



# Temporal Scene Comparison and Similarity Detection

**Aleksander Chalakov**

Lancaster University

A dissertation submitted for the degree of

*BSc Computer Science*

24th March 2023

## **Declaration**

I declare that the work presented in this dissertation is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. Regarding the electronically submitted work, I consent to this being stored electronically and copied for assessment purposes, including the School's use of plagiarism detection systems in order to check the integrity of assessed work.

I agree to my dissertation being placed in the public domain, with my name explicitly included as the author of the work.

Date: 24th March 2023

Signed: Aleksandar Chalakov

# **Temporal Scene Comparison and Similarity Detection**

Aleksandar Chalakov

Lancaster University

A dissertation submitted for the degree of BSc Computer Science. 24 March 2023

## **Abstract**

As advancements in Machine Learning are growing rapidly, Convolutional Neural Networks (CNNs) find their way to be utilised in a large number of innovative and cutting-edge image recognition and computer vision tasks. Temporal scene comparison and similarity detection algorithms are crucial for the detection of similarities and differences between images, which can be used to identify individuals or objects that match a specific pattern, such as a wanted criminal or a missing person. They face multiple challenges in the sphere of image classification in the form of changes in illumination, rotation, translation, and posture, with many various approaches to overcome these limitations, including Siamese Networks, Scale Invariant Features Transform (SIFT), and Class Activation Maps (CAM). This report investigates and focuses on the analysis of the results from deploying and applying those algorithms on a large database and their subsequent behaviour. This large database is restructured in a way that matches the requirements of the algorithms, whilst underlining the importance of data preparation in Machine Learning projects. We investigate the effectiveness of the Siamese Networks in identifying pairs of images with a ground truth relationship, which can face flaws in close scenarios in the form of False positives and False negatives. For this purpose, we also deploy the SIFT, which is a far more demonstrable and explainable approach. It reveals to us the reason the Siamese Network might be wrong and gives us an additional output of the results, achieving a valid hypothesis for a hybrid approach that can successfully classify and compare photos, detecting how similar they are at the same time. A demonstration model is prepared and is further explored how the application of these models can be influential in medical diagnostics and open court.

# Table of Contents

1. Introduction	8
1.1. Project Aims	9
1.2. Report Overview	10
2. Background Research	11
3. Introduction to Convolutional Neural Networks	13
3.1. Artificial Neural Networks (ANNs)	13
3.2. Convolutional Neural Networks (CNNs)	14
3.3. Input layer	15
3.4. Convolutional layer	15
3.5. Pooling layer	16
3.6. Activation function	17
3.7. Fully connected layer	18
3.8. Various CNN encoders	18
4. Methodology	23
4.1. Data	23
4.2. Data Structuring and Preparation	27
4.3. Siamese Networks	31
4.4. Scale Invariant Features Transform (SIFT)	33
4.5. Class Activation Maps	34
4.6. Evaluation	34
5. Results	36
5.1. Outcome of Models	36
5.2. Image Comparison and Error measures	37
6. Discussion	44
7. Conclusion	48
7.1. Personal Reflection	49
References	50
Appendix A: Proposal	53

## List of Figures

<b>Figure 1.</b> A simple three-layered feedforward neural network (FNN), consisting of an input layer, hidden layers, and an output layer. It stands on the basis of common ANN architectures [15].....	13
<b>Figure 2.</b> Simple CNN architecture formed out of 5 layers [15] .....	15
<b>Figure 3.</b> Visual representation of the convolutional layer. [15] .....	16
<b>Figure 4.</b> Max Pooling, Average Pooling and Global Average Pooling are on display [16]..	17
<b>Figure 5.</b> Demonstration of Rectified Linear Unit activation function [17] .....	18
<b>Figure 6.</b> Architecture of AlexNet [18].....	19
<b>Figure 7.</b> Architecture of VGG [18].....	20
<b>Figure 8.</b> Architecture of a ResNet block [18] .....	21
<b>Figure 9.</b> Architecture of DenseNet [18] .....	22
<b>Figure 10.</b> Bus objects from two different scenes from the DAVIS dataset [23] .....	23
<b>Figure 11.</b> Floods from the EU from the EU flood dataset [24] .....	24
<b>Figure 12.</b> Photos from Indoor Places dataset [25] .....	24
<b>Figure 13.</b> Coast with ships coming from the MASATI v2 database [26] .....	25
<b>Figure 14.</b> A bedroom and the inside of the laboratory from the Raghavender Sahdev Places dataset. [27] .....	25
<b>Figure 15.</b> Forestry images from SEMFIRE [28] .....	26
<b>Figure 16.</b> Garden images from the Trimbot [29] .....	26
<b>Figure 17.</b> Living room and Kitchen from an apartment in NYU [30].....	27
<b>Figure 18.</b> Natural images used from the UPenn database [31].....	27
<b>Figure 19.</b> Siamese Networks Architecture .....	32
<b>Figure 20.</b> Data Flow in SIFT Feature Detection Module [43] .....	33
<b>Figure 21.</b> Graph of training over validation accuracy .....	37
<b>Figure 22.</b> 6 comparisons of positively matched pair of images, all of them represent the same scene but from different angles; Comparison 1 - Bridge over water; 2 - Living room with sofas; 3 - Village near a lake; 4 - A person in motion; 5 - Kitchen; 6 – Highway bridges .....	39
<b>Figure 23.</b> 6 comparisons of positively no matched pair of images, all of them represent completely different scenes; Comparison 1 - Bridge and a living room; 2 - Flooding and social interaction; 3 - A dam and a room; 4 - A kiss and a living room; 5 - Tunnel and a room with a bed; 6 – Bathroom and a road .....	40
<b>Figure 24.</b> A comparison between flooding at a dam and under a bridge .....	41
<b>Figure 25.</b> A second comparison between a study room and a living room .....	41

<b>Figure 26.</b> Two images from the same room, but different angles .....	41
<b>Figure 27.</b> A bedroom from two different angles .....	42
<b>Figure 28.</b> Pair of images in comparison utilising the SIFT approach .....	42
<b>Figure 29.</b> The initialising layering of images on top of each other .....	43
<b>Figure 30.</b> Finalised layering of the compared images .....	43
<b>Figure 31.</b> Training loss compared to Validation loss of algorithm model .....	44
<b>Figure 32.</b> Same pair of images from Figure 22 - Comparison 1 .....	46
<b>Figure 33.</b> SIFT visualisation for the first False Positive match.....	47
<b>Figure 34.</b> SIFT visualisation for the second False positive match .....	47

## List of Tables

<b>Table 1.</b> Medium indexes table representing the two connections with a relationship .....	28
<b>Table 2.</b> Equally divided proportions of data sets using K-Fold Cross Validation [34] .....	29
<b>Table 3.</b> Data sets after construction in MATLAB and .csv format .....	30
<b>Table 4.</b> Results of the first 10 epochs in a table format .....	36
<b>Table 5.</b> Results for pair of images in the process of evaluation in the model .....	37

# 1. Introduction

Machine learning is a form of Artificial Intelligence (AI) software that takes experience-based simulations and studies them in order to automate processes in a simpler way [1]. In the computers we use to train those algorithms, this experience that stands at the very basis of machine learning comes in the form of data and it aims to create self-improving algorithms that construct models from the data. The Convolutional Neural Network (CNN) models that result from those actions would be used for accurate predictions or assumptions on unknown observations, thus facilitating complex processes. Those industries have been booming in recent years, as they are now integrated into almost every segment of our day-to-day lives and adopted within all big companies we interact with [2]. Comparing scenes and images and detecting similarities between them is no exception to that and has been deeply embedded within Machine Learning and CNNs as a way of creating models that solve real-life issues. Generally, the similarity would mean just comparing two objects, or in our case images and scenes, and analysing their touching points seeing the analogy between the two [3], however, the more technical and scientific approach, which is going to be used in this paper is to split the images into texture patterns and features, on top of the concepts of image classification and object recognition [4]. This way a neural network can be established which would allow for a scene's features to be examined and the similarity of images to be determined in a greater depth and on a larger scale. We would also be able to determine what are 'good' and 'bad' features. There have been studies like that with various approaches, but the remaining issue, which will be discussed by this paper, is that it is not sure how efficient they are, as they are not very generalisable and not demonstrable.

One of the hardest problems for an algorithm to learn and solve is image classification. The human brain can classify both existing and new images with almost 100% accuracy when presented with several available examples. Machine learning algorithms have been created to precisely replicate this behaviour of the human brain. However, the issue becomes far more complicated when images are taken under different conditions, such as changes in illumination, rotation, or translation of objects, hidden or incomplete objects, and different postures in the case of facial recognition. These conditions result in hundreds of different images containing the same object, which adds to the complexity of the recognition and classification problem [5].

There have been many attempts to achieve scene comparison and similarity detection within image classification and get an understanding of how it works and how it can be put into use. Many of those successes have been put into use in various spheres ultimately helping society solve a lot of issues and find an answer to complex challenges. Of course, those processes have progressed from the beginning and the very early tries in image comparison, as now we have new approaches such as Siamese Networks, SIFT and Class Activation Maps (CAM). Each of those shows a different perspective in scene resemblance and would be limited on its own, thus in this paper, we're going to look at and analyse all of them.



Talking about limitations in scene comparison and similarity detection, several key things inflict stagnation and restrict processes from having positive results. In some approaches, the features that are used to create a model are chosen manually by an expert with specific knowledge in a certain field, which however is not as automatic as an algorithm choosing it, which may lead to some features dominating the performance of the classifier, thus creating a bias in a way of analysing the scene later with those features [6]. In a different manner, image comparison wasn't able to help in the case of Encrochat, when authorities had at their disposal thousands of images of crime scenes and illegal resources, which could be used to catch the perpetrators [7]. If the authorities had a natural general scene comparison model, which was able to analyse similar images and compare them with their features and similarities, extract key points and present them in a way understandable for the police, the prosecuted criminals would be way more. Just as the case is in medical images, most diagnoses are in single shots, while with a model like the one mentioned above, we would be able to build a temporal model of a diagnosis that changes over time and shows doctors invaluable information about how things are accelerating. This is the aim of the report - to create something indispensable, a model which can be used in many industries with multiple benefits and advantages.

## 1.1. Project Aims

The overall aim of this project is to investigate models for image and scene comparison and their potential application in medical diagnostics. In order for this to be achievable, we would need to curate data sets and carry out a literature review, while also examining baseline approaches for scene comparison. As challenging as the project is, it will be completed with the following objectives:

- Outlining of comparable features and common models of comparison
- Investigation of the object classification approach. This will involve Siamese networks and scene comparison between 2 photographs from the data sets concluding if the scenes are the same or not.
- Investigation of the use of Scale Invariant Features Transform (SIFT), as it is able to find similarities and collations, with feature extracting and matching while being invariant to scale and rotation.
- Consideration of the use of Class activation maps, where if there's a similarity in scenes, it gives a heat map to indicate which bits of the image are similar and analyse correlations.
- Investigation of the influence of certain common features in scenes with large similarities.
- Consideration into how models can be applicable and helpful in medical diagnostics or in open court.
- Preparation of a demonstration model.

## 1.2. Report Overview

The structure of the report has been broken down into the following segments that are mentioned below. Each segment contains a brief overview of the information that is going to be included in it.

**Background Research** includes a description of similar projects and previous attempts at scene comparison through Siamese Networks, SIFT approach and Class Activation Maps, while also indulging in presenting the matter through explainable AI.

**Methodology** involves a data preparation section, along with an insight into the chosen approaches, to prove my thesis and show how the data has been retrieved and evaluated through multiple spectrums. It includes detail about the range of images and resources that are going to be analysed throughout the project. The section further discusses the process of how the different approaches have been implemented.

**Evaluation** provides an analysis of what has happened throughout the project with all of the different approaches and evaluates their performance and implementation.

**Results** discuss the findings and results of the study, providing an in-depth analysis of the final results. Additionally, graphs and tables are presented in order to better visualise the outcome of the project and directly see the relationships between the algorithms.

**Discussion** describes the outcomes of our results and evaluates our findings in light of the bigger picture, discussing if the aims and goals of the project were met throughout the different spectrums involved in it. It outlines what was learned.

**Conclusion** is the final segment of the report. It comprises an analysis and key points from the completed project, along with afterthoughts and further work on how the study can be improved.

## 2. Background Research

The earliest machine-learning projects in the sphere of image and scene comparison were directly related to computer vision. Those researchers mostly drew upon an enhanced cascade of simple features [8][9], and had a feature descriptor, which was used to decipher and extract the components in a way called Histograms of Oriented Gradients (HOG). It basically involved dividing the image into small cells and calculating a local 1-D histogram of gradient directions or edge orientations over the pixels in each of those cells. What resulted from that was numerous histograms that were combined to create a representation of the image. To improve accuracy, the local responses are contrast-normalised using a measure of local histogram “energy” over larger spatial regions. The resulting normalised descriptor blocks are referred to as HOG descriptors. These descriptors are tiled across the detection window using an overlapping grid and used as a feature vector for a conventional Support Vector Machine (SVM) based window classifier. For its time, it provided multiple advantages, as it was able to capture edges or gradient structures that are very characteristic of local shape. It allowed for easily controllable invariance to local transformations. It specifically became more powerful when combined with Lowe’s SIFT, which had local spatial histogramming and normalisation built into it [10]. Those approaches proved to be more efficient than anything similar to them and were used as groundbreaking research at the time.

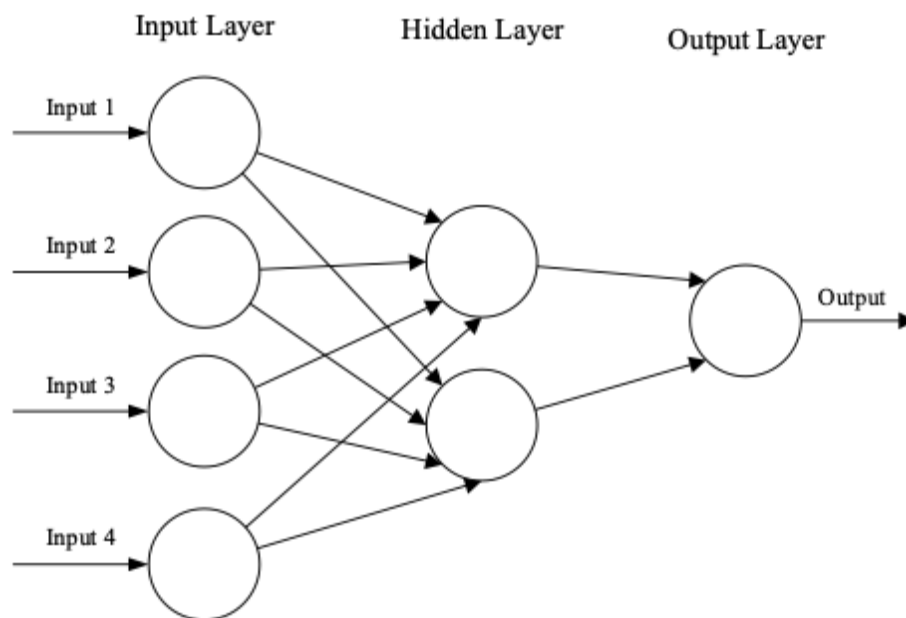
However, this approach was pretty basic and with time passing by, the processes that needed to be serviced became more complex, so the machine learning algorithms had to adjust to that and progress further, becoming more complicated themselves. In that manner, there has been extensive research into automatically identifying individuals via holistic multi-unit knuckle recognition, which is based on a deep neural network and has proven to be successful in solving cases and catching criminals by only having their hands in the image [11]. This study presented an end-to-end deep learning framework for identifying individuals through images of the dorsal side of their hand, using knuckle creases as a biometric trait. The proposed approach includes detecting the knuckle regions and ensuring anatomically-correct detection using a new quality metric, extracting deep features that differentiate individuals, and exploring the identifiability of each knuckle. Out of all that, they create a model, which is also further extended to a holistic matching approach for identification of the whole hand using all available knuckles, making it more accurate and powerful. It is only one of the multiple such resources that have shown how image comparison and similarity detection can be useful in real-life situations and used for good to help people overcome various challenges. The research itself stems from a more general hand-recognition research, which was started by Sue Black, in order to catch paedophiles who have shown only their hands in the photo imagery they’ve taken [12][13]. More broadly, in this study, the authors introduce a hand-based person recognition method called Multi-Branch with Attention Network (MBA-Net) for criminal investigations. The proposed method learns attentive deep feature representations from hand images captured by digital cameras and is trained in an end-to-end manner.

All of this is done in a way for Explainable AI to be created and integrated successfully in every way that's needed. Explainable AI can be split into two main categories, one of them being transparency behind the AI's decisions, understanding of its model structure and its components and understanding of the algorithm that trained it, while the second one is more of a post-hoc explanation into how and why the specific processes have occurred in order for the AI to reach its conclusion, giving analysis and visualisations to help prove its point [14]. This way it becomes a more reliable and acceptable method for more important professions, such as medical doctors, to base their decisions on Explainable AI. This is what we're striving for in this report as well, we want demonstrable and explainable results.

### 3. Introduction to Convolutional Neural Networks

#### 3.1. Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) are computational processing systems that draw a lot of their inspiration from how biological nerve systems, like the human brain, function and operate. They consist of interconnected computational nodes called neurons that work together to learn from input data and optimise the output [15]. A signal can be transmitted from one neuron to another through each connection. The receiving neuron can process the signal and transmit it to other connected neurons downstream. Neurons have states that are represented by real numbers, which can either be 0 or 1 [5]. Usually, those neurons are then organised into various layers. All of this can be modelled as a structure with an input layer, where the input is in the form of a multidimensional vector and then loaded into, then distributed to hidden layers, which make decisions based on the previous layer and use stochastic changes to determine whether the output has been improved or worsened. When there are multiple hidden layers stacked on top of each other together, this process of learning is also called deep learning [15][16]. Figure 1 shows a basic model of this structure, presenting the layers in the following format:



**Figure 1.** A simple three-layered feedforward neural network (FNN), consisting of an input layer, hidden layers, and an output layer. It stands on the basis of common ANN architectures [15]

Such neural networks are capable of learning to perform tasks by considering examples without the need for task-specific programming. As they learn, their performance progressively improves. Even though initially, the original intent of neural networks was to solve problems in a way that mimicked the human brain, however, with time, the focus and

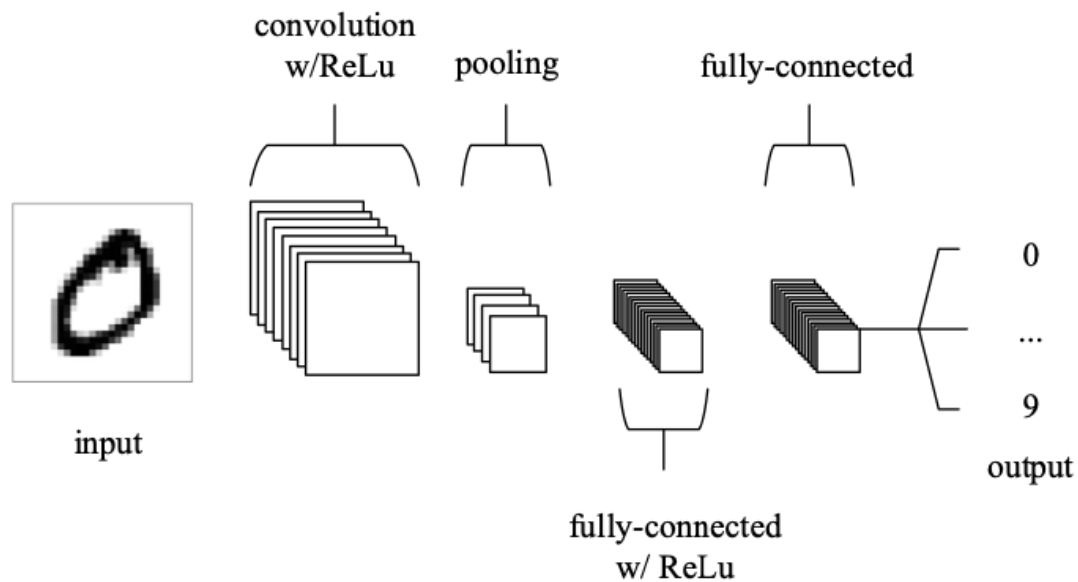
emphasises had shifted to being tailored to certain mental abilities, which resulted in them slowly being introduced in the computer vision sphere, solving tasks of object recognition and image classification [5].

In image processing tasks like ours, there are two main learning paradigms in the face of supervised and unsupervised learning. Supervised learning involves using pre-labelled inputs as targets for training. For each of those training examples, there are input values and defined output values. The aim is to minimise the model's overall classification error by accurately calculating the output value of each training example during the training. Unsupervised learning is different from supervised learning in that there are no labels in the training set. The success of unsupervised learning is typically served by the ability of the network to enhance or decrease an associate cost function. Despite that, most image-based pattern recognition tasks do rely on supervised learning for classification purposes [16].

### 3.2. Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) and conventional ANNs share a common characteristic in that they consist of self-optimising neurons that learn through experience. Following and including the fundamental building block of ANNs, each neuron receives input and performs an operation, such as a scalar product followed by a nonlinear function. There are multiple differences between CNNs and traditional ANNs, however, the key one would be that CNNs are specifically designated for pattern recognition tasks in images. There are image-specific features incorporated into the architecture of the network, making it better suited for image-based tasks. This is done as the layers of the CNN are made up of neurons organised into three dimensions: the depth, the breadth, and the height of the input space. It is important to note that the depth does not refer to the total number of layers within ANN, but rather describes the third dimension of an activation volume. In CNN, the neurons within each layer only connect to a small region of the preceding layer [15].

CNNs generally are built from an input layer and hundreds of feature detection layers. Those layers are split into 3 main types, each doing a different thing with different priorities: convolutional layers, pooling layers, and fully connected layers [5]. When they are stacked, the architecture of a CNN is formed, looking like that:



**Figure 2.** Simple CNN architecture formed out of 5 layers [15]

CNNs utilise weight sharing, which significantly reduces the number of trainable network parameters. This results in an enhanced and improved generalisation that helps prevent overfitting. Furthermore, the feature extraction and classification layers are learned simultaneously, which leads to highly organised and feature-reliant model output. CNNs are substantially easier to implement at a larger scale compared to other neural network architectures. While understanding what Convolutional Neural Networks are, it is not enough only to investigate what they do, but it is rather crucial to delve into the specifics of each layer to gain a better understanding of how to construct and optimise these networks [16].

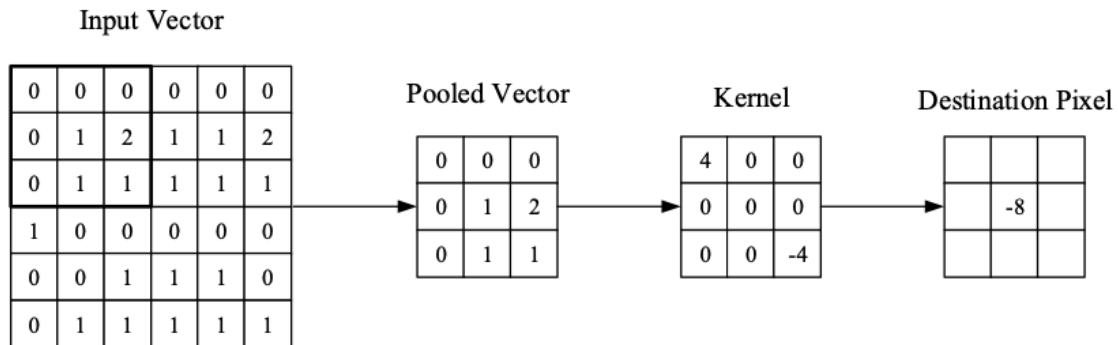
### 3.3. Input layer

The input layer in a CNN would be the image within which objects are going to be classified or compared to. The size and depth of the pixels within the image will determine the number of input values. The goal of the CNN is to identify or translate these values into objects that belong to specific classes. The CNN architecture is inspired by the human visual cortex, which contains cells that are sensitive to specific regions of the visual field. Generally, some of these cells are sensitive to particular image features such as edges and curves [5].

### 3.4. Convolutional layer

The convolutional layer is one of the most important fragments of the architecture of a CNN. It is so, as its parameters centre around the use of learnable kernels, which are typically small in spatial dimensionality but provide an extension throughout the depth of the input. When data enters the convolutional layer, the layer convolves each filter across the input's spatial dimensionality to produce a 2D activation map. Those maps can be visualised. As the network processes the input, it would calculate the scalar product for each value in the

designated kernel. This process is illustrated in Figure 3. Based on this calculation, the network would teach kernels that “fire” when they detect a specific feature at a particular spatial position in the input [15][16]. These actions are frequently referred to as activations.



**Figure 3.** Visual representation of the convolutional layer. [15]

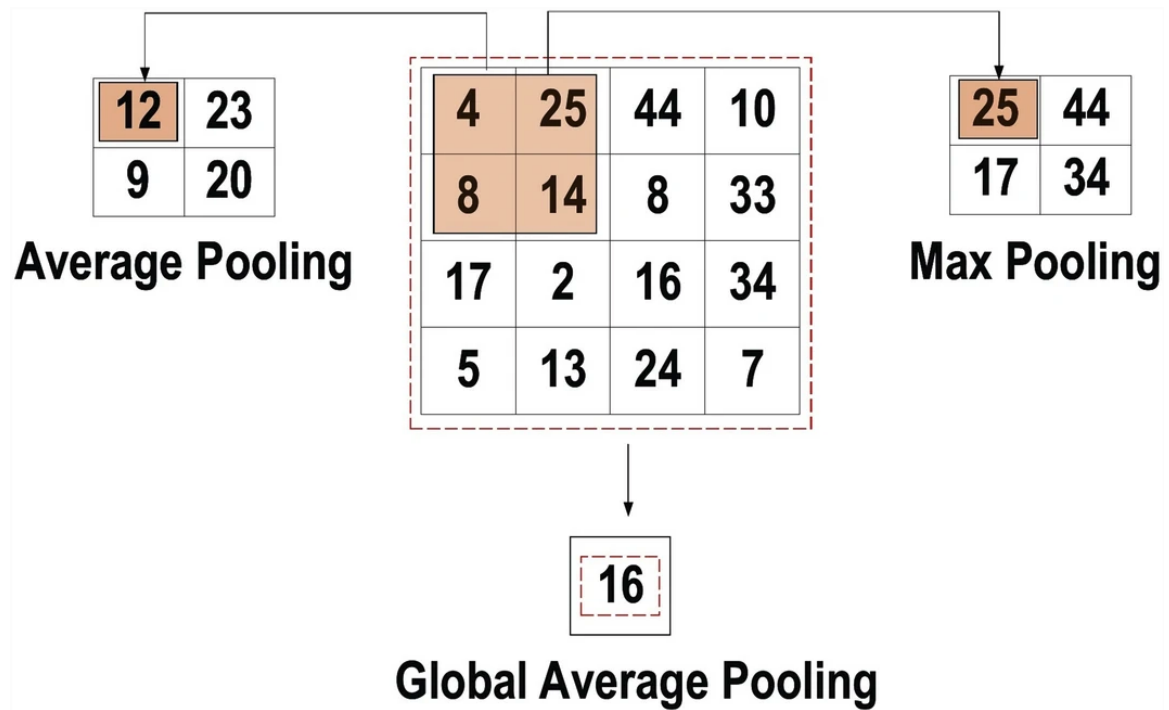
As seen in Figure 3, the kernel’s central element is placed over the input vector, and a weighted sum of itself and any nearby pixels is calculated and replaced with the result. Each kernel in the convolutional layer corresponds to an activation map, which is stacked along the depth dimension to produce the full output from the layer. So, in order to make models easier to train, as mentioned before, CNNs ensure that each neuron in a layer is connected only to a small section of the input volume, also known as its receptive field size [15]. Each convolution layer is followed by an activation function from the likes of ReLU - Rectified Linear Unit, however, more on it in Section 3.6.

### 3.5. Pooling layer

The pooling layer is used to gradually decrease the dimensionality of the representation, hence lowering the number of parameters and computational complexity of the model even more. It literally sub-samples the feature maps that are created after the convolutional layer, shrinking large-size feature maps to smaller-sized ones. These pooling layers operate on each activation map in the input using the “MAX” function, typically in the form of max-pooling layers with 2 x 2 kernels applied with a stride of 2 along the input’s spatial dimensions. By doing this, the depth volume is kept the same at its usual size, but the map is scaled down to 25% of its original size [15][16].

Due to the destructive nature of pooling, there are typically only two methods of max-pooling observed, with the stride and filter size both set to 2 x 2 to extend through the input’s spatial dimensionality. Generally, using a kernel size greater than 3 reduces the model’s performance, so it is not done. Apart from max-pooling, there are a few other types of pooling, the most notable ones being general pooling or also called global average pooling (GAP) and average pooling [15][16]. Figure 4 illustrates all those types of pooling in the best possible way:



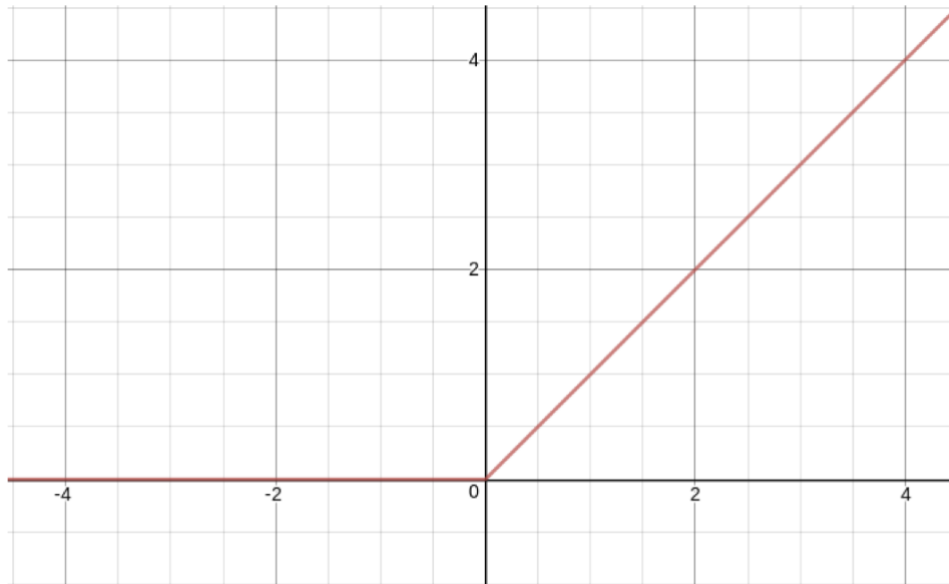


**Figure 4.** Max Pooling, Average Pooling and Global Average Pooling are on display [16]

### 3.6. Activation function

An activation function is something that is seen after every convolutional layer. The core function of an activation function in any type of neural network would be to map the input to the output. The input value is computed by adding the weighted summation of the neuron input with its bias. The activation function then decides whether or not to activate the neuron based on the input and generate the corresponding output. Activation layers demonstrate non-linear behaviour, which enables non-linear mapping of the input to the output, further allowing the network to learn complex patterns. The function must also be differentiable for error backpropagation to train the network in the best possible way. Its placement in the architecture of CNNs majorly improves the performance of a certain task [16].

One of the most notable activation functions that are specifically important for this project is called ReLU, coming from Rectified Linear Units. The way it works is that it generally maintains positive values while mapping all negative values to 0 [17]. Its lower computational load is the key benefit over any other type of activation function:



**Figure 5.** *Demonstration of Rectified Linear Unit activation function [17]*

As seen in Figure 5, it simply outputs 0 whenever  $x < 0$ , and keeps the value of  $x$  within a linear function when  $x \geq 0$ . That's why ReLU is used as the activation function in each hidden layer in a neural network.

### 3.7. Fully connected layer

The Fully connected layer (FC) is typically located at the end of the CNN architecture, as seen in Figure 2 and functions as the classifier. Each neuron in this layer is connected to all neurons in the previous layer, known as the Fully Connected approach. This layer follows the feed-forward principle of a conventional multilayer perceptron neural network. The input to the FC layer is a vector created from the feature maps after flattening the output from the last pooling or convolutional layer [16]. After the FC layer, what's left is the output layer where the final results are presented. Usually, in the output layer, there are some loss functions that are being utilised so that the predicted error across the testing samples can be calculated. Loss functions differ per CNN architectures, so they will be looked into whenever we go to the Siamese Networks.

### 3.8. Various CNN encoders

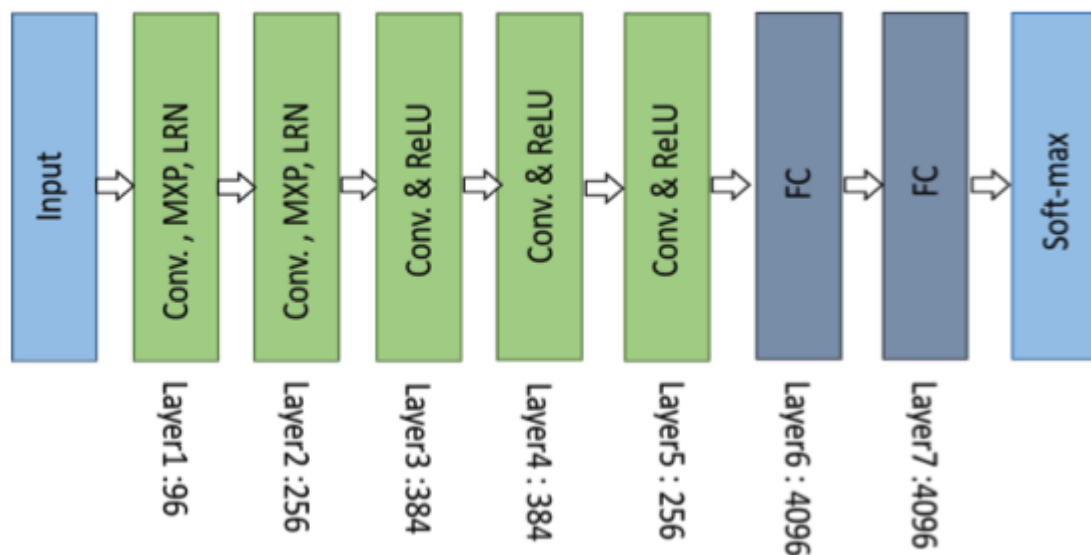
Over the past decade, multiple CNN architectures have been introduced, each with different encoders and specifics tailored for specific tasks, with the model architecture playing a crucial role in improving performance for various other applications. CNN architecture has undergone various modifications, including structural reformulation, regularisation, and parameter optimization, since 1989, with the most significant performance change being achieved through processing-unit recognition and the development of novel blocks in terms of network depth.

In order to understand the architectures we're going to tackle in this project, we first need to analyse and go in-depth into other more popular CNN architectures and distinguish how different they are from what we are applying. Grasping the features of these architectures is essential to choosing the right one for a specific task in a project [16][18].

### *AlexNet*

The history of deep CNNs started with the introduction of LeNet, which was limited to handwritten digit recognition tasks and could not be scaled to all image classes. However, this was up until 2012, when Alex Krizhevsky and his team proposed a deeper and wider CNN model than LeNet, also called AlexNet, which won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in the same year [19]. This breakthrough achievement with AlexNet significantly advanced the field of machine learning and computer vision for visual recognition and classification tasks, marking a turning point in the interest of deep learning. AlexNet achieved state-of-the-art recognition accuracy and outperformed traditional machine learning and computer vision approaches, marking a new era for deep learning [18].

AlexNet's architecture consists of five convolutional layers, two fully connected layers afterwards, followed by a Softmax layer at the end, compared to LeNet's 5 layers. Its architecture can be seen in Figure 6:



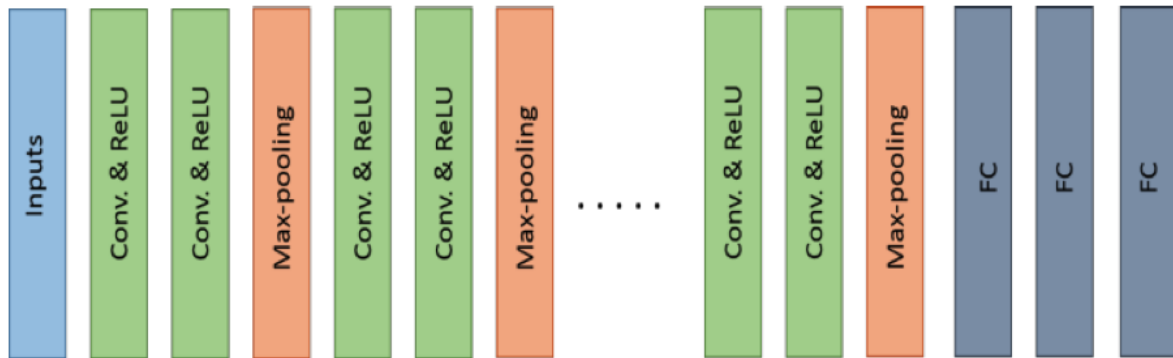
**Figure 6.** Architecture of AlexNet [18]

As seen in Figure 6, the first convolutional layer consists of 96 receptive filters that are 11x11 in size, while performing convolution and max pooling with Local Response Normalisation (LRN). LRN can generally be either applied on a single or across multiple channels or feature maps. What it does is enhance the generalisation in order to decrease overfitting. The second layer uses 5x5 filters with the same operations. The third, fourth, and fifth convolutional layers use 3x3 filters with 384, 384, and 256 feature maps, respectively. In them, ReLU is used as a non-saturating activation function which enhances the rate of convergence and additionally alleviates the vanishing gradient problem. The two fully connected layers are

used with dropout, and two networks with similar structures and the same number of feature maps are trained in parallel for this model [18][19].

### VGG

Another CNN encoder comes in the face of the Visual Geometry Group - VGG [20]. As a runner-up in that ILSVRC competition, their work was highly praised, as it showed that the depth of a neural network is a crucial factor in achieving high recognition or classification accuracy in CNNs.

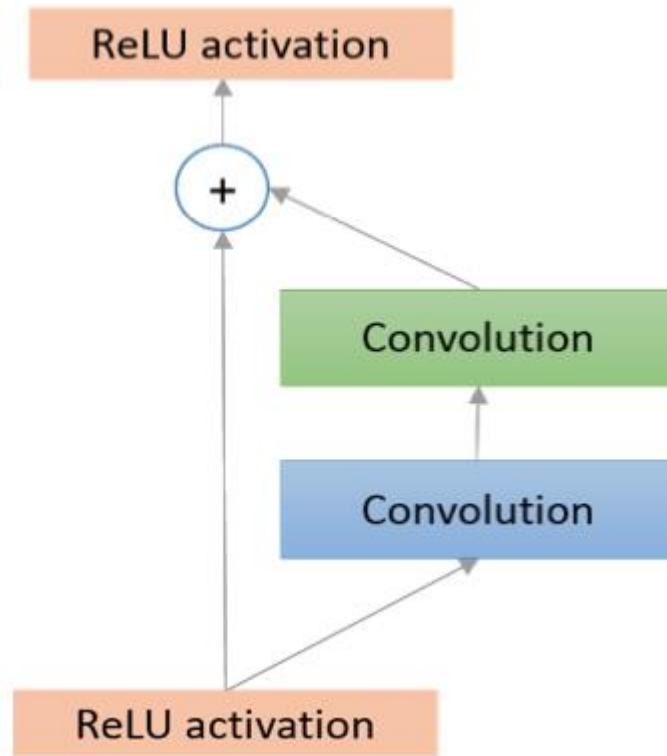


**Figure 7.** Architecture of VGG [18]

As seen in Figure 7, VGG's architecture consists of two convolutional layers using the ReLU activation function, followed by a max pooling layer and several fully connected layers also using ReLU. The final layer in the network is a Softmax layer for classification. It's different from AlexNet in that its most complex version has 19 more layers, aiming to simulate the network's representational capacity in depth. Another innovative idea that was introduced in VGG is that they utilised a layer of 3x3 filters instead of larger filters, which showed that small-size filters could produce the same effect as large-size ones while reducing computational complexity. VGG also inserted 1x1 convolutions to regulate network complexity and implemented max pooling layers and padding for spatial resolution. Nevertheless, it had a major shortcoming in its excessive computational cost due to its use of around 140 million parameters [16][18].

### ResNet

Residual Network - ResNet, was designed in order to tackle and address the vanishing gradient issue of deep networks, winning the ILSVRC 2015. ResNet includes several types of networks that differ based on the number of layers, ranging from 34 to 1202 layers. The most notable type though would be ResNet50, which includes 49 convolutional layers and a single fully connected layer. The main innovative concept in ResNet would be its utilisation of the bypass pathway, which is done by something called ResNet blocks.

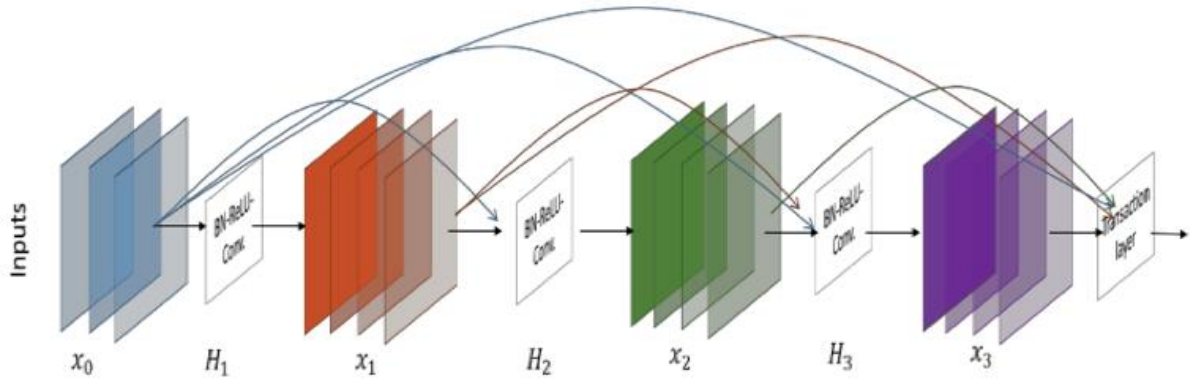


**Figure 8.** Architecture of a ResNet block [18]

As seen in Figure 8, the ResNet block comprises a conventional feedforward network and a residual connection. A ResNet network generally comprises multiple of those basic blocks. Comparing the 2015 ILSVRC ResNet version, which had 152 layers of depth, to VGG and AlexNet would show us that it represents 8 times the depth of VGG and 20 times the depth of AlexNet, with lower computational complexity [16][18].

### *DenseNet*

The last model architecture that's going to be analyzed in this paper would be DenseNet, which is a deep learning network that comprises densely connected CNN layers [22]. In its sense, it builds on top of ResNet, solving its issue of preserving information through individual transformations, with many layers contributing very little or no information, thus resulting in an enormous number of weights. The output of each layer in a dense block is linked to all subsequent layers, promoting efficient feature reuse, and employing cross-layer connectivity, which in its way results in a significant reduction in network parameters. In Figure 9, the architecture is seen as being composed of multiple dense and transition blocks, which are inserted between adjacent dense blocks [16][18].



**Figure 9.** Architecture of DenseNet [18]

The cross-layer connectivity allows for clear discrimination and distinguishing between added and preserved information, as DenseNet concatenates the features of the preceding layer instead of adding them. It has a flaw in that this type of narrow layer structure eventually becomes parametrically expensive, in addition to the increased number of feature maps [16].

## 4. Methodology

### 4.1. Data

This project focuses on scene comparison and similarity detection, so the data used would need to include: i) two types of data sets - one would be the natural general scenes dataset and the other one would feature a more controlled set - artificially generated matching scenes, ii) many photos taken of different angles but of the same scenes, iii) corresponding segmentations - to be used for distinguishing between elements and scenes. The databases that are going to be taken into consideration include DAVIS [23], EU Flood Dataset [24], Indoor Places [25], MASATI [26], Raghavender Sahdev Places [27], SEMFIRE [28], Trimbot [29], NYU Depth V2 [30] and sections from UPenn Natural Image Database [31]. Below we are going to showcase each of those databases with a few of the images that they are containing.

Starting with the DAVIS database featuring different objects in images cut from a video object segmentation point of view. We're using a part of it in the face of objects like a bus, as seen in Figure 10 [23], as well as, a car, dancing, kitesurfing, and a dog built from different scenes:



**Figure 10.** *Bus objects from two different scenes from the DAVIS dataset [23]*

Afterwards, in Figure 11 we can see images from the EU Flood Dataset. It consists of images of floods that happened around the European Union in May and June 2013, used in the context of interactive content-based image retrieval [24]. The algorithms that we utilise in this project are going to have a challenge in distinguishing rather similar photos with large amounts of water in them.





**Figure 11.** Floods from the EU from the EU flood dataset [24]

In Figure 12, we see elements from the Indoor Places dataset, which is part of a Visual Place Recognition experiment, and it features generic photos of rooms, labs, and corridors [25]. The paper that comes with the dataset also offers a comprehensive survey covering different aspects of place recognition from a deep learning perspective.



**Figure 12.** Photos from Indoor Places dataset [25]

The MASATI dataset seen in Figure 13 comprises optical aerial images of maritime scenes from the visible spectrum, which provides colour images of dynamic marine environments suitable for evaluating ship detection methods [26]. Each image in the dataset may depict one or multiple targets under varying weather conditions. It is split into seven distinct classes, including land, coast, sea, ship, multi, coast-ship, and detail.





**Figure 13.** Coast with ships coming from the MASATI v2 database [26]

The Raghavender Sahdev Places dataset and the paper focus on the idea of enabling robots to conduct visual place recognition and categorization [27]. The proposed system leverages experience to enable the recognition of previously observed places in known environments and the categorization of previously unseen places in new environments. It comprises mainly indoor images, as seen in Figure 14.



**Figure 14.** A bedroom and the inside of the laboratory from the Raghavender Sahdev Places dataset. [27]

Another database that is eclipsed in our dataset is SEMFIRE - Safety, Exploration, and Maintenance of Forests with Ecological Robotics [28]. It was created by the Institute of Systems and Robotics at the University of Coimbra team for use in semantic segmentation and data augmentation with a focus on forestry scenes. Figure 15 presents some of those forestry photos.



**Figure 15.** Forestry images from SEMFIRE [28]

In Figure 16, we can see images from the Trimbot2020 Dataset for Garden Navigation, which is a collection of sensor data captured from a robotic platform equipped with cameras and other sensors, as well as external sensors that are used to capture data from the garden environment [29].



**Figure 16.** Garden images from the Trimbot [29]

The NYU-Depth V2 dataset consists of video sequences recorded by RGB and Depth cameras from the Microsoft Kinect in various indoor scenes. The dataset is divided into two parts: the Raw Dataset, which contains all video sequences, and the labelled dataset, which is a subset of the raw one. The labelled dataset includes pairs of synchronised RGB and Depth frames that have been densely labelled [30]. Examples of images from this database are presented in Figure 17 below.





**Figure 17.** *Living room and Kitchen from an apartment in NYU [30]*

The UPenn Natural Image Database consists of approximately 4000 natural scenes from the baboon habitat in Botswana, preprocessed into a suitable format for computer vision research, while also including details about the processing and acquiring of those images [31]. Figure 18 shows 2 of the folder elements that are going to be used in our dataset.



**Figure 18.** *Natural images used from the UPenn database [31]*

All of those databases are suitable for Computer Vision research and provide the needed imagery for comparison and scene detection, as they feature multiple of the same images, but taken from different positions and scenes. This way we can construct the algorithms around them and compare them successfully if they match or not.

## 4.2. Data Structuring and Preparation

Before starting with the approaches and the right methodology, we will need to sort out the datasets and the imagery all of the algorithms are going to work on. Data Preparation is one of the most important parts of most machine-learning projects. Before any data can be analysed and deduced into results, it must be organised into an appropriate form. This is why Data Preparation is being introduced, it is the process of manipulating and organising data prior to analysis. It presents an iterative process of converting raw, unstructured, and messy data into a more structured and useful form that can be investigated. Usually, this involves several major activities, including data profiling, cleansing, integration, and transformation. [32]. Different learning systems have various specific requirements for data processing, so

data has to be transformed to fulfil those requirements first. Generally, the philosophy of data preparation and structuring is to reveal the unknown underlying structure of the problem to enable learning algorithms to work efficiently [33].

For this purpose, an artificial dataset would need to be created, composed of all of the databases mentioned in the Data section above this one. This way a controlled environment is created, with an amount of overlap, which is adjustable, where less overlap means less accuracy. Thus, it can be seen and distinguished between what is an easy, medium or difficult scene match in terms of the overlap between them and ideally what changes from one scene to the next. The use of lots of different datasets allows for cherry-picking various images from multiple datasets from a variety of scenes ensuring no overlap between them at all in the database so that they are completely separate. Afterwards, a random subsection of the image is selected, in the form of a box, though it can be rotated, shifted and sheared, and extracted into a scene. Multiple of those boxes were done per image so that the amount of images is enough. 1000 of those randomly selected sub-images have to be extracted, and each should be compared for the amount of overlap they have with the initial ones - from 80% to 100% overlap were put into the easy category, 40% to 80% were the medium ones, and with anything less than 40% being put into the difficult section. The reason it is done that way is to have a very controlled scenario, where the difference and overlap are exactly known, thus the expectations with these images are familiarised. This is a significant positive, however, it comes with the negatives as well, since if we take for instance image1 and we know it is linked to overlapping image2, and we have overlapping image2 linked to another image3, it is unknown if image1 and image3 are linked, as there's no logic present. Since we have no information if they are linked or not, nothing is stated about them.

Accordingly, in this case, the raw dataset was transformed into Medium and Difficult folders, each representing the level of difficulty of matching the photos inside. Additionally, an Indexes folder was presented that would specify the relations of the images in the difficulty folders. Inside those .txt files, each line or row of text would be a pair of related images. That's how it's known which of them have any initial relation or not. The structure is the following:

Medium_indexes	
Connection1	Connection2
\European_flood_dataset\1\EF_1_25441112.jpg	\European_flood_dataset\1\EF_1_25441113.jpg
\European_flood_dataset\1\EF_1_25441113.jpg	\European_flood_dataset\1\EF_1_25441114.jpg
\European_flood_dataset\2\EF_2_26454139.jpg	\European_flood_dataset\2\EF_2_26454143.jpg
\European_flood_dataset\3\EF_3_26455230.jpg	\European_flood_dataset\3\EF_3_26455618.jpg
\European_flood_dataset\9\EF_9_26458017.jpg	\European_flood_dataset\9\EF_9_26458038.jpg
\European_flood_dataset\10\EF_10_26458129.jpg	\European_flood_dataset\10\EF_10_26458133.jpg
\European_flood_dataset\11\EF_11_26458613.jpg	\European_flood_dataset\11\EF_11_26458736.jpg
\European_flood_dataset\12\EF_12_26459046.jpg	\European_flood_dataset\12\EF_12_26459061.jpg

*Table 1. Medium indexes table representing the two connections with a relationship*





It is accepted that we are going through the medium relation pictures first, and the difficult ones after, as it will be significantly easier after the first try. The aim of the data preparation here is to have 10 different training and testing interactions or in other words, process n cross-validation sets. Specifically, the process we choose to do is named K-Fold Cross Validation. It is perfect for our use case because it finesses the problem of not having enough data, by only using a portion of the available data to train the model and another portion to test it [34]. To implement this, the data is divided into K roughly equal-sized parts. For instance, if our K is equal to 5, the process involved the following scenario:

1	2	3	4	5
Train	Train	Validation	Train	Train

**Table 2.** Equally divided proportions of data sets using K-Fold Cross Validation [34]

In this case, for  $K = 5$ , the expected error can be estimated. Whenever the k in this cross-validation method increases, it is found to have a lower bias and less error, but higher variance. That's why we use it, we want to get as accurate results as possible.

In our case, in order to implement this K-Fold Cross Validation method, we put k to be 10 and randomly select 100 images and split them aside for Test Set 1 - taking care to ensure that it is entirely contained according to the relationships in the index.txt file. This means that if one image is in the test set then any image that it's linked to should be in that set, as well. The next step is to choose another 100 random images for Test Set 2. This is done until there are 10 (k) different testing scenarios in the form of their own sets. Afterwards, all the photos that are left are combined to form the training sets. That's how we get the 10 training sets that are needed all around the size of 892 images:

 tti1	100x2 table	 test_relationships_p1.csv
 tti10	91x2 table	 test_relationships_p2.csv
 tti2	101x2 table	 test_relationships_p3.csv
 tti3	100x2 table	 test_relationships_p4.csv
 tti4	100x2 table	 test_relationships_p5.csv
 tti5	100x2 table	 test_relationships_p6.csv
 tti6	100x2 table	 test_relationships_p7.csv
 tti7	100x2 table	 test_relationships_p8.csv
 tti8	100x2 table	 test_relationships_p9.csv
 tti9	100x2 table	 test_relationships_p10.csv
 ttr1	892x2 table	 train_relationships_p1.csv
 ttr10	901x2 table	 train_relationships_p2.csv
 ttr2	891x2 table	 train_relationships_p3.csv
 ttr3	892x2 table	 train_relationships_p4.csv
 ttr4	892x2 table	 train_relationships_p5.csv
 ttr5	892x2 table	 train_relationships_p6.csv
 ttr6	892x2 table	 train_relationships_p7.csv
 ttr7	892x2 table	 train_relationships_p8.csv
 ttr8	892x2 table	 train_relationships_p9.csv
 ttr9	892x2 table	 train_relationships_p10.csv

**Table 3.** Data sets after construction in MATLAB and .csv format

Additionally, what can be done is to get the no-match images - 5 of them are needed, separately from 100 columns of images with no matches, chosen in random order, but taken care that the no-match pairs cannot be linked at all. This means that if image1 is linked to image2 and image2 to image3, then image1 and image3 cannot be used as a no-match pair and that the no-match pairs are entirely contained within their training and testing datasets.

Now that we have the testing and training sets, what follows is reforming the structure of the data so that it can be used by the code for the Siamese Networks and SIFT. There is a certain requirement that has to be met before the code that does the actual Siamese Networks can be executed. The code uses a specific structure when it inputs the images, done in the following way: `"../input/train/F00002/MID1/P0001_face1.jpg"`. It introduces a hierarchical structure of *families/subfamilies/image* as a way to link different images and subfolders within families. We don't have that structure yet, so we have to create it ourselves. It resembles a classic Bottom-up approach for clustering those images into the folders. For each image, we are going to check out its label, and the corresponding relationships to it. The aim is to identify every other image that is related to the first one, find out its label and change it to be the same label as the first image we take. This basically says that they are all matched now. This doesn't happen immediately, as an iterative approach is needed, so we first take a snapshot of the current image labels and then go one image after the other, until the end. Once we've iterated over all of them, we change the labels. We pass through all of the images however times are needed until all the relationships have been found and all the labels have been successfully changed, removing any conflicts. This was the complex part because it was hard identifying all the labels and managing to change them according to their relationships.

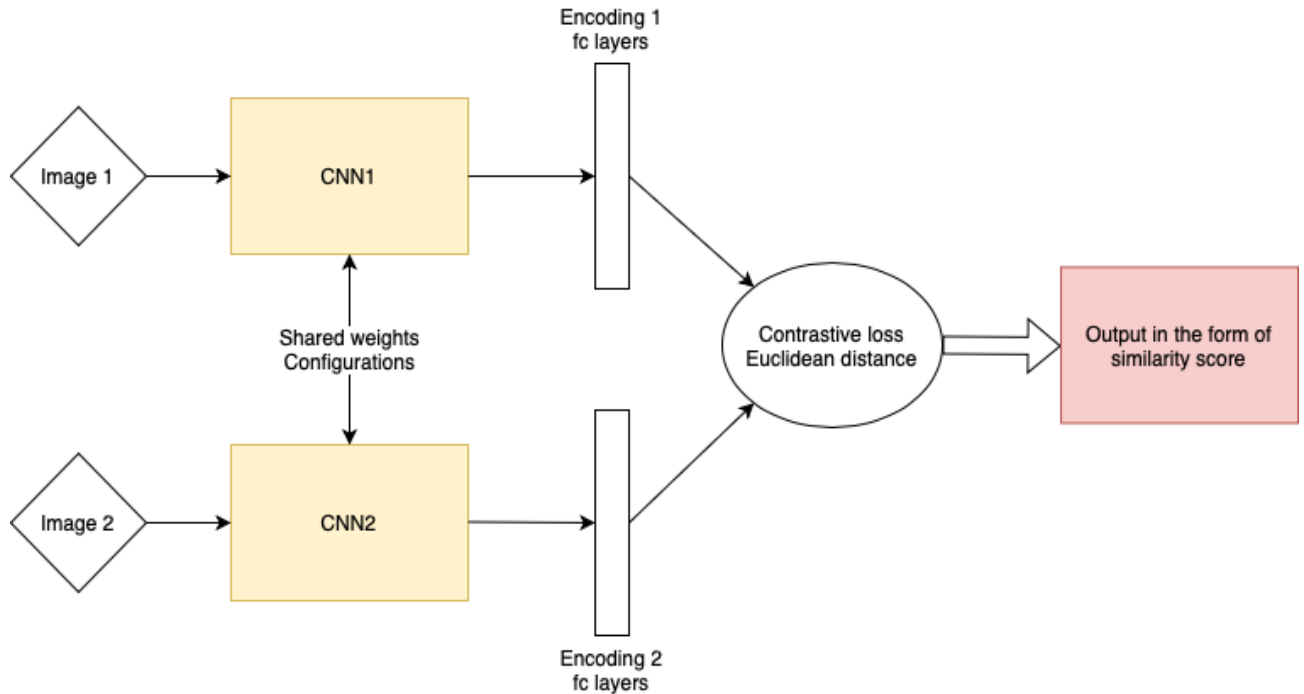


Now that the images were linked with their labels, what was left was to create the families and subfolders and put in the photos accordingly. The way this was represented in the starting template code was by having only one family for validation - in the form of "F09xx". They are relying on the assumption that you can identify related family members, by just taking random pictures, which is extremely experimental and unreasonable. Their validation set is too small, so what we've done in our project was change this specifically because we want to identify training, validation and testing families. This is the right way to get the correct results and accuracy from our Siamese Networks model. From here, we randomly choose 60% of the families and designate them to be from the training set. From the remaining 40%, half of them are randomly chosen to go for validation and then everything else that's left goes for testing. This is done by introducing random permutation to the structuring code. Random permutation here would be taking all of the in-depth indices from one up to the max number there are and then scrambling them so that we are sure the percentage of taken images in all of the sets is completely randomised. The "F09xx" form for the families is discarded and, in its place, we introduce 'TrF' to represent the training sets, 'VaF' for the validation sets and 'TeF' for the testing sets. This way the separation of data and families is way more explicit and makes more sense. Then we only copy the images into their corresponding folders and the data reconstruction is basically complete. An important mention should be given to the adversarial relationships, which are the ones where there's no match for every training image that we use. For them, a no-match will be created at random by picking something from a different family, as that's the only guarantee that there's no match at all. Now that this is ready, we have met the requirements and can look more into the Siamese Networks and SIFT themselves, going through the neural networks, the code and what happens in them.

Various baseline approaches for scene comparison exist, however, the ones that we're going to use are the SIFT approach, the Object Classification approach which would be seen from the sphere of Siamese Networks and eventually Class Activation Maps (CAM). However, CAM is not the main focus of this paper, so it is going to be touched on sparingly.

### 4.3. Siamese Networks

The main problem in this project is one of image similarity. For this purpose, the general approach that is going to be used is Siamese Networks. A Siamese Network is an architecture with two similar parallel neural networks or CNNs used to learn useful data descriptors for comparing subnetwork inputs [35][36].



**Figure 19.** *Siamese Networks Architecture*

As seen in Figure 19, both of the networks share the same weights and configurations, with different inputs in the form of images being fed into each network. Each of those networks is composed of convolutional layers, rectified linear units (ReLU) as non-linearity for the convolutional layers, and fully connected layers [37]. During training, those input images have been fed to the networks, its main objective is to learn optimal feature representations of input pairs, such that similar images in a pair are brought closer together, while dissimilar images are pushed farther apart [37]. The outputs of the two networks are then put through a loss layer and combined to generate predictions.

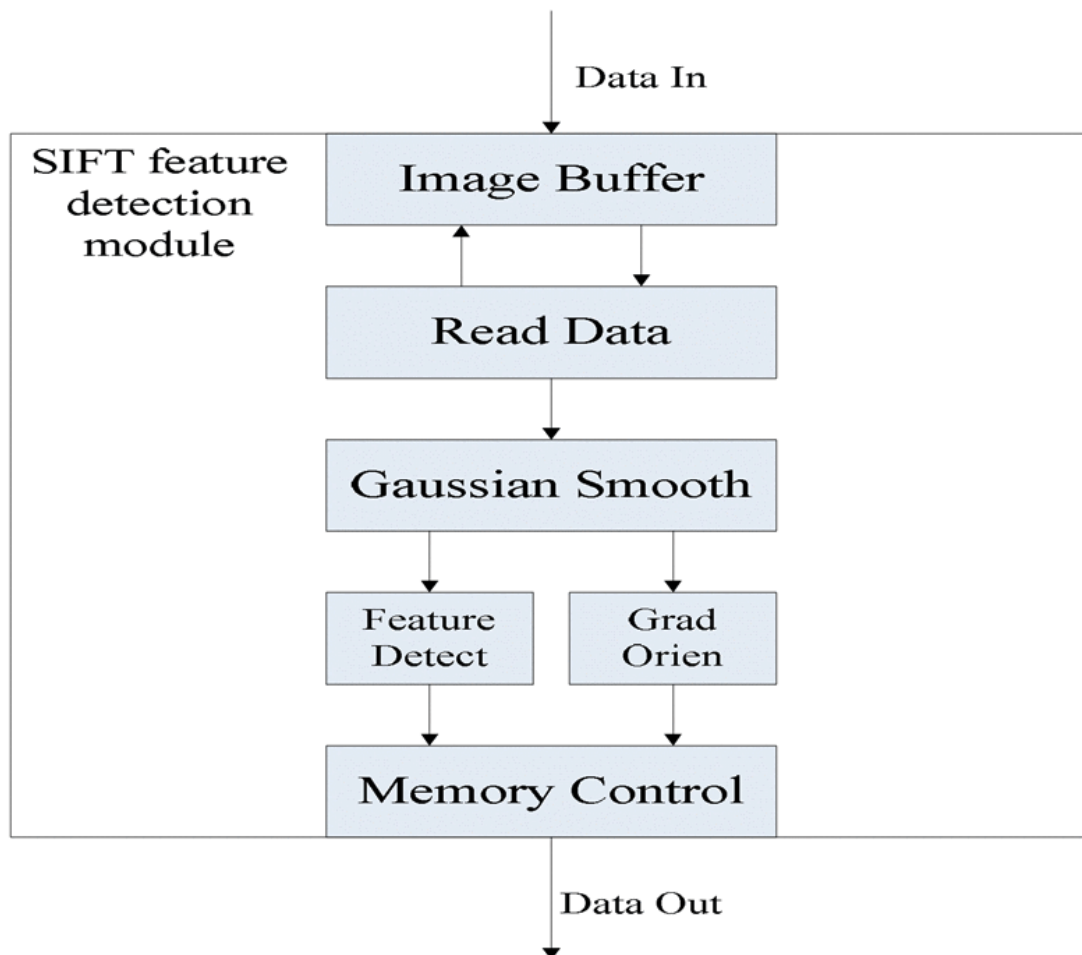
The loss layer generally tries to minimise the squared Euclidean distance between the features of the positively matched pair of images and maximise the distance of negative no-match pairs. This is also usually called a “contrastive loss” - positive pairs tend to be pulled closer to each other, while negative ones are further away [37]. This contrastive loss is used to train models with a weaker form of supervision, as only the knowledge of whether the pairs are positive or negative is required. However, these models can be considered as holistic solutions since they focus on absolute distances rather than relative distances, which depending on the project may be more critical in some cases [36]. There are other types of losses used in CNNs, contrastive loss is mostly and generally used in facial recognition tasks, while a more commonly used loss function would be the cross-entropy loss [38]. It measures the difference between the probability distribution obtained from the current training and the actual distribution. The cross-entropy loss is also known as the softmax loss, and it is exactly what is used in both our Siamese Network experiment, as we use ResNet and other popular CNN architectures, like AlexNet [19]. What it does is compute the penalty value in the form of a logarithmic combination based on the distance between the predicted probability and the actual output - 0 or 1 for each comparison [38].



The main template of code that we're going to use and build upon represents a Siamese Network just like everything that was mentioned above, written in PyTorch and utilises ResNet50 of fastai models [39]. Under certain augmentations on my part, the code provides multiple files of spreadsheets and data, while also including graphs, representing the actual output of a standard Siamese Network, all of which are going to be presented in Chapter 5 of our report.

#### 4.4. Scale Invariant Features Transform (SIFT)

Scale Invariant Features Transform (SIFT) is a rather new approach created by David G. Lowe in 2004 [40] that has gained a lot of traction lately as a very robust and distinctive comparative mechanism able to find similarities and collations, with feature extracting and matching while being invariant to scale and rotation [41][42].



**Figure 20.** Data Flow in SIFT Feature Detection Module [43]

In Figure 20, we can see a part of SIFT's architecture and more specifically its Data Flow. The method can detect more general duplications between scenes based on the image SIFT features. This is done in a few steps, starting with identifying and detecting key points in an image and computing the local maximum and minimum points invariant to scale and rotation, which are also called SIFT features. The next step is to localise all feature points in the

image, create a detailed model, get rid of all outliers and not stable points and only after that match all key points with their closest correspondence in the image. Afterwards, a feature descriptor is generated by sampling image gradient magnitudes and orientations across all of the detected vital points and placing them inside an array of orientation histograms in a piece around the designated points. Gradients are measured with scale invariance and all entries from all histograms are put in a 128-dimensional vector to create the feature descriptor, with which later on we process similarity between images. This is a more traditional approach, which is very demonstrable, but with rather worse results compared to the other ones we're going to use. The point is to visualise the similarities between images and successfully map them on top of each other at the end so it can be seen which parts of them are the breaking points [41]. Siamese neural networks essentially upgrade on top of the Sift Approach, as it operates in pairs together on two different input vectors to compute comparable output vectors, easily being able to make computations between two images and give a result if they are similar or not [35].

## 4.5. Class Activation Maps

Similarly, CAM can deduct for you if two images are similar or not, however, it doesn't stop there and goes deeper into its analysis of the image, highlighting with thermal imagery the regions and parts that have the biggest similarities. This significantly helps with learning the networks and eases up debugging, as it also results in object localization without manually labelling the box bordering the object [44].

One method by itself is not enough, for instance, CAM tells you about the influence but doesn't look into the corresponding regions, while SIFT investigates correspondence between scenes in a better way. That's why we want to use all 3 approaches and prove demonstrability and explainability in our results.

## 4.6. Evaluation

From the beginning of the project, we've known what the final aims and objectives that we have to meet are. We have also known how we are going to evaluate the results and present them in a way that supports our hypothesis and final outcome. The purpose of introducing the concepts of Siamese Networks and Scale Invariant Features Transform is to successfully compare scenes and identify objects from the images we analyse. We need to be able to do that in order to create a model that is able to recognise the objects from different images and compare them to real-life cases and scenarios, such as successfully helping court cases where a criminal is visible partly or in a blurred quality in the form of a photo.

In terms of the Siamese network, there are 2 main methods of evaluation of the scene comparison we are executing - the first one would be a binary comparison, simply telling us if the images we are comparing are the same scene or not. The second one would be having results as a lineup - if we are interested in 1 set, we gather a few candidate scenes from our results database and determine from the lineup we create which one is the most similar to our target image, making an assumption our image is present in the lineup. This can also be done

in a top k way, where if k is 5, we find the 5 more similar ones if the matching image is in this list of images. In a real-life scenario, if we have a picture of a suspect, we won't go through thousands of images in a database, but rather through a condensed list of 10 or 15 photos.

Having that in mind, the way we use SIFT is as a backup for our Siamese Network, we use it to help identify and evaluate where Siamese goes wrong and visualise correctly why and how it is wrong. SIFT provides us with more demonstrable results, which we happily utilize as much as possible.

## 5. Results

### 5.1. Outcome of Models

After constructing and preparing our data in the right way, grouping it in families, we can now finally run our Siamese Networks algorithm and present the results we get from it. We get multiple things when the Networks finish running, mainly collecting data in the form of .csv files. The two main .csv files that are crucial for our project are the eval\_history file, which shows us the pairs of images that we compare, the probability that they are similar, the result that we get from the output of the Siamese, and the ground truth, showing us if they are actually a match, while the other important file would be the overall results of the model, where we get the accuracy and loss of both the training and the validation. Those two files are sufficient in showing us the needed outcome, thus resulting in a valuable discussion where we conclude what those numbers actually mean and represent. The SIFT approach is used as a supplement to the Siamese Networks, outlining if and why they get something wrong.

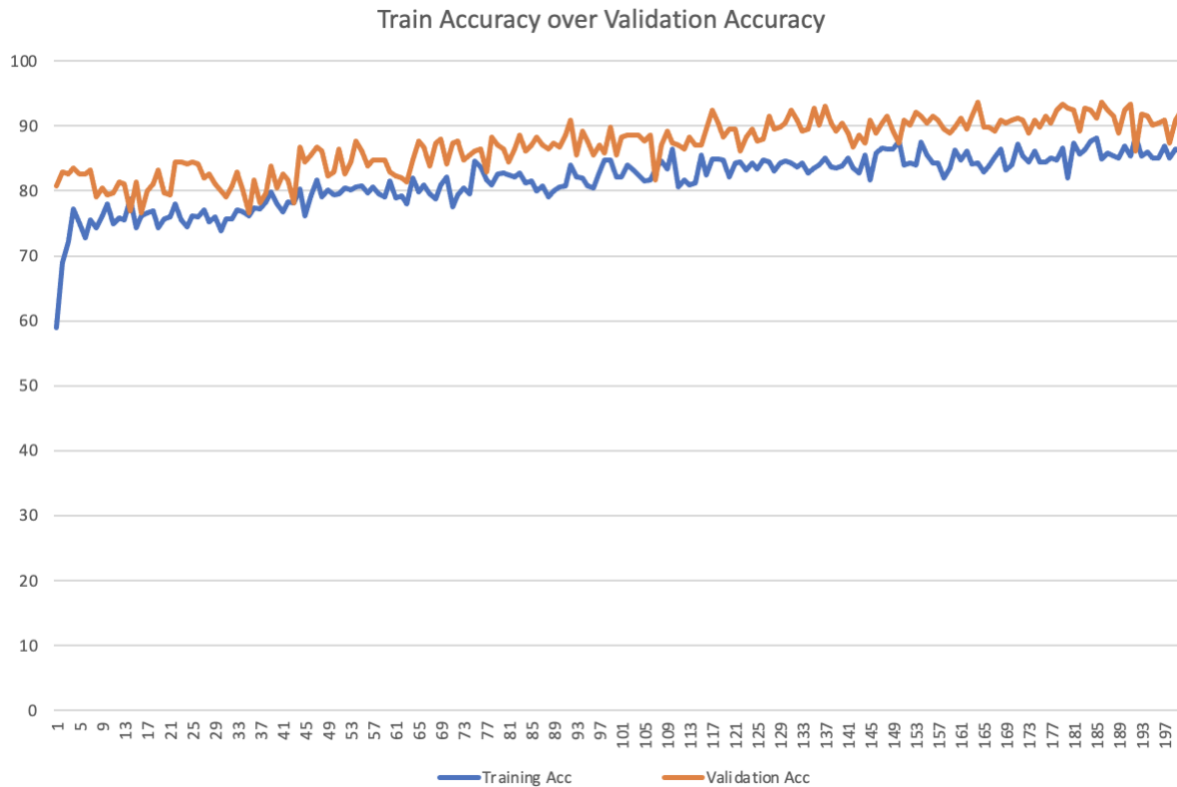
We're going to start by presenting the results from the Siamese Networks model first and then going over to individual matching pairs. This way a clearer picture is presented for the model as a whole, and it can be used in an easier way to analyse particular matching images later on. The way this model is built and created is for the algorithm to be executed a certain number of times, also named epochs, and for each epoch, we calculate the training accuracy and loss, and the validation accuracy and loss, giving us an insight into how the model has developed over the epochs. This methodology allows us to view the accuracy and validation of the model, presented in a table format in Table 4:

epoch	train_acc	train_loss	val_acc	val_loss
0	59.02173913	0.665188277	80.69620253	0.518388563
1	68.91304348	0.595242679	82.91139241	0.479637911
2	72.17391304	0.582510722	82.59493671	0.498040079
3	77.17391304	0.546691837	83.5443038	0.446280783
4	75.10869565	0.539677696	82.59493671	0.440063881
5	72.82608696	0.559916544	82.59493671	0.44810808
6	75.54347826	0.537232859	83.2278481	0.426444177
7	74.23913043	0.525849726	79.11392405	0.455333849
8	76.08695652	0.524296546	80.37974684	0.43140694
9	78.04347826	0.505511882	79.43037975	0.491114897
10	74.89130435	0.520166345	79.74683544	0.452925781

*Table 4. Results of the first 10 epochs in a table format*

However, a lot of data and numbers like that are hard to visualise and present, as we have information about all of the 200 epochs, so in order to demonstrate the model's characteristics more clearly, we can visualise the data in a graph. If we take the training and

validation accuracy, a valid interpretation of the model's accuracy can be made later on. The needed graph for that is presented in Figure 21:



**Figure 21.** Graph of training over validation accuracy

## 5.2. Image Comparison and Error measures

Having the required tables and figures for the model of the Siamese Network, we follow the implementation of the evaluation and now show the evaluation history of the model we use, meaning its process of going through all the pairs of images and distinguishing if they are a match or no match. The file in which we output the results that are needed is built in the following format, shown in Table 5:

	img1	img2	prob	result	ground truth
0	('data/test/TeF0001/EF_100_26502599.png',)	('data/test/TeF0001/EF_100_26502599.png',)	0.9718733429908752	TRUE	1.0
1	('data/test/TeF0001/EF_100_26502600.png',)	('data/test/TeF0114/EF_58_26471027.png',)	0.926079511642456	TRUE	0.0
2	('data/test/TeF0002/EF_102_26503072.png',)	('data/test/TeF0002/EF_102_26503072.png',)	0.9584039449691772	TRUE	1.0
3	('data/test/TeF0002/EF_102_26503074.png',)	('data/test/TeF0002/EF_102_26503074.png',)	0.9792018532752991	TRUE	1.0
4	('data/test/TeF0002/EF_102_26503084.png',)	('data/test/TeF0002/EF_102_26503074.png',)	0.9807242155075073	TRUE	1.0
5	('data/test/TeF0004/EF_106_26504205.png',)	('data/test/TeF0004/EF_106_26504205.png',)	0.9836217761039734	TRUE	1.0
6	('data/test/TeF0004/EF_106_26504208.png',)	('data/test/TeF0004/EF_106_26504205.png',)	0.9772144556045532	TRUE	1.0
7	('data/test/TeF0009/EF_114_26505776.png',)	('data/test/TeF0009/EF_114_26505776.png',)	0.9749109745025635	TRUE	1.0
8	('data/test/TeF0009/EF_114_26505778.png',)	('data/test/TeF0009/EF_114_26505779.png',)	0.9610742926597595	TRUE	1.0
9	('data/test/TeF0009/EF_114_26505779.png',)	('data/test/TeF0615/TV_hug_0014_060.png',)	0.009873994626104832	FALSE	0.0
10	('data/test/TeF0010/EF_116_26513966.png',)	('data/test/TeF0010/EF_116_26513969.png',)	0.88742595911026	TRUE	1.0
11	('data/test/TeF0010/EF_116_26513967.png',)	('data/test/TeF0313/NYU_dining_room_0030_2.f	0.01242313627153635	FALSE	0.0

**Table 5.** Results for pair of images in the process of evaluation in the model

In this table, “img1” and “img2 ” represent image 1 and 2 filenames, in their fully constructed order including the families in the path. In the “prob” field there is the probability of a match

resulting from the model - closer to 1 means more likely to be a match, while closer to 0 is less likely to be a match. In the “result” field there is the binarization of the “prob”, where it is ceiled to TRUE if the probability is  $\geq 0.5$  and floored to FALSE if it is  $\leq 0.5$ . The ground truth represents the actual nature of the relationship between the images, being the official correct label if the images are a match or not, respectively 1.0 being a match and 0.0 a no-match. From those results, there are 4 possible scenarios between the pairs of images that are compared:

- True Positive - where the images are a match, and we conclude it is the same scene.
- True Negatives - where the images are not a match, and they are totally different scenes.
- False Positives - where the Siamese Networks conclude it is a match, however, it really is not, as the images are not similar.
- False Negatives - where the Siamese Networks conclude it is not a match, however, it is a match according to the ground truth.

In those scenarios, the first two are right, while the latter two are wrong. For all of them, samples are going to be shown and discussed why they are in one of the 4 categories.

### **True Positive**

At random, 6 pairs of images are going to be picked to show each of the scenarios. Starting with the main one, which is a True Positive. The images in this one are a match both as a result of the Siamese Network and in their official relationship, as seen in Figure 22 in the form of a 3 x 2 matrix:

## Matched Pairs

cls1=TeF0013 conf=0.97 cls2=TeF0013



Comparison 1

cls1=TeF0315 conf=0.93 cls2=TeF0315



Comparison 2

cls1=TeF0004 conf=0.98 cls2=TeF0004



Comparison 3

cls1=TeF0576 conf=0.96 cls2=TeF0576



Comparison 4

cls1=TeF0363 conf=0.88 cls2=TeF0363



Comparison 5

cls1=TeF0082 conf=0.93 cls2=TeF0082



Comparison 6

**Figure 22.** 6 comparisons of positively matched pair of images, all of them represent the same scene but from different angles; Comparison 1 - Bridge over water; 2 - Living room with sofas; 3 - Village near a lake; 4 - A person in motion; 5 - Kitchen; 6 - Highway bridges

All of the comparisons have a probability over 0.5, which puts them in the category of positive matches.

### True Negatives

In this scenario, the images have to successfully be different by nature and as a result of the algorithm. Figure 23 represents exactly that:

## No Match Pairs

cls1=TeF0013 conf=0.04 cls2=TeF0444



Comparison 1

cls1=TeF0132 conf=0.02 cls2=TeF0580



Comparison 2

cls1=TeF0073 conf=0.03 cls2=TeF0243



Comparison 3

cls1=TeF0663 conf=0.02 cls2=TeF0315



Comparison 4

cls1=TeF0023 conf=0.01 cls2=TeF0225



Comparison 5

cls1=TeF0191 conf=0.01 cls2=TeF0167



Comparison 6

**Figure 23.** 6 comparisons of positively no matched pair of images, all of them represent completely different scenes; Comparison 1 - Bridge and a living room; 2 - Flooding and social interaction; 3 - A dam and a room; 4 - A kiss and a living room; 5 - Tunnel and a room with a bed; 6 - Bathroom and a road

## False Positives

This scenario is far more interesting. It shows an error in the Siamese Networks, where it was concluded it is a match, however, the ground truth shows us that it is not, resulting in a disagreement between the two. Examples of this scenario are shown in Figures 24 and 25:



cls1=TeF0001 conf=0.93 cls2=TeF0114



**Figure 24.** A comparison between flooding at a dam and under a bridge

cls1=TeF0334 conf=0.58 cls2=TeF0444



**Figure 25.** A second comparison between a study room and a living room

### False Negatives

Just like the False Positives, this one is also an error in the Siamese Networks, with this one stating that they are not similar when the truth is that they actually are. Figures 26 and 27 visualise this rather rare scenario:

cls1=TeF0451 conf=0.49 cls2=TeF0451



**Figure 26.** Two images from the same room, but different angles

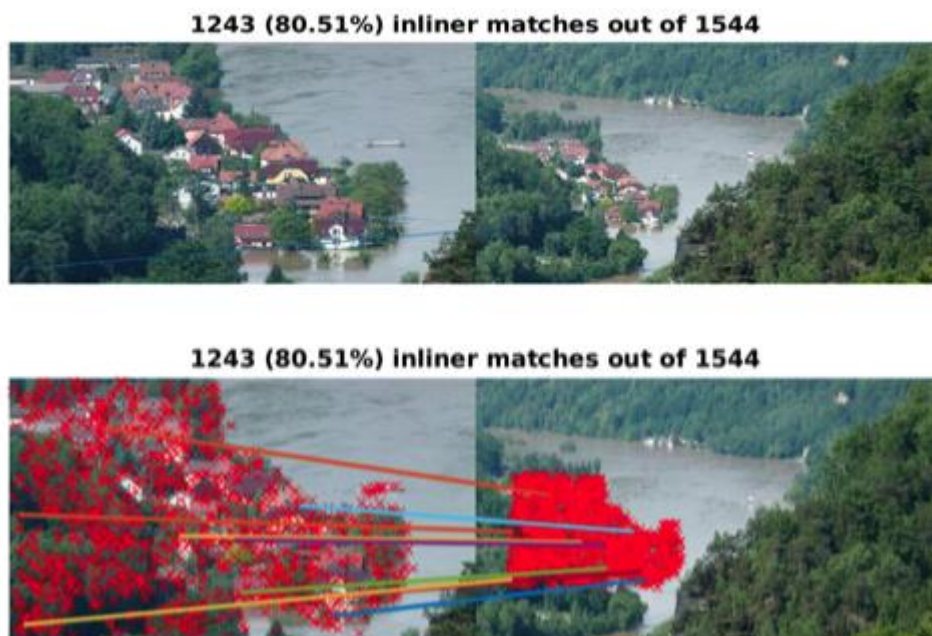
cls1=TeF0225 conf=0.38 cls2=TeF0225



**Figure 27.** *A bedroom from two different angles*

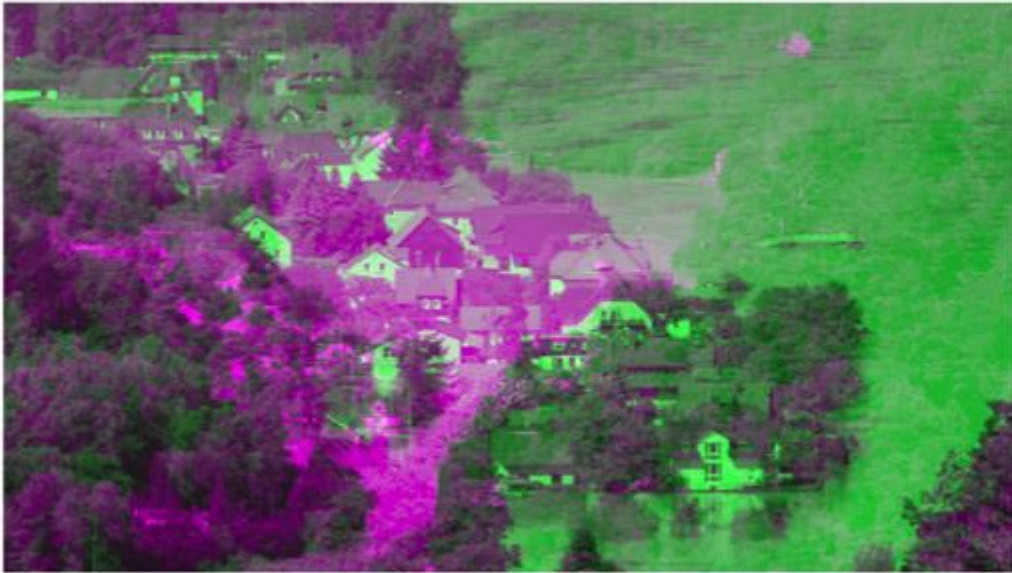
### Scale Invariant Features Transform

From here on, as it was said in the Evaluation section, we can use the SIFT approach to outline any errors that the Siamese Neural Networks make. It is applied as a support to them, coming in and helping whenever there is a mistake. Specifically, it is being utilized on the False Positives and Negatives, as it can pinpoint the exact confusing points, which algorithms recognize and fall for. In a normal scenario, such as for a matched pair of images, SIFT would give us the following results:

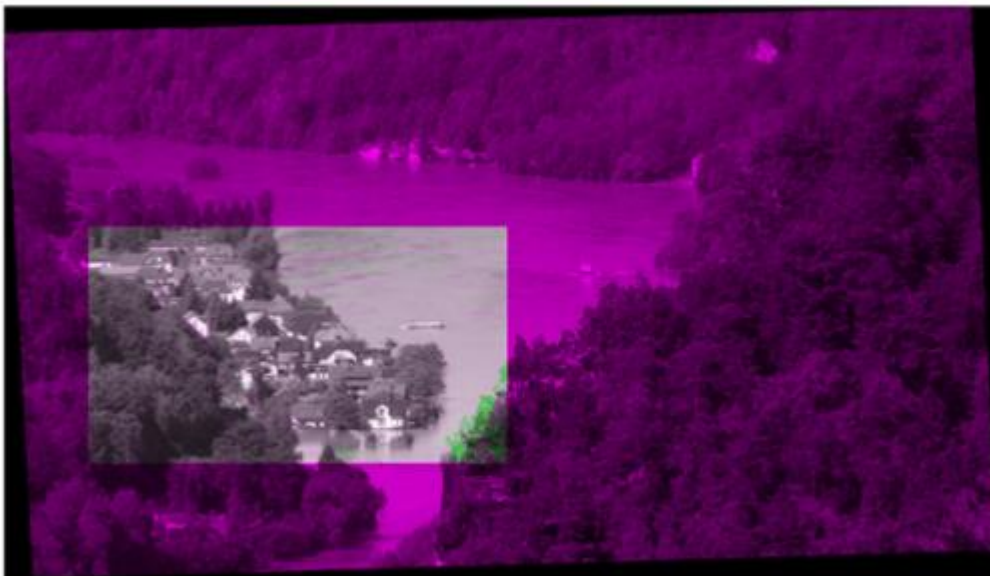


**Figure 28.** *Pair of images in comparison utilising the SIFT approach*

In Figure 28, it can easily be seen why exactly these two images match. This is why we provide SIFT in addition to the Siamese Networks. Figures 29 and 30 show how exactly SIFT maps the compared photos on top of each other:



**Figure 29.** *The initialising layering of images on top of each other*



**Figure 30.** *Finalised layering of the compared images*

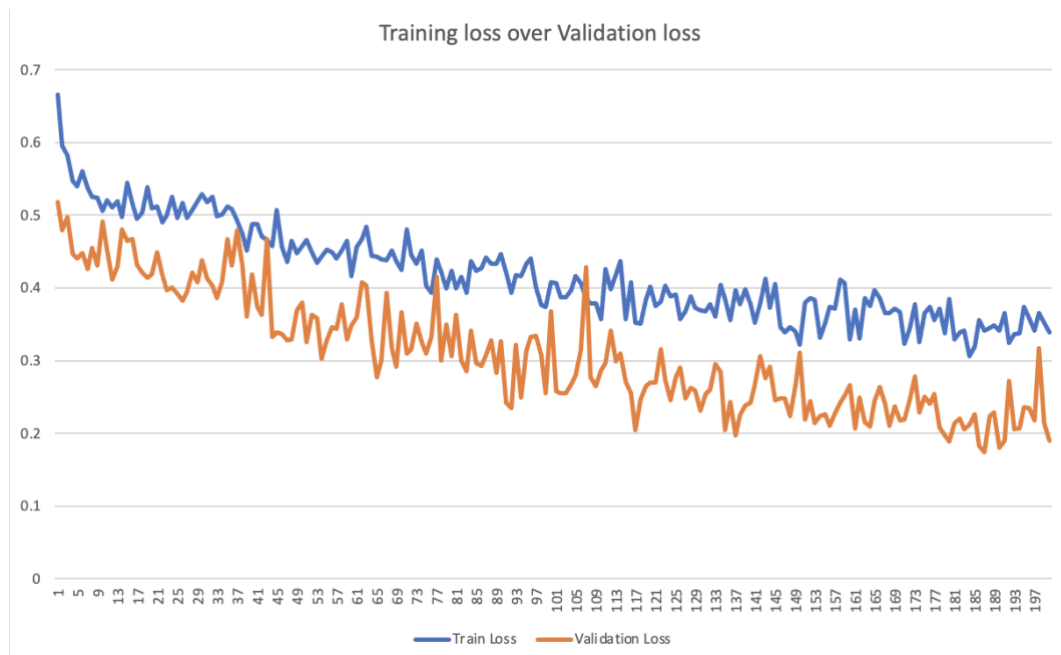
This kind of visualisation in SIFT is done for all of the available pairs of photos that can be compared. It provides us with a way of justifying images as being a match or no match, as well as, giving the exact unique feature points that are used to distinguish what parts are similar. The straight lines in Figure 28 show us those feature points being linked between the images, making a decision based on the direction, presenting SIFT as an explainability concept.

In the next section, discussions about the results are going to take place, evaluating them to the aims, while also providing quantitative and qualitative analysis for the best possible interpretation of the outcomes we've got from the project.

## 6. Discussion

The importance and significance of the results do not come from only presenting them, but thoroughly analysing what they mean and what the essential takeaway the project gets from them is. Here, we refer to most of the figures illustrated in the Results section, while also supplying more material to reinforce our statements, in addition to relating them to the aims and the purpose of the report.

Starting with the model of the Siamese Neural Network in Table 4 and Figure 21, we are presented with the results of 200 epochs, having the training and validation accuracies and losses for each one. From it, we can deduct a few things. The first and most essential one is that our model is both training and validating well. It starts with mediocre values, and it has an upwards stream of improvement until the latter stages of the progression through the epochs. We can instantly see in Table 4 how it starts with 59.0% of accuracy, which is also the lowest training accuracy value in the model but jumps in quickly to 68.9% with the second epoch, showing the massive headway it goes through at the start. From then on the steady advancement continued, reaching around  $\approx 86\%$  of training accuracy at the end of the 200 epochs, with a maximum training accuracy of 88.4% in epoch 191. The validation must follow the flow of the training, as this ensures our model is not only getting trained properly but tested decently as well, finding the optimal model with the best performance. The bigger the validation, the more accurate the training model is. Following this upwards progress, it starts at 80.7% validation accuracy, rising to 93.7% at the most in epoch 163, to fall a bit to  $\approx 91\%$  of validation accuracy at the end. It is good that it goes up and down all the time, ultimately rising to a decent percentage, which means that it reacts well to being fed new data. Another evidence that the model is working in a good manner is presented below in Figure 31:



**Figure 31.** Training loss compared to Validation loss of algorithm model

In the figure, both the training loss and the validation loss are steadily going in a downward trend, showing that there is less and less that the model mistakes or hasn't learned yet. A problem we encountered with the visualisation of this model at first was that the one we created at the beginning was not validating all that well. It had good training accuracy, but it struggled to validate, going over 80% but then rapidly declining, assuring us that it is not a good model for this kind of comparison, and it was only reacting well to initially fed data. Additionally, at first, the model's top values for training and validation accuracy were both situated around epoch 11, which is far too early for it to be deemed trustworthy. When it was reworked, the results were satisfying, concerning our aims as proved above, which is just another reason why testing different and multiple strategies on models is crucial for finding the most efficient one.

Having analysed the behaviour of the overall Siamese Neural Network algorithm so far, we can now indulge in the individual pairs of images and their comparisons and relationships. The logic behind structuring the images into different families was that most photos in the same family would be related in some way. This way we can differentiate between families and easily conclude if they are connected in the first place. In this matter, the Siamese Network is directional. In Figure 22, all the pairs of images are True Positive match couples, as all of them, even though they are from different angles and different subfamilies, have the same main family and are essentially the same scene. If we look at any of the comparisons made in that figure, we can naturally say that this is true, it is easily visible why they are matched. This is now a successful part of the experiment, being sure that we can identify images with a ground truth relationship. Figure 23 shows us the other side of the experiment that is being conducted, and that has to be concluded to be true in order for the Siamese Network algorithm to be completed. In figure 23, the no matched pairs are presented, and similar to the matched ones, it is clear why they are different. In all of them, the scenes that are compared are completely different, netting no more than 0.04% similarity percentage.

This is the concept of Siamese Networks, and specifically one of the aims of this project - to successfully investigate the approach of object classification and scene comparison by deploying Siamese Networks and concluding if 2 photographs are the same scene or not. This concept of classification teaches us how to interpret a single image, which you can then use for image comparison within the neural network. Nevertheless, Siamese Network algorithms are flawed, as in our project this is introduced in the form of False Positive and False Negative pairs of images. In Figures 24 and 25, two examples of False Positives are presented, and after analysing them, we see it as a reasonable action for the algorithm to make such a mistake. In Figure 24, both of the pictures are ones of a flood, even though they are from different families, where turbulent water takes the bigger part of the picture, so if the platform we've built takes into consideration just this box of water - it can be seen why it would mistake it for a positive match. In Figure 25, the case is similar, one of the pictures depicts a study room, with multiple furniture objects, which are also shown in the second picture of a living room. Although the probability percentage is lower in this one, being 58%, it is way more different and distinctive than the first example with the floods. The other problem that's occurred is the False Negatives, where the procedure has mapped them as not



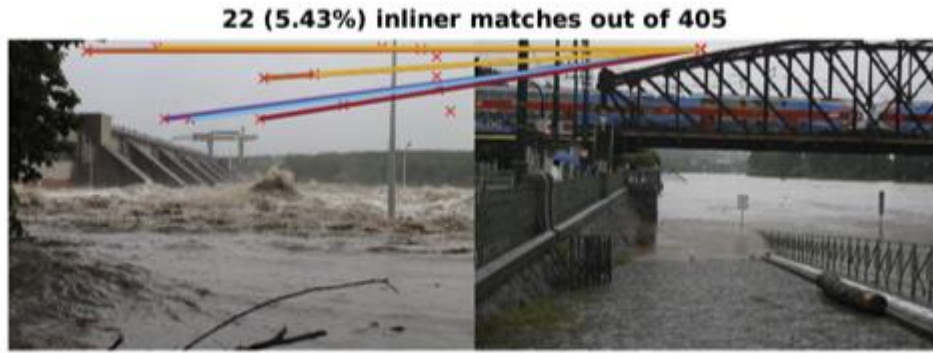
a match when in their nature, they are the same scene, coming from the same families. Both Figures 26 and 27 depict the same scene, but from very different angles, making it hard for the Siamese to catch up on it. In the first one, it is extremely close, with a similarity probability of 49%, only 1% off from it being classified as a match, concurring that there were not enough similarities recognised in the photos for it to be a match. At a first sight, only the carpet really matches. In the second one, the common elements are the bed and the pattern on it, but apparently, this wasn't enough as well, either, netting a 38% similarity. It can be seen here how certain common features influence in a different way in scenes with larger similarities. Quantitatively, there are 24 instances of False Positives out of 314 image couples for comparison, which equates to a  $\approx 7.6\%$  proportion of errors. Compared to the False Negatives, which are only 3 out of 314, resulting in  $\approx 0.9\%$  out of all errors, the first ones are way more frequent. This means that it has a tendency to rather match images that are not similar than to not match similar ones. The project itself is heavily based on CNNs, which although currently is the state of the art, with which valuable results can be introduced, require huge amounts of data to train from, and mainly struggles to generalise and demonstrate.

Thus, in the aftermath of these flaws, the SIFT approach is introduced, and it is utilised as a support for the Siamese, to come in and help them fix the mistakes, as well as identify and make clearer why False Positives and Negatives occur. We can't use either of them on their own, as SIFT has many general problems in terms of accuracy and robustness. Even though it is great for demonstrating and visualising results, it simply is not great for matching different types of images, as looking for unique features in the first photo, which is hard to be found in the second we are comparing the similarities to.



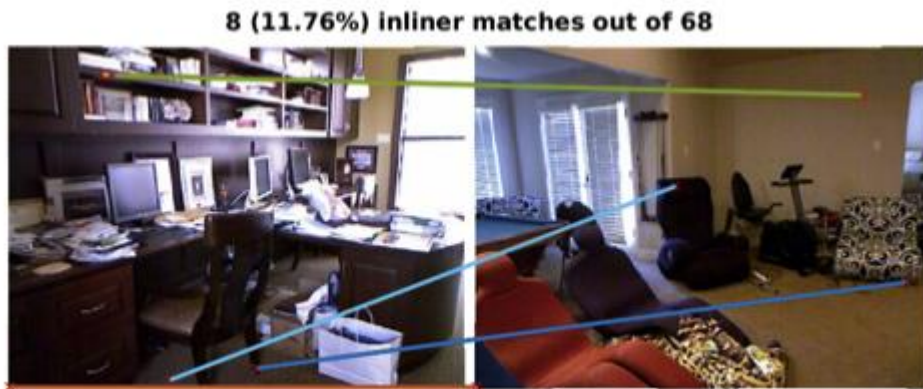
**Figure 32.** Same pair of images from Figure 22 - Comparison 1

In Figure 32, we can see an exact example of why it is not that applicable for different scenes of the same image, we know that the bridge is in the same place, and we have it confirmed by both the Siamese and the ground truth, however, SIFT only finds 8 matches and 1.55% similarity. It basically finds points that should be distinctive and consider those for comparison rather than every single pixel and the image itself. Nevertheless, SIFT does the perfect job in helping us analyse and further solve the false results the Siamese initially get. It is able to tell us and demonstrate to us that the false positives are actually unfamiliar images and have no relationship between them. This is shown in Figure 33:



**Figure 33.** *SIFT visualisation for the first False Positive match*

In Figure 33, we can see the same figure that is our initial False Positive in Figure 24, however, SIFT reveals to us their actual relationship, concluding that they are not in fact related or from the same family with only a 5.43% similarity index. The result of this algorithm is also revealed for the other presented False Positive in Figure 34:



**Figure 34.** *SIFT visualisation for the second False positive match*

In this one, once again we are able to prove that those two images are not a match with their 11.76% similarity index. Then again, allowing us to showcase the results in such a manner, also provides us with the opportunity to see why the Siamese Network would confuse them and get them out as a positive match. In Figure 33, apart from having a large amount of water, SIFT chooses the sky as the biggest feature point, but even though it is the same colour it manages to distinguish them and show there is nothing else that is as similar. In Figure 34, it finds some similar points in the furniture but shows how distinct they are in any other way. This is why SIFT is used, to identify and correct those mistakes.

From here on, the approach that is undertaken is now a hybrid approach - both Siamese Networks and SIFT are used in the sphere of image comparison, just like it was talked about in the aims of the projects. As SIFT is bidirectional, we are now presented with multiple decisions and ways to determine if two images are the same scene. As a way of combining them, an option as the majority decision is available now, if two out of the three say it is a match, then it probably is actually a similar scene.



## 7. Conclusion

In conclusion, this study delved into the use and architectures of Siamese networks and SIFT in terms of temporal scene comparison and similarity detection, as a part of a bigger research into Convolutional Neural Networks. After a complex data structuring and preparation process, we were able to build it so that it meets the requirements for the Siamese Network and SIFT code, thus successfully running both of those algorithms. Our experiments showed that the Siamese Networks can learn similarity measures between pairs of photos effectively, much better than the traditional SIFT-based approach in terms of accuracy and robustness, however, we were able to use SIFT as a tool to identify and further analyse the Siamese Networks mistakes. Even though we were able to show the algorithms great promise in terms of scene comparison tasks, achieving our aims and objectives of deploying those models, especially when they dealt with a heterogeneous database, built out of many different datasets, there are still multiple challenges and issues to be further resolved and investigated.

Thus, having the opportunity to investigate those limitations and applications of both the Siamese Networks and the SIFT, we managed to identify many key areas for improvement, such as enhancing the scalability and efficiency of the training process, which comes hand in hand with optimising the models' performance. Furthermore, a conclusion was reached that on one hand the Siamese can be mistaken in some close scenarios in the form of False Positives and False Negatives, hinting at some deeper issues within their CNN structure. On the other hand, SIFT struggles with the storage of its unique points, as it identifies distinctive points in one image, which it looks for specifically in the other one, creating a massive matrix between those points, which in many scenarios is hard to store and process. Having a bad accuracy rate on top of that, it reveals to us that it is a demonstrable and fast approach but can run into significant difficulties. That is why we adopted the hybrid approach of combining the two methodologies so that they can eliminate their disadvantages and help each other build a more efficient method for classifying and comparing photos.

Further research into this area and topic can significantly lead to innovative advancements in computer vision and similarity detection spheres. Apart from tackling the limitations directly and looking to solve their issues, Class Activation Maps are an interesting approach that may be used as well. Even though it is generally not completely accepted as a way of achieving explainable AI, it might help with presenting the results in the form of a heat map. However, this heat map does not really show us the similarities, but rather the most visible and heavily tracked objects in both of the compared images, which in our case is not functional. What we conclude and gather from SIFT is stronger and better for similarity detection, demonstrating how to explain the comparison between photos. SIFT one-ups and builds on top of CAMs.

The conclusions we get from these experiments are largely applicable back to the real world and have the potential to make a positive change. Our research would be able to help with linking up different scenes together and draw conclusions from them, as to how similar they are and if they surely are the same scene. This case of effective identification and image comparison can be accepted in a courtroom, with the hybrid model of SIFT plus Siamese

Network providing accurate results and making them more demonstrable and explainable. In this way, there is going to be an additional layer of support for judges and juries to consider, in order to make their decision and successfully catch criminals, who have committed a crime and have been caught with evidence in the form of a photo from different angles, or generally any other case that may use scene detection.

## 7.1. Personal Reflection

Overall, this project has provided me with a greater and more in-depth understanding of Neural Networks and their challenges and applications. Prior to the project, my knowledge and skills were very limited to only what we've covered in Artificial Intelligence modules. I knew the basics of AI, but not how it evolves or how such algorithms are created. Furthermore, I knew that images could be compared for similarities, but not exactly how it happens or what exactly are we looking for. However, due to the work undertaken, I've got an immense amount of knowledge in these rather new and innovative technologies in the face of Convolutional Neural networks, Siamese Networks and Scales Invariant Features Transform. These skills are not limited to this project alone but can be transferred to the development of any Machine Learning algorithms in the sphere of Computer Vision.

The approach to the project work was largely successful, resulting in not only theoretical achievements but managing to fully build, run and visualise results from two different types of CNNs within a 20-week timeframe. Weekly goals were set and based on current progress throughout the research, architecture, development, and result discussion stages. This massively helped the project have steady and continuous progress. Nonetheless, there is definitely a place for improvement, as for instance the data preparation and structuring and then furthermore the development of the networks took longer than expected, causing further delays. If that could be improved, there would be more time to iron out any small complications that we had on the CNN algorithms, as well as develop them further to show us more and clearer results. This would also allow us to introduce thoroughly the concept of Class Activation Maps and visualise its heating-like maps, which make discussion way more demonstrable.

Coming in on this project with almost little to no knowledge or expectations of what's coming, I now feel way more confident in these areas and definitely look forward to applying the lessons taught to future projects. It presented an opportunity to exercise just the needed skills and concepts taught and solidify my understanding, opening the doors for me in the Computer Vision sphere. Data Science is certainly clearer for me, as well, so a future career in it is not out of question.

# References

- [1] Zhou, Zhi-Hua, 'Machine learning', Springer Nature, pp. 1-24, 2021
- [2] Stanford University, "Artificial Intelligence (Ai) Adoption Worldwide 2021, by Industry and Function." *Statista*, Statista Inc., March 15 2022, [Online]. Available: <https://www-statista-com.ezproxy.lancs.ac.uk/statistics/1112982/ai-adoption-worldwide-industry-function/>
- [3] Tversky, Amos., "Features of similarity.", *Psychological Review*, vol. 84, no. 4, 1977
- [4] W. Y. Ma and B. S. Manjunath, "Texture features and learning similarity," *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 425-430, 1996
- [5] Hemanth, D. Jude, and V. Vieira Estrela, eds. "Deep learning for image processing applications", Vol. 31. IOS Press, 2017.
- [6] K. H. Sun, H. Huh, B. A. Tama, S. Y. Lee, J. H. Jung and S. Lee, "Vision-Based Fault Diagnostics Using Explainable Deep Learning With Class Activation Maps," in *IEEE Access*, vol. 8, pp. 129169-129179, 2020
- [7] Cox, J. "How police secretly took over a global phone network for organized crime." *Motherboard Tech by VICE*, July 2 2020, [Online]. Available: <https://www.vice.com/en/article/3aza95/how-police-took-over-encrochat-hacked>
- [8] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR 2001, pp. I-I, 2001
- [9] Viola, P., Jones, M.J., "Robust Real-Time Face Detection", *International Journal of Computer Vision* 57, pp. 137–154, 2004
- [10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 05)*, vol. 1, pp. 886-893, 2005
- [11] R. Vyas, H. Rahmani, R. Boswell-Challand, P. Angelov, S. Black and B. M. Williams, "Robust End-to-End Hand Identification via Holistic Multi-Unit Knuckle Recognition," 2021 *IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1-8, 2021
- [12] R. Benson, 'To catch a paedophile, you only need to look at their hands', *WIRED UK*, 2021. [Online]. Available: <https://www.wired.co.uk/article/sue-black-forensics-hand-markings-paedophiles-rapists>
- [13] Baisa, Nathanael L., et al. "Multi-Branch with Attention Network for Hand-Based Person Recognition.", 2021
- [14] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., Zhu, J. "Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges." *Natural Language Processing and Chinese Computing (NLPCC 2019)*, vol 11839. Springer, Cham., 2019
- [15] O'Shea, Keiron, and Ryan Nash, "An introduction to convolutional neural networks.", 2015, *arXiv preprint arXiv:1511.08458*.
- [16] Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 8, 2021, 53. <https://doi.org/10.1186/s40537-021-00444-8>

- [17] Agarap, Abien Fred. "Deep learning using rectified linear units (relu)." *arXiv preprint*, 2018, *arXiv:1803.08375*
- [18] Alom, Md Zahangir, et al. "The history began from alexnet: A comprehensive survey on deep learning approaches." *arXiv preprint*, 2018, *arXiv:1803.01164*
- [19] Krizhevsky, A., Sutskever, I., and Hinton, G. E. "ImageNet classification with deep convolutional neural networks", In NIPS, 2012, pp. 1106–1114
- [20] Simonyan K, Zisserman A. "Very deep convolutional networks for large-scale image recognition"; 2014. *arXiv preprint arXiv:1409.1556*.
- [21] He K, Zhang X, Ren S, Sun J. "Deep residual learning for image recognition.", *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016
- [22] Huang, Gao, et al. "Densely connected convolutional networks." *arXiv preprint*, 2016, *arXiv:1608.06993*
- [23] F. Perazzi and J. Pont-Tuset and B. McWilliams and L. {Van Gool} and M. Gross and A. Sorkine-Hornung}, "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation", *Computer Vision and Pattern Recognition*, 2016, [Online]. Available: <https://davischallenge.org>
- [24] Björn Barz, Kai Schröter, Moritz Münch, Bin Yang, Andrea Unger, Doris Dransch, and Joachim Denzler., "Enhancing Flood Impact Analysis using Interactive Image Retrieval of Social Media Images." *Archives of Data Science*, Series A, 5.1, 2018, [Online]. Available: <https://github.com/cvjena/eu-flood-dataset>
- [25] Xiwu Zhang, Lei Wang, and Yan Su. "Visual Place Recognition: A Survey From Deep Learning Perspective", *Pattern Recognition*, November 2020, [Online]. Available: [https://github.com/ZhangXiwu/Awesome\\_visual\\_place\\_recognition\\_datasets](https://github.com/ZhangXiwu/Awesome_visual_place_recognition_datasets)
- [26] Antonio-Javier Gallego, Antonio Pertusa, and Pablo Gil, "Automatic Ship Classification from Optical Aerial Images with Convolutional Neural Networks", *Remote Sensing*, vol 10, no.4, 2018, [Online]. Available: <https://www.iuii.ua.es/datasets/masati/>
- [27] R. Sahdev and J. K. Tsotsos, "Indoor Place Recognition for Localization of Mobile Robots," *In 13th Conference on Computer and Robot Vision*, 2016, Victoria, BC, June 1-3, 2016, [Online]. Available: <https://www.raghavendersahdev.com/place-recognition.html>
- [28] Bittner, Dominik, Andrada, Maria Eduarda, Portugal, David, & Ferreira, João Filipe. "SEMFIRE forest dataset for semantic segmentation and data augmentation (1.0.0)" [Data set], Zenodo, 2021, [Online]. Available: <https://zenodo.org/record/5751906#.Y1p8xi8w2L1>
- [29] TrimBot2020 Dataset for Garden Navigation, [Online]. Available: <http://trimbot2020.webhosting.rug.nl/resources/public-datasets/>
- [30] Nathan Silberman, Derek Hoiem, Pushmeet Kohli and Rob Fergus, "Indoor Segmentation and Support Inference from RGBD Images", *ECCV*, 2012, [Online]. Available: [https://cs.nyu.edu/~silberman/datasets/nyu\\_depth\\_v2.html](https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html)
- [31] Tkacik G et al, "Natural images from the birthplace of the human eye", UPenn Natural Image Database, PLoS ONE 6: e20409, 2011, [Online]. Available: <https://web.sas.upenn.edu/upennidb/albums/>
- [32] Abdallah, Zahraa & Du, Lan & Webb, Geoffrey, "Data Preparation", *Encyclopedia of Machine Learning and Data Mining*, 2017, 10.1007/978-1-4899-7687-1\_62.
- [33] Jason Brownlee, "Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python", *Machine Learning Mastery*, 2020

- [34] Hastie, Tibshirani and Friedman, "The Elements of Statistical Learning (2nd edition)", Springer-Verlag, 2009, 241 - 250 pages (763).
- [35] Chicco, Davide, Siamese neural networks: an overview", *Artificial Neural Networks, Methods in Molecular Biology*, vol. 2190 (3rd ed.), New York City, New York, USA: Springer Protocols, Humana Press, pp. 73–94, 2020
- [36] Roy, Soumava Kumar, Mehrtash Harandi, Richard Nock, et al "Siamese networks: The tale of two manifolds." *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 2019
- [37] I. Melekhov, J. Kannala and E. Rahtu, "Siamese network features for image matching," *2016 23rd International Conference on Pattern Recognition (ICPR)*, Cancun, Mexico, 2016, pp. 378-383, doi: 10.1109/ICPR.2016.7899663.
- [38] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 6999-7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.
- [39] Jiangstein. "A Very Simple Siamese Network in Pytorch." *Kaggle*, Kaggle, 12 June 2019, <https://www.kaggle.com/code/jiangstein/a-very-simple-siamese-network-in-pytorch/notebook>
- [40] Lowe, D.G., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision* 60, pp. 91–110, 2004
- [41] X. Wangming, W. Jin, L. Xinhai, Z. Lei and S. Gang, "Application of Image SIFT Features to the Context of CBIR," *2008 International Conference on Computer Science and Software Engineering*, pp. 552-555, 2008
- [42] X. Pan and S. Lyu, "Detecting image region duplication using SIFT features," *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1706-1709, 2010
- [43] L. Yao, H. Feng, Y. Zhu, Z. Jiang, D. Zhao and W. Feng, "An architecture of optimised SIFT feature detection for an FPGA implementation of an image matcher," *2009 International Conference on Field-Programmable Technology*, Sydney, NSW, Australia, 2009, pp. 30-37, doi: 10.1109/FPT.2009.5377651
- [44] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva and Antonio Torralba, "Learning Deep Features for Discriminative Localization", *MIT*, 2015

# Appendix A: Proposal

## 1. Introduction

Machine learning is a form of Artificial Intelligence (AI) software that takes experience-based simulations and studies them in order to automate processes in a simpler way [1]. In the computers we use to train those algorithms, this experience that stands at the very basis of machine learning comes in the form of data and its aim is to create self-improving algorithms that construct models from the data. The models that result from those actions would be used for accurate predictions or assumptions on unknown observations, thus facilitating complex processes. Those industries have been booming in recent years, as they are now integrated into almost every segment of our day-to-day lives and adopted within all big companies we interact with [2]. Comparing scenes and images and detecting similarities between them is no exception to that and has been deeply embedded within Machine Learning as a way of creating models that solve real-life issues. Generally, similarity would mean just comparing two objects, or in our case images and scenes, and analysing their touching points seeing the analogy between the two [3], however, the more technical and scientific approach which is going to be used in this paper is to split the images into texture patterns and features [4]. This way a neural network can be established which would allow for a scene's features to be examined and the similarity of images to be determined in a greater depth and in a larger scale. We would be able to determine what are 'good' and 'bad' features. There have been studies like that with various approaches, but the remaining issue, which will be discussed by this paper, is that it is not sure how efficient they are, as they are not very generalisable and not demonstrable.

There have been many attempts to achieve scene comparison and get an understanding of how it works and how it can be put into use. Many of those successes have been put into use in various spheres ultimately helping society solve a lot of issues and find an answer to complex challenges. Of course, those processes have progressed from the beginning and the very early tries in image comparison, as now we have new approaches such as SIFT, Class Activation Maps (CAM) and Siamese Networks. Each of those shows a different perspective in scene resemblance and would be limited on its own, thus in this paper, we're going to look at and analyse all of them.

Talking about limitations in scene comparison and similarity detection, several key things inflict stagnation and restrict processes from having positive results. In some approaches, the features that are used in order to create a model are chosen manually by an expert with specific knowledge in a certain field, which however is not as automatic as an algorithm choosing it, which may lead to some features dominating the performance of the classifier, thus creating a bias in a way of analysing the scene later with those features [5]. In a different manner, image comparison wasn't able to help in the case of Encrochat, when authorities had at their disposal thousands of images of crime scenes and illegal resources, which could be used to catch the perpetrators [6]. If the authorities had a natural general scene comparison model, which was able to analyse similar images and compare them with their features and

similarities, extract key points and present them in a way understandable for the police, the prosecuted criminals would be way more. Just as the case is in medical images, most diagnoses are in single shots, while with a model like the one mentioned above, we would be able to build a temporal model of a diagnosis that changes over time and shows doctors invaluable information about how things are accelerating. This is the aim of the project - to create something indispensable, a model which can be used in many industries with multiple benefits and advantages.

This project proposal will be split into the following segments; motivation and background, aims and objectives, methodology and challenges, the data; the project timeline; and references. The motivation and background will include a description of similar projects and previous attempts at scene comparison through Siamese networks and Class activation maps, while also indulging in my desire to present the matter through explainable AI. The aims and objectives will contain the main focus and key points for the project, and the methodology and challenges section will go through the chosen approaches for proving my thesis and showing how the data has been retrieved and evaluated through multiple spectrums. The data section will detail the range of images and resources that are going to be analysed throughout the project. The project timeline would outline the time and order of every process that has been taken within the project. At the same time, the references will encompass all references to any sources and papers that were used in the proposal.

## **2. Motivation and Background**

The earliest machine-learning projects in the sphere of image and scene comparison were directly related to computer vision. Those researches mostly drew upon an enhanced cascade of simple features [7],[8], and had a feature descriptor, which was used to decipher and extract the features in a way called Histograms of Oriented Gradients (HOG) [9]. Those approaches proved to be more efficient than anything similar to them and were used as groundbreaking research at the time.

However, this approach was pretty basic and with time passing by, the processes that needed to be serviced became more complex, so the machine learning algorithms had to adjust to that and progress further, becoming more complicated themselves. In that manner, there has been extensive research into automatically identifying individuals via holistic multi-unit knuckle recognition, which is based on a deep neural network and has proven to be successful in solving cases and catching criminals by only having their hands in the image [10]. It is only one of the multiple such resources that have shown how image comparison and similarity detection can be useful in real-life situations, and used for good to help people overcome various challenges. The research itself stems from a more general hand-recognition research, which was started by Sue Black, in order to catch paedophiles who have shown only their hands in the photo imagery they've taken [11][12].

All of this is done in a way for Explainable AI to be created and integrated successfully in every way that's needed. Explainable AI can be split into two main categories, one of them being transparency behind the AI's decisions, understanding of its model structure and its



components and understanding of the algorithm that trained it, while the second one is more of a post-hoc explanation into how and why the specific processes have occurred in order for the AI to reach its conclusion, giving analysis and visualisations to help prove its point [13]. This way it becomes a more reliable and acceptable method for more important professions, such as medical doctors, to base their decisions on Explainable AI. This is what we're striving for in this project as well, we want demonstrable and explainable results.

### 3. Aims and Objectives

The overall aim of this project is to investigate models for image and scene comparison and their potential application in medical diagnostics. In order for this to be achievable, we would need to curate data sets and carry out a literature review, while also examining baseline approaches for scene comparison. As challenging as the project is, it will be completed with the following objectives:

- Outlining of comparable features and common models of comparison
- Investigation of the object classification approach. This will involve Siamese networks and scene comparison between 2 photographs from the data sets concluding if the scenes are the same or not.
- Investigation of the use of Class activation maps, where if there's a similarity in scenes, it gives a heat map to indicate which bits of the image are similar and analyse correlations.
- Investigation of the influence of certain common features in scenes with large similarities.
- Investigation into how my models can be applicable and helpful in medical diagnostics.
- Preparation of a demonstration model

### 4. Methodology and Challenges

Before starting with the approaches and the right methodology, we will need to sort out the datasets and the imagery all of the algorithms are going to work on. For this purpose, an artificial dataset would need to be created, composed of all of the databases mentioned in the Data section below this one. This way a controlled environment is created, with an amount of overlap, which is adjustable, where less overlap means less accuracy. The experiment is going to be controlled in that sense as well since the dataset we create and use can be of whatever size we want. Various baseline approaches for scene comparison exist, however, the ones that we're going to use are the SIFT approach, the Object Classification approach which would be seen from the sphere of Siamese Networks and Class Activation Maps (CAM).

Scale Invariant Features Transform (SIFT) is a rather new approach created by David G. Lowe in 2004 [14] that has gained a lot of traction lately as a very robust and distinctive comparative mechanism able to find similarities and collations, with feature extracting and matching while being invariant to scale and rotation [15][16]. The method can detect more general duplications between scenes based on the image SIFT features. This is done in a few



steps, starting with identifying and detecting keypoints in an image and computing the local maximum and minimum points invariant to scale and rotation, which is also called SIFT features. The next step is to localise all feature points in the image, create a detailed model, get rid of all outliers and not stable points and only after that match all keypoints with their closest correspondence in the image. Afterwards, a feature descriptor is generated by sampling image gradient magnitudes and orientations across all of the detected keypoints and placing them inside an array of orientation histograms in a piece around the designated points. Gradients are measured with scale invariance and all entries from all histograms are put in a 128-dimensional vector to create the feature descriptor, with which later on we process similarity between images. This is a more traditional approach, which is very demonstrable, but with rather worse results compared to the other ones we're going to use, with one of them being Siamese Networks in an Object Classification manner.

Siamese neural networks essentially upgrade on top of the Sift Approach, as it operates in pairs together on two different input vectors to compute comparable output vectors, easily being able to make computations between two images and give a result if they are similar or not. [17][18]

In a similar way, CAM can deduct for you if two images are similar or not, however, it doesn't stop there and goes deeper into its analysis of the image, highlighting with thermal imagery into the regions and parts that have the biggest similarities. This significantly helps with learnings of the networks and eases up debugging, as it also results in an object localization without manually labelling the box bordering the object. [19][20]

One method by itself is not enough, for instance, CAM tells you about the influence, but doesn't look into the corresponding regions, while SIFT investigates correspondence between scenes in a better way. That's why we want to use all 3 approaches and prove demonstrability and explainability in our results.

## 5. Data

This project focuses on scene comparison and similarity detection, so the data used would need to include: i) two types of data sets - one would be the natural general scenes dataset and the other one would feature a more controlled set - artificially generated matching scenes, ii) many photos taken of different angles but of the same scenes, iii) corresponding segmentations - to be used for distinguishing between elements and scenes. The databases that are going to be taken into consideration include DAVIS [21], EU Flood Dataset [22], Indoor Places [23], MASATI [24], Raghavender Sahdev Places [25], SEMFIRE [26] and Trimbot [27].

## 6. Project Timeline

In order to present the project's timeline in an easy-to-understand manner, the following Gantt chart was constructed and shown below. Gantt charts are perfect for summarising the overall perspective of the project, efficiently cutting the processes and actions into manageable portions and creating a productive schedule [28].

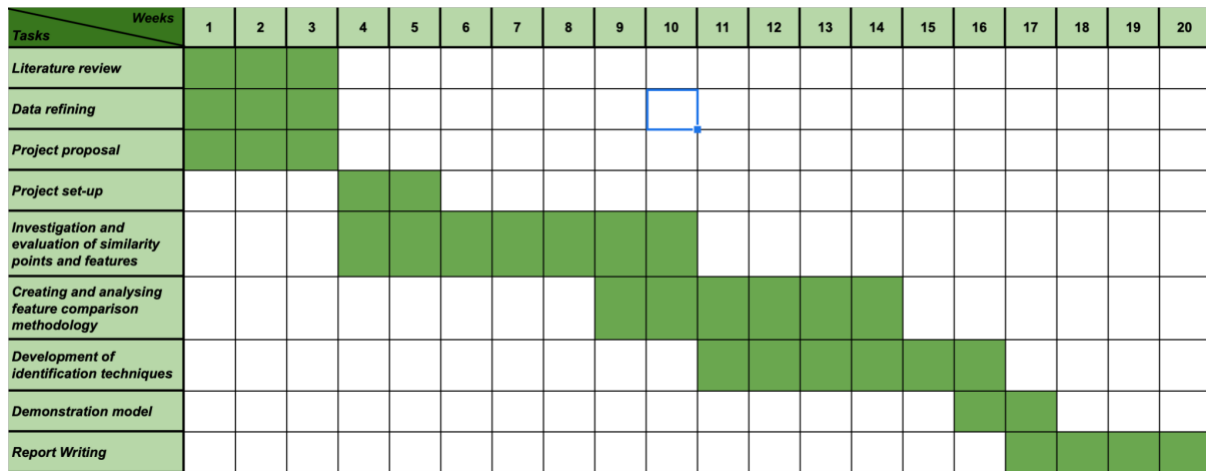


Figure 1: Gantt Chart

## 7. References

- [1] Zhou, Zhi-Hua, 'Machine learning', Springer Nature, pp. 1-24, 2021
- [2] Stanford University, "Artificial Intelligence (Ai) Adoption Worldwide 2021, by Industry and Function." Statista, Statista Inc., March 15 2022, [Online]. Available: <https://www-statista-com.ezproxy.lancs.ac.uk/statistics/1112982/ai-adoption-worldwide-industry-function/>
- [3] Tversky, Amos., "Features of similarity.", *Psychological review*, vol. 84, no. 4, 1977
- [4] W. Y. Ma and B. S. Manjunath, "Texture features and learning similarity," *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 425-430, 1996
- [5] K. H. Sun, H. Huh, B. A. Tama, S. Y. Lee, J. H. Jung and S. Lee, "Vision-Based Fault Diagnostics Using Explainable Deep Learning With Class Activation Maps," in *IEEE Access*, vol. 8, pp. 129169-129179, 2020
- [6] Cox, J. "How police secretly took over a global phone network for organized crime." *Motherboard Tech by VICE*, July 2 2020, [Online]. Available: <https://www.vice.com/en/article/3aza95/how-police-took-over-encrochat-hacked>
- [7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, pp. I-I, 2001
- [8] Viola, P., Jones, M.J., "Robust Real-Time Face Detection", *International Journal of Computer Vision* 57, pp. 137-154, 2004
- [9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886-893, 2005
- [10] R. Vyas, H. Rahmani, R. Boswell-Challand, P. Angelov, S. Black and B. M. Williams, "Robust End-to-End Hand Identification via Holistic Multi-Unit Knuckle Recognition," 2021 *IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1-8, 2021
- [11] R. Benson, 'To catch a paedophile, you only need to look at their hands', *WIRED UK*, 2021. [Online]. Available: <https://www.wired.co.uk/article/sue-black-forensics-hand-markings-paedophiles-rapists>

- [12] Baisa, Nathanael L., et al. "Multi-Branch with Attention Network for Hand-Based Person Recognition.", 2021
- [13] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., Zhu, J. "Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges." *Natural Language Processing and Chinese Computing* (NLPCC 2019), vol 11839. Springer, Cham., 2019
- [14] Lowe, D.G. , "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision* 60, pp. 91–110, 2004
- [15] X. Wangming, W. Jin, L. Xinhai, Z. Lei and S. Gang, "Application of Image SIFT Features to the Context of CBIR," 2008 *International Conference on Computer Science and Software Engineering*, pp. 552-555, 2008
- [16] X. Pan and S. Lyu, "Detecting image region duplication using SIFT features," 2010 *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1706-1709, 2010
- [17] Chicco, Davide, "Siamese neural networks: an overview", *Artificial Neural Networks, Methods in Molecular Biology*, vol. 2190 (3rd ed.), New York City, New York, USA: Springer Protocols, Humana Press, pp. 73–94, 2020
- [18]
- [19] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva and Antonio Torralba, "Learning Deep Features for Discriminative Localization", *MIT*, 2015
- [20]
- [21] F. Perazzi and J. Pont-Tuset and B. McWilliams and L. {Van Gool} and M. Gross and A. Sorkine-Hornung}, "A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation", *Computer Vision and Pattern Recognition*, 2016, [Online]. Available: <https://davischallenge.org>
- [22] Björn Barz, Kai Schröter, Moritz Münch, Bin Yang, Andrea Unger, Doris Dransch, and Joachim Denzler., "Enhancing Flood Impact Analysis using Interactive Image Retrieval of Social Media Images." *Archives of Data Science*, Series A, 5.1, 2018, [Online]. Available: <https://github.com/cvjena/eu-flood-dataset>
- [23] Xiwu Zhang, Lei Wang, and Yan Su. "Visual Place Recognition: A Survey From Deep Learning Perspective", *Pattern Recognition*, November 2020, [Online]. Available: [https://github.com/ZhangXiwuu/Awesome\\_visual\\_place\\_recognition\\_datasets](https://github.com/ZhangXiwuu/Awesome_visual_place_recognition_datasets)
- [24] Antonio-Javier Gallego, Antonio Pertusa, and Pablo Gil, "Automatic Ship Classification from Optical Aerial Images with Convolutional Neural Networks", *Remote Sensing*, vol 10, no.4, 2018, [Online]. Available: <https://www.iuii.ua.es/datasets/masati/>
- [25] R. Sahdev and J. K. Tsotsos, "Indoor Place Recognition for Localization of Mobile Robots," *In 13th Conference on Computer and Robot Vision*, 2016, Victoria, BC, June 1-3, 2016, [Online]. Available: <https://www.raghavendersahdev.com/place-recognition.html>
- [26] Bittner, Dominik, Andrada, Maria Eduarda, Portugal, David, & Ferreira, João Filipe. "SEMFIRE forest dataset for semantic segmentation and data augmentation (1.0.0)" [Data set], Zenodo, 2021, [Online]. Available: <https://zenodo.org/record/5751906#.Y1p8xi8w2L1>
- [27] [Online]. Available: <http://trimbot2020.webhosting.rug.nl/resources/public-datasets/>
- [28] James M. Wilson, 'Gantt charts: A centenary appreciation', *European Journal of Operational Research*, vol 149, issue 2, 2003, pp. 430-437, ISSN 0377-2217

Artificial Neural Networks, Methods in Molecular Biology, vol. 2190 (3rd ed.), New York City, New York, USA: Springer Protocols, Humana Press, pp. 73–94, 2020[Artificial Neural Networks, Methods in Molecular Biology, vol. 2190 (3rd ed.), New York City, New York, USA: Springer Protocols, Humana Press, pp. 73–94, 2020[Artificial Neural Networks, Methods in Molecular Biology, vol. 2190 (3rd ed.), New York City, New York, USA: Springer Protocols, Humana Press, pp. 73–94, 2020[Artificial Neural Networks, Methods in Molecular Biology, vol. 2190 (3rd ed.), New York City, New York, USA: Springer Protocols, Humana Press, pp. 73–94, 2020[Artificial Neural Networks, Methods in Molecular Biology, vol. 2190 (3rd ed.), New York City, New York, USA: Springer Protocols, Humana Press, pp. 73–94, 2020