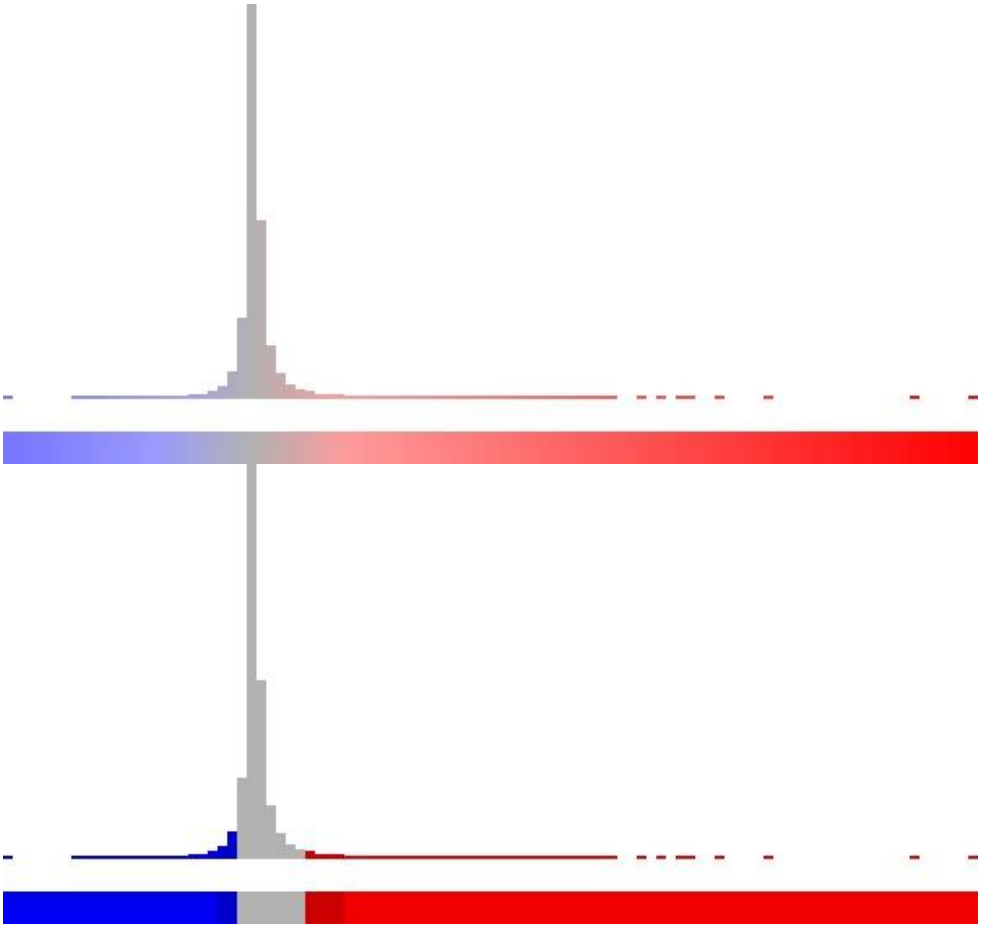


Figure 2.20 Histogram comparison between linear and histogram heatmap.

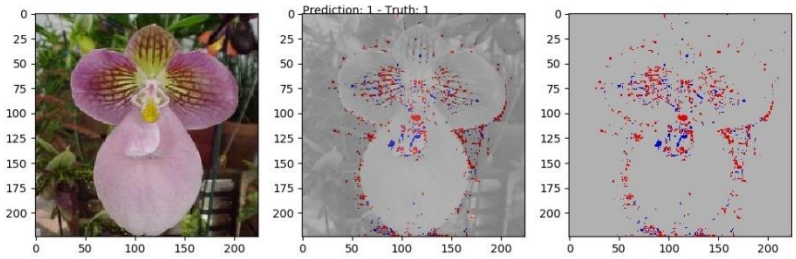


Figure 2.21 Histogram heatmap

2.4.2.3 The "blurred" heatmap

The third visualisation tries to provide a looser and intentionally less detailed heatmap. This aims to direct focus from the exact pixels of relevance and help more with summing areas of positive and negative relevance and a more intuitive analysis.

Each pixel contributes to the colouring of neighbouring pixels (receiving pixels) within a distance of 12. The contribution scales with the relevance of the contributing pixel and diminishes with distance. The process is done for all the pixels in the heatmap so that by the end each pixel has a colouring depending on its own relevance but also on the relevance of its neighbours. The colouring is accumulative, here meaning that all pixels in the heatmap start as almost black (0.1, 0.1, 0.1) then red (x, 0, 0) and blue (0, 0, x) are added bit by bit.

As a set of equations where R_c is the relevance of the pixel currently contributing, D is the distance between the contributing pixel and the receiving pixel, the heatmap colouration of the receiving pixel before the contribution is (r_0, g_0, b_0), and the heatmap colouration of the receiving pixel afterwards is (r_1, g_1, b_1):

$$R_c \geq 0: \qquad (r_0\,, g_0\,, b_0) + \frac{R_c * (1\,, 0\,, 0)}{D * 10 + 1} = (r_1\,, g_1\,, b_1) \tag{2.16}$$

$$R_c < 0: \qquad (r_0\,, g_0\,, b_0) + \frac{R_c * (0\,, 0\,, 1)}{D * 10 + 1} = (r_1\,, g_1\,, b_1) \tag{2.17}$$

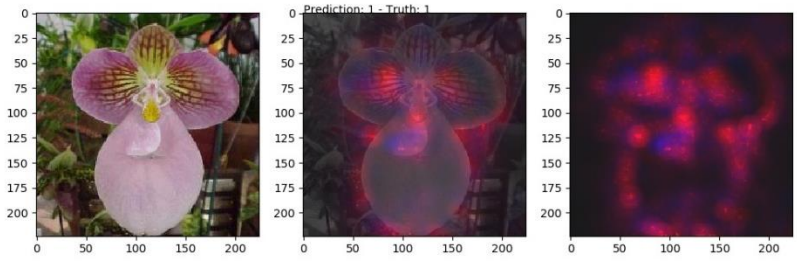


Figure 2.22 Blurred heatmap variant

2.4.3 **Changes after preliminary testing**

As the visualisations were being developed there was some preliminary testing conducted by showing the visualisations to persons not involved in the thesis.

In feedback during preliminary testing one person provided feedback on the presentation. The heatmap on the right did not contribute much and that she would prefer two hybrid images, one for positive relevance and one for negative. It was also believed that this would help to find possible separate patterns in positive and negative relevance hybrids respectively.

Because of this the presentation was changed (Figure 2.23). The left image was kept as the RGB original. The middle image was a hybrid between the positive relevance heatmap and the greyscale original. The right image was a hybrid between the negative relevance heatmap and the greyscale original.

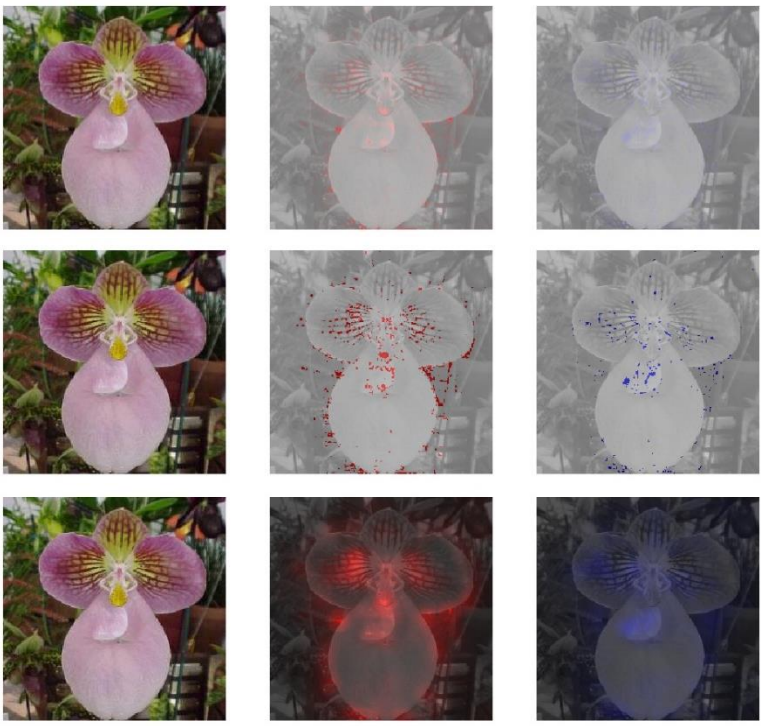


Figure 2.23 Second presentation layout. Linear visualisation (top row), Histogram visualisation (Middle row), and Blurred visualisation (Bottom row). Original images (Left column), hybrid image showing positive relevance (Middle column), hybrid image showing negative relevance (Right column)

2.5 Examples of visualisations

For more examples on how the final visualisations look, please see Appendix C

2.6 Test

While this study does not implement all the components Rubin and Chisnell (2008, p67) most are present. Objectives and research questions have been discussed in the Introduction chapter but participant characteristics, test design, environment, materials and data to be collected will be discussed under this heading. As this study is not a typical product usability test the task list and test moderator role will not be discussed in as great a detail.

2.6.1 **Test participants**

There were no specific requirements for set for test participants. As the task was considered fairly abstract, classifying English flowers, it was not expected that a significant proportion of test participants would have special knowledge to give misleading results. This means that the test design does not impose any special limitations on the selection of test participants.

As the thesis is a test of concept for whether these visualisations communicate any information and not an evaluation of the exact average size of this effect.

The balance to found in the strictness of selection of test participants can be said to be one of quantity of responses versus quality of responses. More responses will enable the study to more precisely measure any effect there might be, with higher statistical significance. The recommended minimum for statistical significance from Rubin & Chisnell (2008, p72) is 12 but the more the better. A better quality of responses mean that the selection of participants will be more representative of some relevant population. For a general study such as this one it might be the general population. The better the quality the more factors will be accounted for to ensure representativeness.

As an early study in the area this thesis is more concerned about the existence of an effect rather than the results being very representative of the general population. This means that quantity of responses will be heavily favoured. Because of this it was decided to not limit the recruitment of test participants in any way.

2.6.2 **Test design**

The test can be said to consist of five main parts; Introduction, learning examples, main test questions, visualisation evaluation and total evaluation.

Please note that the second test version can be found in Appendix B. There were some substantial changes between the first and the second test version and therefore some details of the first version are not described in detail here.

The introduction aims to build the basic understanding required to complete the rest of the test. Some are essential pieces of understanding and some are simply beneficial, giving the user some background to frame the essential understanding in. Here comes a listing of information and understanding the user must or preferably should have after the introduction.

Table 2.2 Information to be communicated to the user through the introduction

The test participant should preferably understand that an algorithm has been tasked with classifying images of flowers according to species.
The test participant must understand that the algorithm is somewhat prone to error.
The test participant should preferably understand that the errors occur when classifying images which the algorithm is supposed to be able to classify and not images of random flowers.
The test participant must understand that their role is to determine whether an image has been classified as the correct species or not.
The test participant must understand that the determination should be based in the visualisations.
The test participant should preferably understand that the visualisations show what parts of the image were relevant for the algorithm reaching its classification.
The test participant must understand that they should study the learning examples to deduce how they wish to differentiate correct classifications from incorrect ones.
The test participant should preferably understand that the different visualisations are variants based on the same analysis method.
The test participant must understand how to answer the main test questions.

The introduction begins with background questions regarding the test participants age, experience with programming and their attitude towards machine learning technology. The former is answered by number of years and both of the latter are implemented as rating scales.

The introduction continues with some background on the task of the algorithm followed by an explanation of the test participants task in the study. After this the visualisations are presented and explained along with an example question. All three types of visualisation are presented, each with the same correctly and incorrectly classified example to underline that they are variations of the same analysis. The two examples were, like all the images in the test, selected randomly to avoid the test construction skewing the results.

Please note that the visualisation methods are named in a more abstract way during the tests as to minimally influence the test participants. A is the histogram heatmap variant, B is the linear heatmap variant, and C is the blurred heatmap variant.

The next three sections; learning examples, test questions and evaluation of visualisation are repeated once for each visualisation method studied. The order is rotating to avoid distortion by learning effects (Rubin & Chisnell 2008, p75). The different orderings are [A→B→C], [C→A→B], and [B→C→A]. These are not all the possible permutations suggested by Rubin & Chisnell but since every visualisation method variant is tested about equally as the first, second or third there is still a balancing effect.

This is followed by the first set of learning examples of correctly and incorrectly classified images along with their visualisations. These are meant to provide practice and help the user construct some intuition or methodology for differentiating correctly and incorrectly classified images. There is a total of 8 learning examples per visualisation method, 4 correct and 4 incorrect and all clearly labelled. The same input images were used for each set of learning examples to give all the visualisation methods the same preconditions.

The next section contains main test questions as a set of images, each with the same question to answer with a Likert-style rating scale. Translated, the main test question reads "I would have confidence in this classification" [7]. The answer is provided as a rating scale from 1 to 4 where 1 is "No, absolutely not" and 4 is "Yes, absolutely". There is an even number of options to avoid having test participants selecting the neutral option whenever they are unsure. This could produce a dataset where due to the abstract nature of the task the test participants would prefer not to divulge small preferences. This is also thought to better represent real applications where there is often no middle ground, either the user must trust the algorithm classification or contradict it. Often, there is no maybe.

There is a total of 24 main test questions per visualisation method. While the ordering of the different visualisation methods rotates the images used are in the same order each time. This means that the first section always contains the same images in the same order, just visualised using different methods. The 24 main test questions per visualisation contain 16 correctly and 8 incorrectly classified images. This information is not shared with the test participants. The numbers are skewed from the real algorithm accuracy to provide a good sample size for both correctly and incorrectly classified images without making the test unnecessarily long. It is deemed that a true distribution of 87% and 13% (21 correctly and 3 incorrectly classified images) would have to small a sample size of user responses for images which are incorrectly classified.

This is followed by an evaluation of the visualisation method used in the prior two sections. All the questions mentioned here are design as Likert-type questions (A statement followed by a rating on a scale from "Strongly disagree" to "Strongly agree") unless specified otherwise. The first question is about the more general helpfulness of this visualisation method. The next question asks if the visualisation method helped in deducing what the algorithm considered positively relevant or negatively relevant. The third question asks if the visualisations helped in approaching the problem through intuition. The next two questions deal with whether the test participant felt different levels of confidence when trusting or

[7] In swedish: Jag skulle ha tilltro till denna bedömningen.

criticizing the algorithm. This aims to give further insight into the human-machine relationship for this task.

The final question asks for free text comments with a suggestion prompt regarding how the subjects used the visualisations.

The test ends with a simple comparative evaluation of the visualisation methods. The test participant is asked to select their favourite visualisation method and provide a free text comment on why that is.

2.6.3 **Updating the test design**

Before testing was fully launched a couple of preliminary tests were made with extra supervision and feedback. There was a total of three preliminary test participants and the thesis supervisor also assisted in refining the test. As mentioned earlier the final version of the test can be found in Appendix B.

The two biggest issues were with understanding what the algorithm task was and what the main test question meant. There were also some issues with the rating scale for the main test questions and with how the visualisations were to be interpreted.

One preliminary participant understood the algorithm task as to be determining whether the image contained a flower or no flower. To combat this issue a mental model based in a physical process was introduced and some language was clarified. The mental model was one of sorting the images into boxes depending on the species instead of just using the word "classification" which was considered by the same participant to be a more abstract, programming related word. This is in accordance with the Rubin and Chisnell (2008, 184) principle of avoiding jargon.

Multiple preliminary test participants found the main test question unclear and experienced general confusion as to what their answers was supposed to contribute. It was unclear to some whether the algorithm already had processed the images or was about to, mostly due to unclear writing in the introduction. To remedy this both the introduction and the test question were rewritten. The introduction went from stating the participants goal very simply to placing it in a context instead. Where the first description simply stated their goal as "…should determine, based on three different kinds of visualisations, if the algorithm has classified an image correctly or not" [8] the second one described a process in four steps. Beginning with the algorithm determining the species in an image, then being asked to provide a "motivation", the motivation is processed into the visualisations and only then do we get to the test participants role. Their role is described as to, based on the

[8] In Swedish: "ska bedöma utifrån tre olika sorters visualiseringar (A, B och C) om en AI har lyckats klassa en bild korrekt eller inte."

visualisations, determine whether the determination of the species was correctly done in the first step.

On a similar note the main test question was changed to put more direct emphasis on whether the test participant think the visualisation indicates a correct or incorrect classification rather than asking then to form a feeling of confidence in a hypothetical situation. The new main test question translates to "Do you think the algorithm determined the species in this image correctly" [9]. Furthermore, the example question in the introduction was clarified to explain what the correct answer would be given that the example image connected to it is explained to have been incorrectly classified.

There was also feedback on having four different ratings, that it gave too little granularity. Therefore, the scale was increased to go from 1 to 6 instead.

One preliminary participant suggested that both the positive and negative relevance maps should use the same colouring. This would make comparison between positive and negative easier. However, it was deemed to make the explanation of the visualisation in the introduction more difficult and less intuitive at first.

Had the preliminary participants been more expert users, e.g. computer science students the feedback might have been more expert. This might however distort the feedback towards being to positive as they would have a better chance of understanding compared to the liberally selected test participants that would later provide responses. As the preliminary participants contained a mix of older subjects without computer science training (expected to have large difficulty) and younger subjects with some computer science training (expected to have little difficulty) there was some healthy mix in the feedback.

2.6.4 **Test materials**

The test was put together as three google forms, one for each ordering of the visualisation methods([A→B→C], [C→A→B], and [B→C→A]). This was deemed practical as the test could be administered either in person or online whenever it fit the test participant the best.

The author being present might affect the test results as this means some participants would have better opportunities for asking questions and might feel less discomfort and uncertainty due to having support nearby. To somewhat mitigate this, it was crucial to answer as few questions as possible and preferably just confirm the things which the test participant has understood correctly as suggested by Rubin and

[9] In Swedish: " Tror du algoritmen lyckades artbestämma denna bilden korrekt?"

Chisnell (2008, p184). There might still be confounding effects due to this but both options were kept open to help with accumulating a greater quantity of responses.

3 Results

This chapter will cover results from the user tests, first quantitative and then qualitative. When applicable results will be structured in way to aid with later discussion.

3.1 Quantitative

3.1.1 **Performance**

The results from the user analysis can be described as a table of four different outcomes divided along two axes. Each image presented to the user can be placed in one of the four outcomes and by summing the results of all images for all users a large total sample size is acquired from which discussion can spring.

Table 3.1 Concept for result matrix

	Classified correctly by network	Classified incorrectly by network
Correctly identified by test participant	Images which are correctly classified and rightly trusted.	Images which are incorrectly classified and rightly not trusted.
Incorrectly identified by test participant	Images which are correctly classified and wrongly not trusted	Images which are incorrectly classified and wrongly trusted.

The first axis signifies whether the image was correctly or incorrectly classified by the network. This is represented as a split of two columns with left being correctly classified images and right being incorrectly classified. This split is defined by the test construction in which 66.7% were chosen from correctly classified images and 33.3% were chosen from incorrectly classified images.

The second axis signifies whether the test participant was able to correctly identify the image. This is represented as a split of two rows with the top row being images which the test participant correctly identified the image as trustworthy or not trustworthy and the bottom row being images in which the user failed to correctly identify the image as trustworthy or not. This split is defined by the answers the test participants give to each main question.

A good result for the left column (correctly classified by the network) would have test participants trusting as many of these images as possible, thus correctly identifying them as trustworthy (top row). This would place these images the top left of Table 3.1.

A bad result for the left column (correctly classified by the network) would have test participants trusting as few of these images as possible, thus incorrectly identifying them as not trustworthy. This would place these images in the bottom left of Table 3.1.

A good result for the right column (incorrectly classified by the network) would have test participants trusting as few of these images as possible, thus correctly identifying them as not trustworthy. This would place these images in the top right of Table 3.1.

A bad result for the right column (incorrectly classified by the network) would have test participants trusting as many of these images as possible, thus incorrectly identifying them as trustworthy. This would place these images in the bottom right of Table 3.1.

First presented here are the total sum results for all the three visualisation methods summed up, Table 3.2.

Table 3.2 Total performance. Performance of the test participants on identifying correctly and incorrectly classified images. Summed for all visualisation types.

	Classified correctly by network. 66.7%, 960 images	Classified incorrectly by network. 33.3%, 480 images
Correctly identified by test participant (55.7%)	39.4% 568 images	16.3% 234 images
Incorrectly identified by test participant (44.3%)	27.2% 392 images	17.1% 246 images

Broken down by visualisation method tables Table 3.3, Table 3.4, and Table 3.5are obtained for A, B, and C respectively.

Table 3.3 Visualisation A, histogram heatmap, performance. Performance of the test participants on identifying correctly and incorrectly classified images.

	Classified correctly by network. 66.7%, 320 images	Classified incorrectly by network. 33.3%, 160 images
Correctly identified by test participant (55.7%)	37.1% 178 images	17.3% 83 images
Incorrectly identified by test participant (44.3%)	29.6% 142 images	16% 77 images

Table 3.4 Visualisation B, linear heatmap, performance. Performance of the test participants on identifying correctly and incorrectly classified images.

	Classified correctly by network. 66.7%, 320 images	Classified incorrectly by network. 33.3%, 160 images
Correctly identified by test participant (55.7%)	39% 187 images	15.8% 76 images
Incorrectly identified by test participant (44.3%)	27.7% 133 images	17.5% 84 images

Table 3.5 Visualisation C, blurred heatmap, performance. Performance of the test participants on identifying correctly and incorrectly classified images.

	Classified correctly by network. 66.7%, 320 images	Classified incorrectly by network. 33.3%, 160 images
Correctly identified by test participant (55.7%)	42.3% 203 images	15.6% 75 images
Incorrectly identified by test participant (44.3%)	24.4% 117 images	17.7% 85 images

As the test participants were presented with 66.7% correctly classified images instead of the 87% which the network produced in actuality. To extrapolate the results onto the real network it will be assumed that the user's ability to correctly identify correctly and incorrectly classified images is the same. This means that the proportions between the top left and bottom left cells will be the same and that the

proportions between the top right and bottom left cells will be the same. See the result of the extrapolation in Table 3.6.

Table 3.6 Extrapolated performance of the test participants on identifying correctly and incorrectly classified images. Given that 87% of the data has been correctly classified by the network.

	Classified correctly by network. 87%, 1253 images	Classified incorrectly by network. 13%, 187 images
Correctly identified by test participant (55.7%)	51.5% 741 images	6.3% 91 images
Incorrectly identified by test participant (44.3%)	35.5% 512 images	6.7% 96 images

3.1.2 **Preferences**

Quantitative preference results were collected both as six Likert-style questions and as a preference selection at the end of the test. The Likert prompts are shown translated in Table 3.7 and the original Swedish as footnotes.

Table 3.7 Likert prompts

Aspect targeted by the questions	Question prompt
Clarity of location	I felt that the visualisation helped me judge where the algorithm had focused[10]
Clarity of positive and negative relevance	I felt that the visualisation helped me judge what spoke for or against the suggested classification[11]
Intuitiveness	I felt that the visualisation helped me approach the problem intuitively[12]
Confidence in agreement	I felt confident when assessing that the AI had classified successfully[13]
Confidence in criticism	I felt confident when assessing that the AI had not classified successfully[14]

[10] In Swedish: ”Jag kände att visualiseringen hjälpte mig att bedöma var algoritmen fokuserat”

[11] In Swedish: ”Jag kände att visualiseringen hjälpte mig att bedöma vad som talade för eller emot den föreslagna klassificeringen”

[12] In Swedish: Jag kände att visualiseringen hjälpte mig angripa problemet intuitivt

[13] In Swedish:Jag kände mig säker när jag bedömde att AI:n lyckats klassificera

[14]In Swedish: Jag kände mig säker när jag bedömde att AI:n misslyckats klassificera

The Likert scale was a 9-choice scale from “Strongly disagree” to “Strongly agree”. The results have been normalised around 5, meaning that a 9 becomes a +4 and a 2 a -3. The average for each question and each visualisation method can be seen in Table 3.8.

Table 3.8 Likert scale average results by class and question

	Clarity of location	Clarity of positive and negative relevance	Intuitiveness	Confidence in agreement	Confidence in criticism
Vis A	1.35	0.5	0.3	-0.45	-0.2
Vis B	0.7	-0.25	0.25	-1.15	-1.4
Vis C	1.2	0.05	0.35	-1.2	-1.15

As for overall preference at the end of the test those can be seen in Table 3.9

Table 3.9 Preference at the end of the test.

Visualisation method	Number of people who preferred it
Visualisation A, Histogram	6
Visualisation B, Linear	3
Visualisation C, blurred	9
No preference	2

3.2 Qualitative

This heading contains a selection of the comments made in the visualisation evaluation and the total evaluation sections of the test. The comments are divided into three main categories; difficulty, solution methods, and other. Furthermore, comments are grouped by the visualisation method the belong to. When several test participants have provided very similar comments only one is shown but the number of similar comments is mentioned. Some comments are not included here due to being non-specific or not contributing to discussion. All the comments have been translated from Swedish and are available in their original language as footnotes.

It can be noted that test participants provided far more comments with regards to solution methods.

In addition to direct comments there should also be noted that the study counted at least five instances of prospective test participants finding the task intimidating or confusing and opting not to take the survey. As the test was available online there may have been more.

3.2.1 **Difficulty**

3.2.1.1 A, Histogram heatmap

"It felt easier to assess when it (the algorithm) was wrong than when it was right."[15]

3.2.1.2 B, Linear heatmap

"It was very difficult to assess whether the AI succeeded based on the visualisations"[16]

Two test participants presented this comment or one very much like it.

3.2.1.3 C, Blurred heatmap

"This one was more difficult to assess"[17]

– With regards to this visualisation method.

---

[15]In Swedish: "Kändes lättare att bedöma när det var fel än när det var rätt."

[16] In Swedish: "Det var väldigt svårt att avgöra om AI:n lyckades utifrån visualiseringarna."

[17] In Swedish: "Denna var svårare att bedöma."

3.2.2 **Solution methods**

3.2.2.1 A, Histogram heatmap

> *"Looked at the focus points to determine if it took in information which 'felt' superfluous. "*[18]

> *...if the algorithm said that the "for" and "against" pixels were close to each other it was probably incorrect. "*[19]

Two test participants provided the comment above or similar. The opposite was also made.

> *"When the blue and the red overlapped a lot, it seemed it was also correct. "*[20]

> *"Mostly went with my gut, this made it difficult to answer these questions as I did not analyse while assessing the flowers"*[21]

> *"There seemed to be a higher chance of success when the edges of the flower were more defined in the visualisation. Could also see when the algorithm was being disturbed by the background"*[22]

> *"I am more and more inclined towards that the algorithm determines correctly when it has considered the largest possible flower area, as many parts of the flower as possible, and only the flower area. "*[23]

Two test participants provided the comment above or similar.

---

[18]In Swedish "Tittade på fokuspunkterna för att se om den tog in information som "kändes" överflödig"

[19]In Swedish "...om algorithmen sa att "för" och "emot" pixlarna låg tätt intill varandra så blev det förmodligen inkorrekt."

[20]In Swedish "När det blå och de röda överlappade mycket var det också rätt verkade det som."

[21]In Swedish "Körde mest på magkänsla, så var svårt att svara på dessa frågor då jag inte analyserade medan jag bedömde blommorna."

[22]In Swedish "Verkade finnas en högre chans att lyckas när blommans kanterna var mer definierade i visualiseringen. Kunde också se när algoritmen störtdes av någonting i bakgrunden."

[23]In Swedish "Mer och mer lutar jag mot att AI:n bedömer korrekt när den bedömt över så stor blomyta som möjligt, så många delar av blomman som möjligt, och endast blomyta"

3.2.2.2 B, Linear heatmap

> *"Once again went more on gut feeling so hard to be precise"*[24]
>
> *"Often the AI was focused outside the plant ... then it was probably wrong"*[25]
>
> *"...it spoke for the AI classifying correctly when the red and blue showed similar patterns"*[26]
>
> *"One thing that that signified incorrect classification was when the algorithm had focused on shadowy parts and therefore gotten the wrong colour"*[27]
>
> *"Checked whether it had gotten the whole structure of the petals and other critical features of the flower."*[28]

[24] In Swedish"Gick igen mer på magkänsla så svårt att vara exakt."

[25]In Swedish "Ofta var AIn fokuserad utanför växten ... då var det nog fel."

[26] In Swedish "...det talade för att AI:n klassifierat korrekt när röda och blå visade liknadne mönster."

[27]In Swedish " En annan sak som talade för att de blivit felaktigt klassificerade var om algoritmen fokuserat på skuggpartier och därmed fått felaktig färg."

[28]In Swedish " Kollade på om den hade fått med hela strukturen av bladen samt andra kritiska egenskaper av blomman"

3.2.2.3 C, Blurred heatmap

"There seemed to be a higher chance of success when the middle and edges of the flower had a stronger colour in the visualisations."[29]

"Could see if the algorithm was disturbed by something in the background by strength of colour"[30]

"When the areas where the AI looked were small strong clusters it was also more correct"[31]

"…it seemed to classify wrongly when it focused on too much. Meaning when the whole flower was bright red."[32]

3.2.3 **Preference motivations**

3.2.3.1 A, Histogram heatmap

"Easier to see points of focus"[33]

"A seemed most consistent in the visualisations"[34]

"Clearer and easier to discern differences in areas it had focused on"[35]

---

[29] In Swedish: "Verkade finns en högre chans att lyckas när blommans mitten och kanterna hade en starkare färg i visualiseringarna."

[30] In Swedish: "Kunde se om algoritmen störde av något i bakgrunden (enligt färgstyrka)"

[31] In Swedish: "När ytorna som den tittade på var små starka kluster var också AIn mer korrekt."

[32] In Swedish:" ... den verkade klassificera fel när den fokuserade på för mycket. Dvs när nästan hela blomman var starkt röd."

[33] In Swedish: "lättare att se fokuspunkter"

[34] In Swedish: "A verkade mest konsekvent i visualiseringarna"

[35] In Swedish: "Tydligare och enklare att tyda skillnader i områden den fokuserat på"

3.2.3.2 B, Linear heatmap

There were no comments by test participants who preferred B.

3.2.3.3 C, Blurred heatmap

> *"The contrast in C made the focused areas clearer."*[36]

> *"Easier with more intense display of colour"*[37]

Three test participants provided the comment above or similar.

3.2.4 **Other**

These comments are more general and not clearly tied to any of the visualisation methods. Therefore, they are not subdivided.

> *"I tried to find patterns in the examples but it felt quite hard to understand how the algorithm works. Does it compare nearby points and contrasts between these…? Can it discern larger shapes?"*[38]

> *"The motivation for and against are often in the same part of the image, abstract for me."*[39]

> *"Colour information should play a large role for the assessment of the system, this is largely missing in the visualisations"*[40]

---

[36] In Swedish: "Kontrasten i C gjorde de fokuserade omårdena mer tydliga"

[37] In Swedish: "Enklare med mer intensiv visning av färg"

[38] In Swedish: "Jag försökte hitta mönster i exemplen men det kändes ganska svårt att förstå hur algoritmen fungerar. Jämför den närliggande punkter och kontraster mellan ... Kan algoritmen urskilja större former?"

[39] In Swedish: "Argumenten för och emot ligger ofta i samma del av bilden, abstrakt för mig."

[40] In Swedish: "Färginformationen borde spela stor roll för systemets bedömning, saknas i mångt och mycket i visualiseringarna"

4 Discussion

This chapter will discuss the results of the tests, primarily the quantitative ones but also some qualitative results.

4.1 Effectiveness in identifying incorrect images

To start with the main research questions. How effective are LRP visualisations for identifying correctly or incorrectly classified images?

The effectiveness will be judged by comparison of two cases, one where the network classifications are trusted as is (the default case), with all the errors this may produce.

The other is the process evaluated in the tests. Subjecting the classifications to user analysis by visualised LRP and separating them into two categories, images where the network algorithm is to be trusted and those where it is not to be trusted.

This will of course have further consequences such as what to do with the images deemed not trustworthy or risks of overconfidence. They would probably require some further analysis or handling or simply be discarded, either way they do create costs to be compensated for by avoiding error.

To compare different visualisations and the two aforementioned cases the following table will be used (Table 4.1). It contains the four possible outcomes for images in this thesis and labels them.

Table 4.1 The four possible outcomes.

	Classified correctly by network	Classified incorrectly by network
Correctly identified by test participant	Images which are correctly classified and rightly trusted.	Images which are incorrectly classified and rightly not trusted.
Incorrectly identified by test participant	Images which are correctly classified and wrongly not trusted	Images which are incorrectly classified and wrongly trusted.

The table will first be investigated for the case of not using user analysis.

This means that all the correctly classified images are rightly trusted, this places them in the top left cell of Table 4.1. This is very desirable. Meanwhile, all incorrectly classified images are wrongly trusted, placing them in the bottom right cell of Table 4.1. This is not very desirable and may have serious consequences.

Onwards to the case of using user analysis.

For images classified correctly by the network there are two possible outcomes.

The first is that the user correctly identifies them as trustworthy, this is the top left cell of Table 4.1. These cases are desirable as they maintain the information quality. However, the network might as well have been trusted to begin with, for these. The second possible consequence is that the user identifies them as not trustworthy, this is the bottom left cell of Table 4.1. These deteriorate the information quality as these images will not be trusted as they should be.

For images classified incorrectly by the network there are two possible outcomes.

The first is that the user correctly identifies them as not trustworthy, this is the top right cell of Table 4.1.These cases are desirable as they increase the information quality. These incorrect classifications can be sent for further analysis or be discarded to avoid error. The second outcome is that they are incorrectly identified as trustworthy this is the bottom right cell of Table 4.1. This does not decrease the information quality as these images would otherwise have been trusted anyway but it is by no means desirable.

The ultimate user analysis would be able to always determine if a network classification is correct or not. Moving all the images which in the default case belong to the bottom right cell to the top right cell (Table 4.1). Meanwhile it would move none of the images which in the default case belong to the top left cell to the bottom left cell (Table 4.1).

While the final network classified 87% of images correctly the sample shown to the test participants had 66.7% of images correctly. This can be seen as the test participants testing a fictional network with lower accuracy.

For the four different cells of the Table 4.1 will of course be evaluated differently depending on what the real-life task is. There are different costs associated with not trusting network outputs or incorrectly and different costs associated with wrongly trusting incorrect classifications. To evaluate the user analysis case two different comparisons will be used.

The first comparison between the default case and the user analysis case is on a simple axis of desirable and undesirable outcomes. Desirable outcomes are defined as the user correctly identifying correct and incorrect classification. Undesirable outcomes are defined as the user failing to correctly identify correct and incorrect classification. This comparison assumes that both the bottom left and right cells of Table 4.1 are equally undesirable and that the top left and right cells are equally desirable. Reality may be very different depending on the task. This comparison is represented in Table 4.2

Table 4.2 Comparison of desirable and undesirable as tested.

	Default case (66.7% network)	User analysis case (All visualisations)	Visualisation A, histogram heatmap	Visualisation B, linear heatmap	Visualisation C, Blurred heatmap
Desirable outcomes	66.7% 960 images	55.7% 802 images	54.4% 261 images	54.8% 263 images	57.9% 278 images
Undesirable outcomes	480 images	44.3% 638 images	45.6% 219 images	45.2% 217 images	42.1% 202 images

For this evaluation method the user analysis does not constitute an improvement. On the contrary, it decreases the proportion of desirable outcomes over all. The blurred heatmap has a somewhat higher proportion of desirable outcomes when compared to the others and the histogram heatmap is the worst by a small margin.

Table 4.3 Comparison of desirable and undesirable extrapolated for 87% network.

	Default case (87% network)	User analysis case (All visualisations) Extrapolated
Desirable outcomes	87% 1253 images	57.8% 832 images
Undesirable outcomes	13% 187 images	42.2% 608 images

When applying the comparison to the extrapolated data it is interesting to notice that while the difference in the desirable outcomes increases, the proportion of desirable outcomes after user analysis is improved somewhat. Overall there is still no improvement.

The second comparison between the default case and the user analysis case is with regards to the proportion of correctly classified images compared to the total amount of trusted images. This means dividing the value of the top left cell of Table 4.1 by the sum of the top left and bottom right.

Table 4.4 Comparison of proportion of correctly classified images among total trusted images

	Default case (66.7% network)	User analysis case (All visualisations)	Visualisation A, histogram heatmap	Visualisation B, linear heatmap	Visualisation C, Blurred heatmap
Proportion rightly trusted	66.7% 960 images	69.8% 568 images	69.8% 178 images	69.0% 187 images	70.5% 203 images
Proportion wrongly trusted	33.3% 480 images	30.2% 246 images	30.2% 77 images	31% 84 images	29.5% 85 images

For this evaluation method the user analysis constitutes a small improvement of post-classification. While the differences may seem small, there is statistically significant improvement in this measure. This is calculated as the cumulative binomial probability of getting X correct classifications out of Y inputs given a 66.6% chance of success.

To elaborate, what is the probability for achieving a ratio (as calculated in Equation 4.1), equal or better than what the user analysis created, by simply trusting a network which has a success rate of 66.7%. This provides an answer for whether, by this measure, it is better to simply use a 66.7% network success rate (the default case) or to apply the user analysis afterwards.

Equation 4.1 Calculation used to create Table 4.4

$$\frac{correctly\ trusted}{correctly\ trusted + incorrectly\ trusted}$$

To evaluate the total user analysis for all the visualisations. The default case has a 3.162% chance of reproducing the ratio of 69.8% that the user analysis produced (568 or more correct classifications out of 814 trusted classifications).

To evaluate the user analysis using the histogram heatmap. The default case has a 15.9% chance of reproducing the ratio of 69.8% that the user analysis using the histogram heatmaps produced (178 or more correct classifications out of 255 trusted classifications).

To evaluate the user analysis using the linear heatmap. The default case has a 22.7% chance of reproducing the ratio of 69.0% that the user analysis using the linear heatmaps produced (187 or more correct classifications out of 271 trusted classifications).

To evaluate the user analysis using the histogram heatmap. The default case has a 9.37% chance of reproducing the ratio of 70.5% that the user analysis using the histogram heatmaps produced (203 or more correct classifications out of 288 trusted classifications).

Table 4.5 Comparison of proportion of correctly classified images among total trusted images extrapolated for 87% network

	Default case (87% network)	User analysis case (All visualisations) Extrapolated
Proportion rightly trusted	87% 1253 images	88.6% 741 images
Proportion wrongly trusted	13% 187 images	11.4% 96 images

When applying the comparison to the extrapolated data the difference again seems small. Here the probability of obtaining an equal or better result is calculated as the cumulative binomial probability of getting X correct classifications out of Y inputs given an 87% chance of success.

The default case has a 10.141% chance of reproducing 741 or more correct classifications out of the 837 trusted ones.

This is probably a sign that the positive effect encountered is relative to the accuracy of the network. It is harder to improve on a more accurate network.

It must be remembered that while by this measure there is some gains to be had this comparison assumes low to no cost to distrusting network classifications. In reality, just disregarding or further investigation of distrusted classifications can have real costs. This measure also assumes low to no cost for conducting user analysis. In reality this might mean many hours of pre-training in addition to time spent with analysis.

Overall these results are deemed to signify no presence of a general benefit. At same time seem to signify a small beneficial effect when measuring the proportion of correctly classified images among trusted images.

It should be noted that this thesis investigated whether a general intuitive effect was present even for very abstract tasks and minimal training. Further research should investigate if and how the small effect with regards to the second comparison can be amplified.

4.2 Contrast against original LRP paper

This section will contrast the very intuitive results of the original LRP paper and the outcome of this study. As mentioned in 2.3 the original paper provided very intuitive LRP visualisations which clearly showed that the algorithm could focus on relevant areas of the image. This was not the case in this study as evidenced both by the visualisation method evaluations and many of the comments. It is true that visualisations A and C received average scores of 1.35 and 1.2 respectively for clarity of location. However, this must be put in context of a scale from -4 to +4 and furthermore all visualisations received evaluations close to 0 for the other characteristics. These scores close to zero are deemed as the visualisations having no significant positive receival among the test participants.

How come the visualisations in this test were deemed as unclear and difficult to interpret? This thesis used the VGG-16 architecture while the original paper used architecture by Krizhevsky et.al (2012), bag-of-words classifiers and a small convolutional architecture. Furthermore, it studied very different datasets which can be contrasted against the flower dataset in this thesis.

Firstly, there is a contrast of distinctiveness and abstractness. Where the original paper shows very different animals and objects (e.g. cat, chicken, cup, spider), all very distinct from one another, the classes in this thesis were all flowers. While the original paper demonstrates through showing location of relevance the quality of

LRP analysis this paper tries to use the same measure to show what qualities in those locations motivated the decision.

Secondly, there is a context of complexity and clarity when compared to the MNIST handwritten numbers dataset. Not only are number perhaps more clearly distinct from each other than flowers are they are also far simpler in their differences in that all differences are geometric. By clarity it is meant that most pixels in the number's dataset is either black or white with some grey in between. This enables LRP to have more clean-cut differences between positive and negative relevance. For example, if a number 3 has been classified as an 8 there will be negative relevance in the empty areas of the 3 where and 8 would have black pixels.

This thesis sheds some light on how LRP analysis applies to evaluating image classification networks tasked with differentiating more complex and similar classes.

4.3 User preferences

To begin, it should be noted that this subject (Neural networks), is not very intuitive and was hard to explain at a correct level. This is true both for explain how the AI works and for explaining what the LRP analysis means. This had both as result that multiple prospective test participants declined to finish the test but probably also coloured some of the performance.

This was somewhat counteracted when the author was present during a test to answer questions and clarify things. Often, the test participant had understood correctly but was intimidated by the subject and upon having their understanding confirmed they could go on without issue.

While the differences between the different visualisations were not very large in terms of performance there are clearer results in terms of preference. The blurred heatmap was the most popular having 9 people choosing it, often motivated by its clear colours. Second was the histogram heatmap with 6 participants preferring it, often due to the points of relevance being very distinct. The linear heatmap was the worst with 3 test participants preferring it, this is probably due to the same issue that was encountered during development, often most of the image is very grey with only a few clearly red and blur pixels.

Many users were confused by the fact that very nearby areas can have strong positive and strong negative relevance. This was the main confusing factor in almost all tests where the author was present. As the blurred heatmap put less emphasis on exact location it counteracted that confusing, letting people approach the task with more confidence and peace of mind.

5 Further research

Like any research project, in answering some questions this project has created many new ones. It is important to remember that the research directions suggested here are not mutually exclusive.

5.1 Amplifying effects

The main barrier identified in this study to applying methods such as this one in practical applications is the size of the displayed effect. While it is positive it is far too small to be of practical use. One avenue which should be investigated is if and how much the effect can be amplified.

Many of the avenues described here are aimed at combating the confusion experienced by many test participants and which might have affected the results. It is also important to remember that these are not mutually exclusive and may be combined in many ways.

5.1.1 **Expert users**

Further research could focus a lot more on creating expert users in order to amplify the positive effects. This can be done by providing more training, potentially with continuous feedback to amplify the speed of learning. This could also mean a lot more interaction with those who conduct the tests. One might also opt to select test participants to evaluate images for a task area in which they already have considerable knowledge.

5.1.2 **Changing visualisations**

An avenue for further research which might be useful for combating participant confusion are alterations to the methods for creating the visualisations.

5.1.2.1 Not caring for positive and negative

A lot of the confusion surrounding the tests were due to the proximity between areas of positive and negative relevance. Furthermore, some users were unsure of how to interpret negative relevance. By simply focusing on the arguments for a lot of confusion can be avoided but this might be done at the expense of performance.

5.1.2.2 Combining visualisations

Users gave different reasons for preferring different visualisations. This might indicate that they have different strengths and weaknesses. By combining them, either side by side or in a single hybrid image, might it be possible to amplify positive effects?

5.1.2.3 Splitting along colour channels?

For the task of identifying flowers, colour plays a large role. Might there be a way to differentiate the relevance along colour channels to provide better information to the test participant? This could help with the problem of LRP showing where the algorithm focused but not explaining how.

5.2 Generalisation

An issue with the results of this study is that they were all from a single network architecture (VGG-16), a single task (Flower classification) and a single dataset (Oxford 102 species flower dataset). Further research could do well to use different network architectures and tasks. Maybe even return to the architecture and datasets used in the original layerwise relevance propagation paper zero (Bach et al. 2015)

5.3 Practical applications

A very interesting avenue for further research would be to approach practical applications. Either gradually or by implementation.

This can be done by keeping the task abstract but varying the distribution in the dataset. One specific variation in task and dataset would be to have a network designed for detecting features which are rare in the input dataset. This could simulate for example the task of separating a few cases of skin cancer from a large amount of benign skin marks.

Another could be to implement a study which focuses on having participants conduct classifications alongside neural networks, either by receiving the suggested class from the network or by also receiving a LRP visualisation. By comparing these

two one could reveal if the visualisations are beneficial in a real-life diagnostic scenario as a supplement rather than just for identifying correctly and incorrectly classified images.

References

Bach S, Binder A, Montavon G, Klauschen F, Müller KR, et al. (2015) *On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation*. PLOS ONE 10(7): e0130140. https://doi.org/10.1371/journal.pone.0130140

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016*). Deep learning (Vol. 1)*. Cambridge: MIT press.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Lapuschkin, S., Binder, A., Montavon, G., Müller, K. R., & Samek, W. (2016). The LRP toolbox for artificial neural networks. *The Journal of Machine Learning Research*, *17*(1), 3938-3942.

Nilsback, M. E., & Zisserman, A. (2008, December). Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on* (pp. 722-729). IEEE.

Rothe, R., Timofte, R., Van Gool, L. (2015). *Deep expectation of real and apparent age from a single image without facial landmarks.* International Journal of Computer Vision (IJCV)raza

Rubin, J., & Chisnell, D. (2008). Handbook of usability testing.

Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 806-813).

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition.* arXiv preprint arXiv:1409.1556.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks?. In *Advances in neural information processing systems* (pp. 3320-3328).

Appendix A Work distribution and time plan

A.1 Project plan and outcome

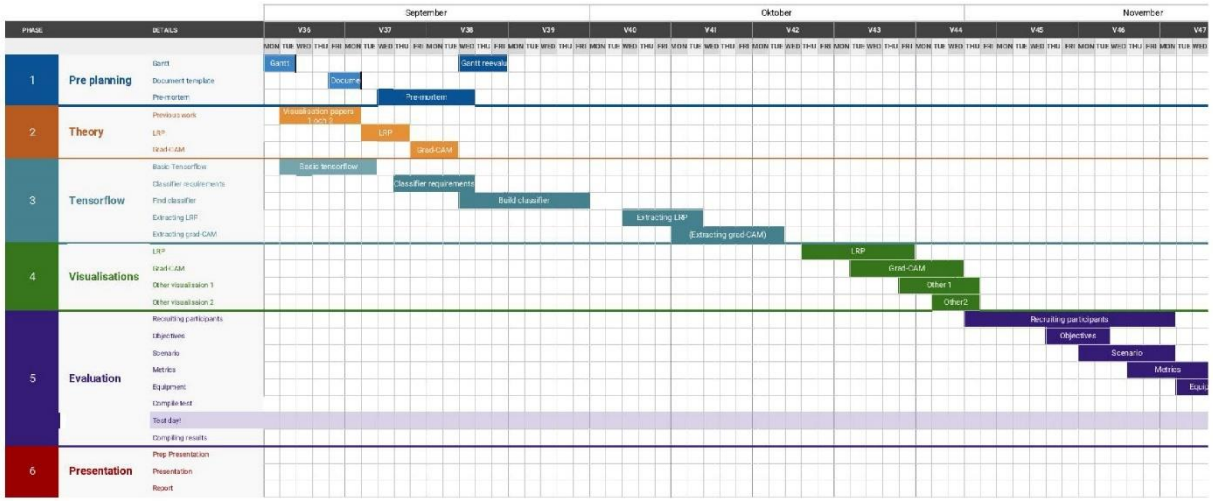


Figure A.1 Planned activities.

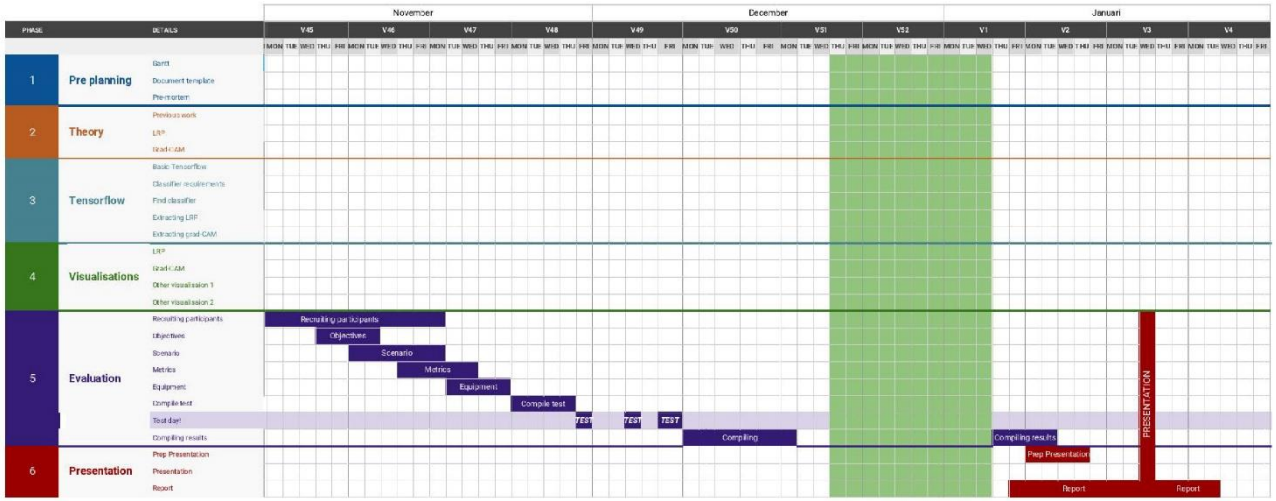


Figure A.2 Planned activities part 2.

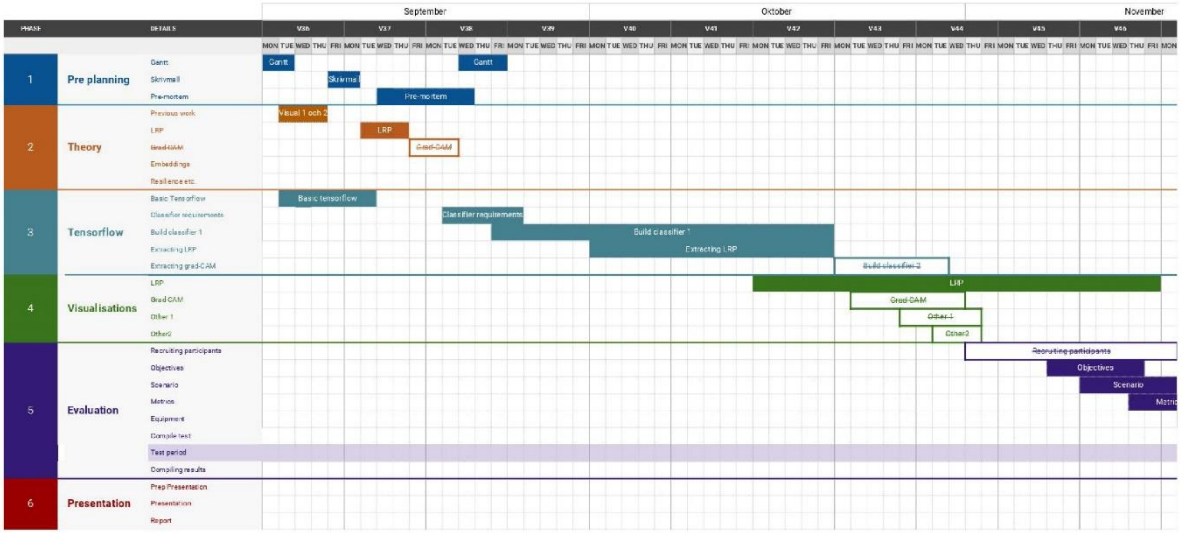


Figure A.3 Performed activities.

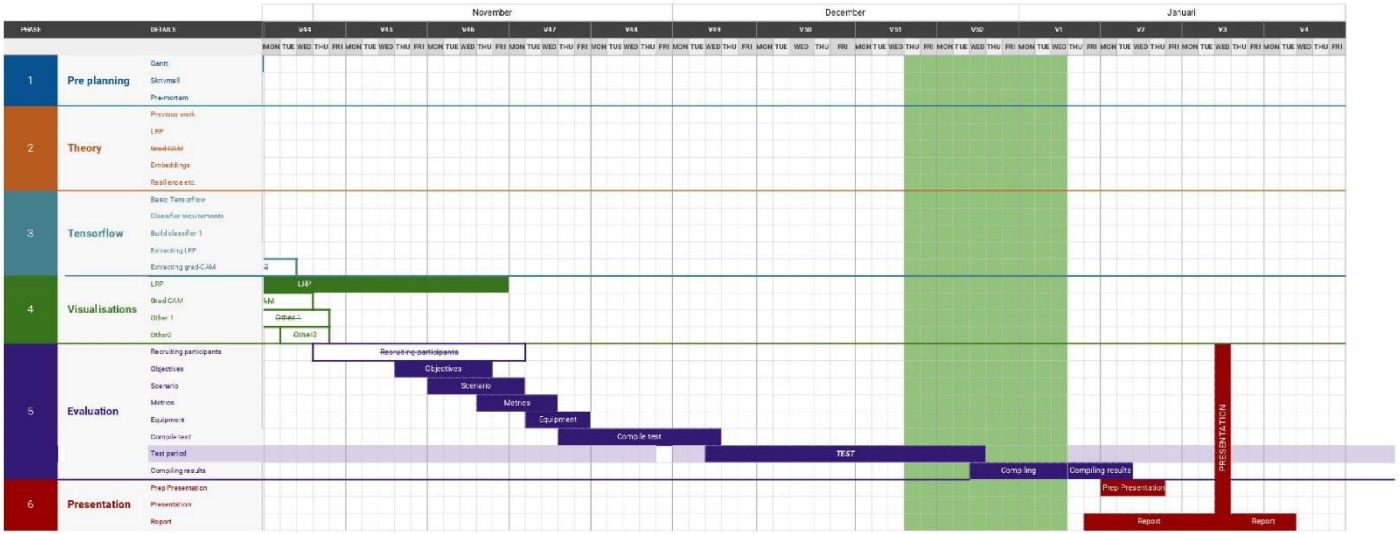


Figure A.4 Performed activities part 2.

Appendix B . Final test variant.

This appendix contains the final test as it was given to test participants. Please note that the test was conducted in Swedish. Some comments for clarification are given in dashed line boxes.

B.1 Introduction

Utvärdering av AI genom visualisering

Hej och tack för att du tar dig tid och hjälper mig med mitt exjobb!

* Required

1. **Ålder**

Förberedelse

Lite bakgrund:
Detta test berör en AI-algoritm som är tränad att artbestämma blommor.
Artbestämmningen innebär att den försöker placera blommor av 102 olika arter i fack beroende på art, ett fack per art den är tränad på.

När algoritmen sedan ska placera bilder den inte tränat på blir det ibland fel, fastän de tillhör samma arter som den tränat på. Din uppgift är att genom att studera var i bilden algoritmen fokuserat försöka bedöma om du tror att den lyckades placera bilden i rätt fack eller ej.

Hela processen för varje bild ser ut så här:
1. Algoritmen får artbestämma en bild och placera den i det fack den tycker matchar bäst.

2. Algoritmen bes om en motivation för varför den placera bilden i just det facket.

3. Motiveringen ges som två visualiseringar som visar var algoritmen fokuserat vid artbestämningen.

4. Här kommer din involvering som testperson, din uppgift är att gissa baserat på den visualiserade motiveringen om algoritmen artbestämde korrekt i steg 1.
Du kommer först att få se några exempel på korrekt och inkorrekt artbestämda bilder. Försök hitta mönster i vad som signalerar att algoritmen lyckades placera blomman i rätt fack, det vill säga artbestämma den korrekt.

!!!Kom ihåg att detta är mycket svårt och det är visualiseringarna som ska testas och inte du som testperson!!!

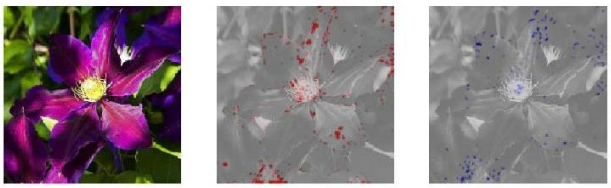
Visualiseringarna:
För att bedöma om algoritmen lyckats placera blomman i rätt fack så kommer du ha visualiseringar till hjälp.

Den vänstra bilden är orginalbilden. Samma som matats in i algoritmen.

Den mittersta bilden och högra bilden visualiserar de pixlar som talar för(i RÖTT) respektive emot(i BLÅTT) den föreslagna klassen. Detta visar var algoritmen har fokuserat vid sin klassificering men också vad den ignorerat.

Testet är indelat i tre delar efter tre olika visualiseringsmetoder.

Visualiseringsmetod A Exempel på inkorrekt artbestämd bild (nedan)



2. **Exempel: Tror du algoritmen lyckades artbestämma denna bilden(ovanför) korrekt? (Rätt svar skulle här vara 1-3 då den är ett exempel på en inkorrekt artbestämd bild)**
Mark only one oval.

	1	2	3	4	5	6	
Nej, absolut inte	◯	◯	◯	◯	◯	◯	Ja, absolut

Visualiseringsmetod A Exempel på korrekt artbestämd bild (nedan)

Visualiseringsmetod B Exempel på inkorrekt artbestämd bild

Visualiseringsmetod B Exempel på korrekt artbestämd bild (nedan)

Visualiseringsmetod C Exempel på inkorrekt artbestämd bild (nedan)

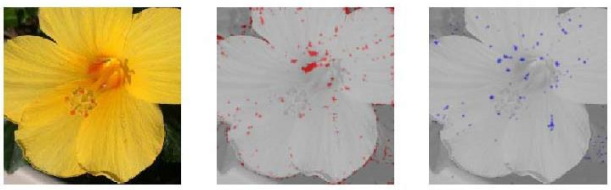
Visualiseringsmetod C Exempel på korrekt artbestämd bild (nedan)

B.2 Learning examples

Exempel visualiseringsmetod A
Här kommer några exempel på korrekt och inkorrekt artbestämda bilder, se om du kan finna några mönster för när datorn gör rätt respektive fel.

Visualiseringsmetod A inkorrekt artbestämd

Visualiseringsmetod A korrekt artbestämd



The examples continue beyond what is in this appendix.
The test contained a total of 8 examples.

B.3 Main test questions

Visualiseringsmetod A

1. **Tror du algoritmen lyckades artbestämma denna bilden korrekt?** *

Mark only one oval.

	1	2	3	4	5	6	
Nej, absolut inte	◯	◯	◯	◯	◯	◯	Ja, absolut

2. **Tror du algoritmen lyckades artbestämma denna bilden korrekt?** *

Mark only one oval.

	1	2	3	4	5	6	
Nej, absolut inte	◯	◯	◯	◯	◯	◯	Ja, absolut

The test questions continue beyond what is in this appendix. The test contained a total of 24 test questions per visualisation.

B.4 Visualisation evaluation

Utvärdering Visualiseringsmetod A

Hoppas första delen gick bra, här har du en katt-gif: https://giphy.com/gifs/cat-funny-cute-11IYKJ5sN73twk

27. **"Jag kände att visualiseringen hjälpte mig att bedöma var algoritmen fokuserat"**
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Håller inte alls med	◯	◯	◯	◯	◯	◯	◯	◯	◯	Håller helt med

28. **"Jag kände att visualiseringen hjälpte mig att bedöma vad som talade för eller emot den föreslagna klassificeringen"**
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Håller inte alls med	◯	◯	◯	◯	◯	◯	◯	◯	◯	Håller helt med

29. **"Jag kände att visualiseringen hjälpte mig angripa problemet intuitivt"**
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Håller inte alls med	◯	◯	◯	◯	◯	◯	◯	◯	◯	Håller helt med

30. **"Jag kände säker när jag bedömde att AI:n lyckats klassificera"**
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Håller inte alls med	◯	◯	◯	◯	◯	◯	◯	◯	◯	Håller helt med

31. **"Jag kände säker när jag bedömde att AI:n misslyckats klassificera"**
Mark only one oval.

	1	2	3	4	5	6	7	8	9	
Håller inte alls med	◯	◯	◯	◯	◯	◯	◯	◯	◯	Håller helt med

32. **Kommentarer (exempelvis hur du använde visualiseringarna)**

B.5 Total evaluation

Utvärdering Total

93. **Föredrog du någon av visualiseringsmetoderna?**
Mark only one oval.

- A
- B
- C

Visualiseringsmetod A

Visualiseringsmetod B

Visualiseringsmetod C

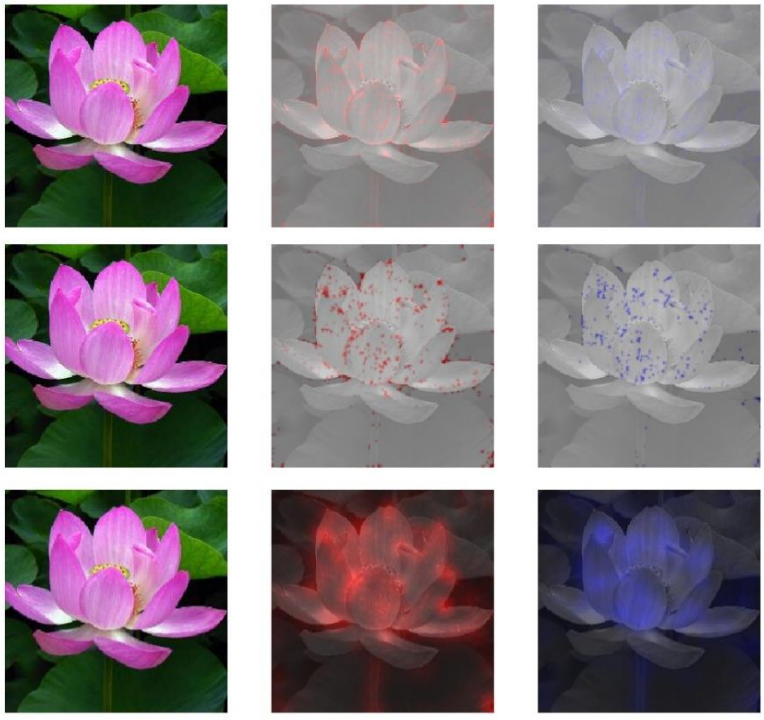
94. **Om du föredrog någon, varför?**

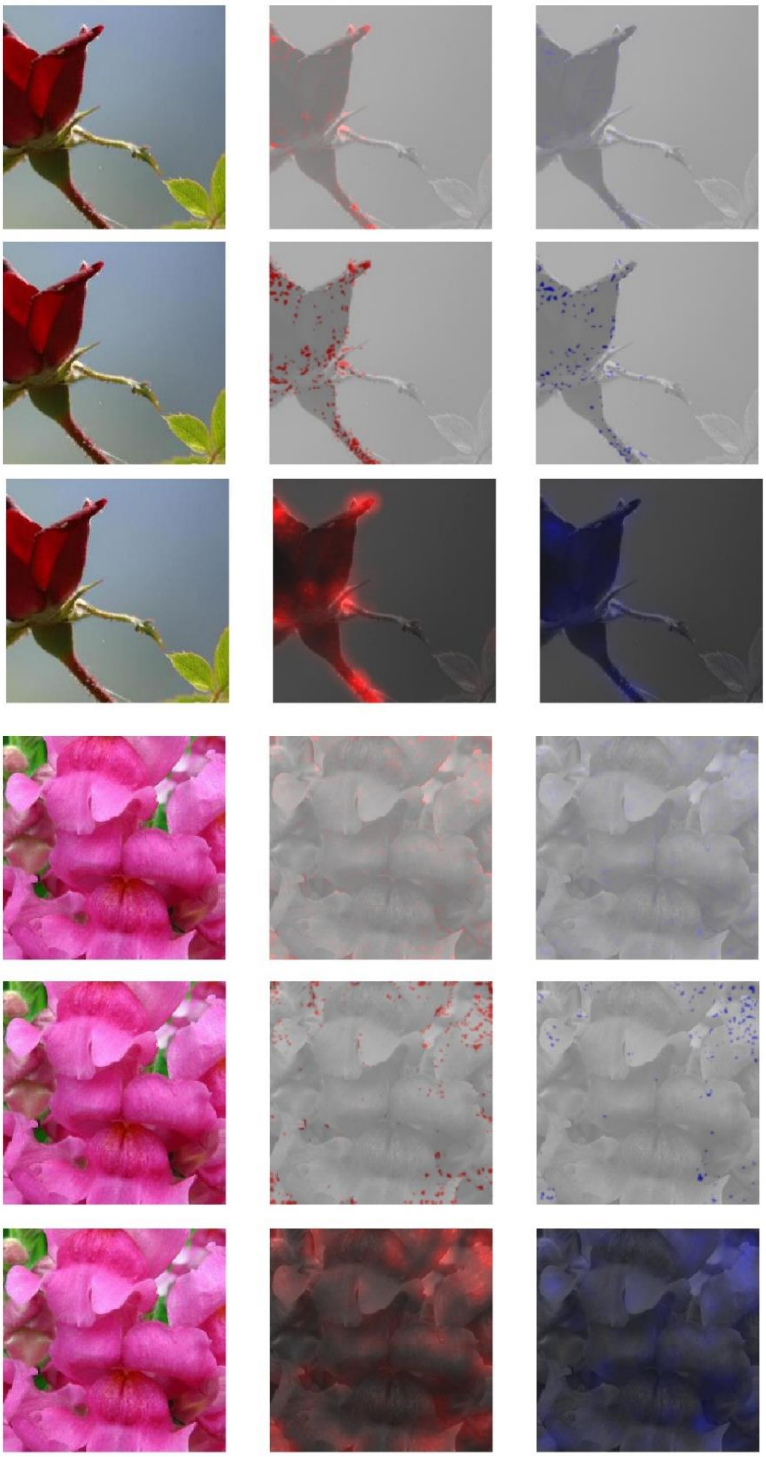
95. **Har du några andra kommentarer?**

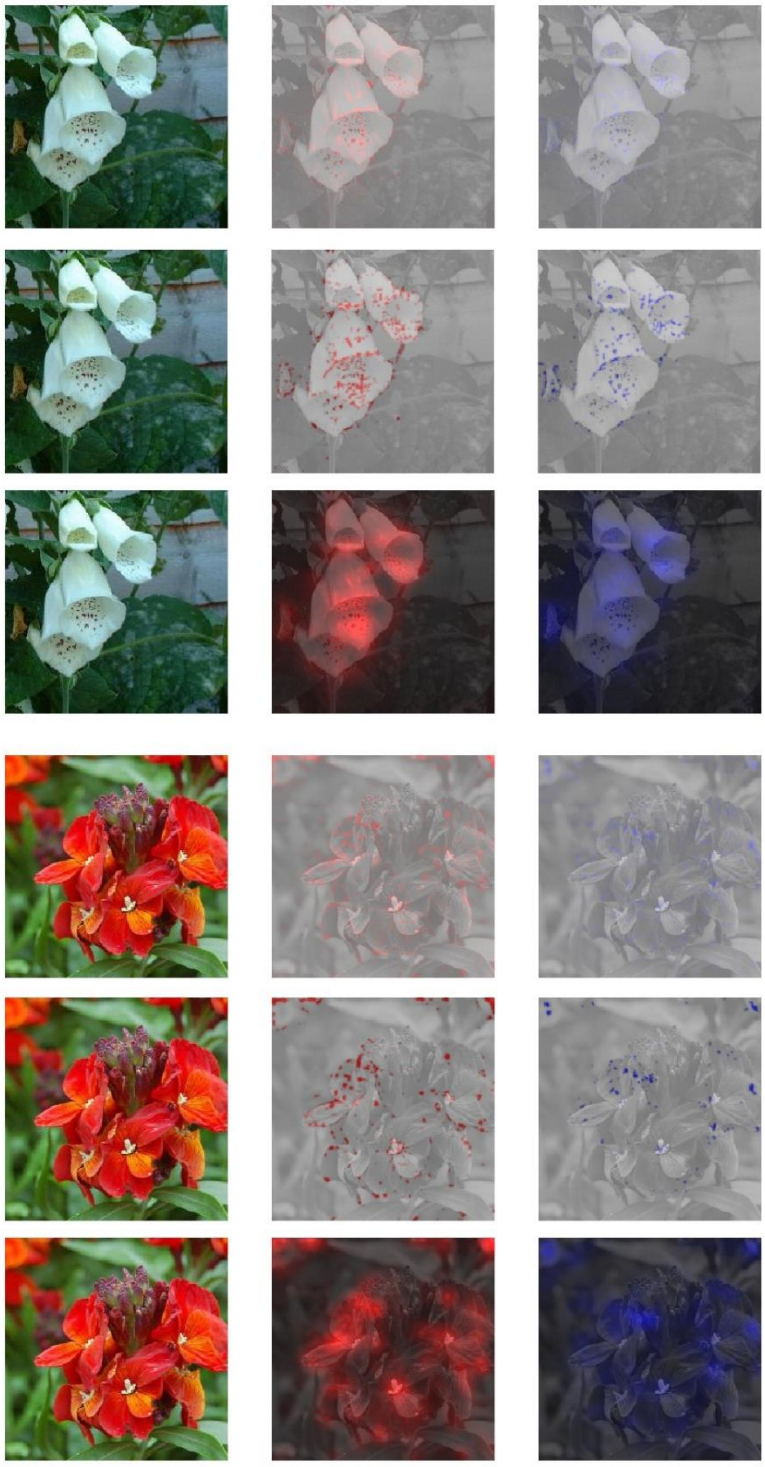
Appendix C Further visualisation examples

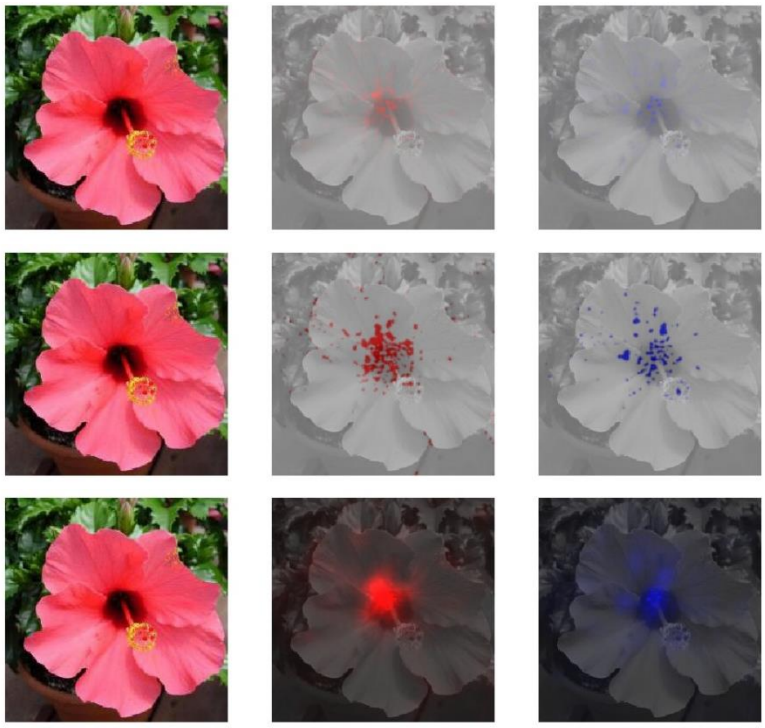
This appendix serves to provide some volume of example images for the interested reader.

C.1 Visualisations of images classified correctly by the network









C.2 Visualisations of images classified incorrectly by the network