

Interactively classifying visual art



Ruben Ahrens

Layout: typeset by the author using L^AT_EX.

Cover illustration: By Leonardo da Vinci - Cropped and releveled from File: Mona Lisa, by Leonardo da Vinci, from C2RMF.jpg. Originally C2RMF: Galerie de tableaux en très haute définition: image page, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=15442524>

Interactively classifying visual art

Ruben Ahrens
12201049

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*



University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor

Dr. N.J.E. van Noord

Institute for Logic, Language and Computation
Faculty of Science
University of Amsterdam
Science Park 907
1098 XG Amsterdam

9 July, 2021

Abstract

Data annotation for machine learning can be a time-consuming and inefficient process. Interactive machine learning aims to make this process more efficient. This study aims to determine in what way interactive machine learning can be used to classify visual art pieces into discrete classes. In this context, interactive machine learning is defined as a machine learning algorithm that incorporates user feedback into the training process. Furthermore, this study compares the performance of interactive machine learning to conventional computer vision methods. Interactive machine learning is expected to outperform conventional machine learning algorithms in terms of data efficiency.

To evaluate if interactive machine learning provides a viable computer vision approach, this method was applied to a subset of the open-source WikiArt dataset. Approximately two thousand images were annotated by a human expert to identify approximately 500 portrait and 500 landscape paintings. These images were subsequently transformed with a vision transformer that extracted the appropriate features of the images. The features extracted from these images were used with the interactive machine learning algorithm that employed a support vector machine for classification. Three variations of interactive machine learning were compared to find the best approach for deciding which images to present to the user. The results indicate that uncertainty sampling allows for the lowest number of images needing to be annotated with interactive machine learning.

The present findings suggest that interactive machine learning is a viable alternative to other machine learning methods, especially when data is not readily available and needs to be annotated by a human.

Contents

1	Introduction	2
2	Background	5
2.1	Artificial neural network	5
2.1.1	Feedforward	6
2.1.2	Backpropagation	7
2.2	Convolutional neural network	7
2.3	Vision transformer	10
2.4	Support vector machine	12
2.5	Interactive machine learning	14
3	Related Works	16
4	Methods	19
4.1	Data	20
4.2	Data preparation	20
4.3	Getting image transformations with CLIP	24
4.4	Classifying images with CLIP	25
4.5	Support vector machine classification	25
4.6	Interactive learning machine learning	26
4.6.1	Uncertainty sampling	27
4.6.2	Query by committee	27
4.6.3	Random sampling	27
5	Results	28
6	Conclusion & Discussion	32
6.1	Conclusion	32
6.2	Discussion	33
A	Setup	38

Chapter 1

Introduction

Annotating art has long been a time-consuming and laborious task for art historians. Annotation is the act of providing extra information about a document. In this study, this term is defined by labelling visual works of art. Not only in the art history field but also in the entire computer vision domain, a lot of time is being spent manually annotating images and video, Joshi et al. (2009). In short, this process is cumbersome and should be optimized to increase efficiency. For some time, computer vision has been reliant on big data algorithms such as convolutional neural networks. Using algorithms of this nature may put a strain on the annotation process. Therefore, it is beneficial to find a method to reduce the need for data. Compared to artificial intelligence, humans tend to require fewer data to learn a task or classification. A technique called interactive machine learning (IML) takes advantage of this phenomenon: it incorporates humans in the training cycle of a machine learning algorithm. A human agent is asked to classify data like images, video or audio in the case of low certainty with interactive learning algorithms. Most classification algorithms classify data based on the probability that it belongs to a certain class. When the probabilities of classes are highly similar, the algorithm has a low certainty of its classification. By making the interacting agent (human being or simulated human being) classify this uncertain data, it will both have a correct classification and it will be more represented in the training data, making the training data more balanced between edge cases and regular cases. Incorporating humans into the training loop could reduce the need for annotated data and thus time annotating this data.

In particular, art historians will benefit from this development, since a lot of their time is spent annotating works of art, Carneiro (2011). By improving the efficiency of the annotation process, they can focus on more challenging research. Moreover, art research itself will benefit from a more efficient classification method. Ultimately, more data will be classified faster since the annotation step is a bot-

tleneck in the computer vision training process. This will allow art researchers to use more data in their research, improving the validity of their results.

To determine if interactive machine learning decreases the need for art annotation, it has been applied to a dataset of visual art. A first step in developing the needed interactive workflow is using the interactively trained algorithm to distinguish a portrait from a landscape painting. In this study, a portrait painting is defined as a painting where a human being has been portrayed recognizably. Both facial expressions and posture are important to recognizing a portrait. When using this definition, images of ideas like the Virgin Mary or Jesus Christ are portrayals of ideas, not actual people. Though visually, these paintings may seem very similar to portraits. However, to make the data useful for further work, religious images as aforementioned are not annotated as a portrait. A landscape is simply defined as a stretch of land that stands on its own. Meaning cityscapes or seascapes will not be classified as landscapes.

The interactive machine learning algorithm will be applied to images of paintings belonging to a wide variety of styles, eras, and artists. The features of the images will be extracted through a neural network. Interactive learning differs from other machine learning methods by incorporating user feedback into the training cycle. With non-interactive supervised learning, the annotation process for the training cycle will be less efficient. With interactive learning, there is some assurance that the annotation an expert is making will improve the performance of the algorithm. In essence, it helps to reduce the need to annotate images that the algorithm already performs well on and will thus allow experts to focus on the edge cases. When interactively classifying visual art, an image is presented to the user for expert validation. In the user interface, the user classifies the image and sends this data back to the algorithm, which will be trained differently based on the user-generated feedback.

Interactive machine learning has a history of being used for a wide array of machine learning tasks, including computer vision Rui et al. (1998). Its main purpose is to reduce training set size and overfitting. If abstract images like paintings could be classified using interactive machine learning, there will be a decrease in the workload for annotating data and a reduction in necessary computing power.

Based on these opportunities and challenges, several questions arise. The main question is: How can an interactive classifier be trained to classify visual art? To answer this question, the following sub-questions need to be answered:

- How can user feedback be incorporated into the training?
- Which images should be presented to the user and how?
- How does an interactive classifier compare to its supervised learning counterpart in terms of accuracy when being trained on the same number of

samples?

It is hypothesised that it is possible to develop an interactive machine learning classifier able to classify visual art that can compete with deep learning techniques. User feedback will be incorporated into training by fitting the model on the image data and the labels the user has provided for said images. The images that will increase the model accuracy the most when the model is fit to those images should be presented to the user. An interactive classifier will be more accurate when trained on a low amount of data compared to its supervised learning counterpart. This thesis will aim to answer these research questions as accurately as possible. To provide the reader with sufficient context, this thesis is structured as follows: Firstly, all background knowledge like neural networks and interactive learning will be explained in chapter 2. Secondly, in chapter 4 the data will be discussed, the used algorithms will be illustrated, the interactive and supervised learning solutions will be described, and, finally, the accuracy of the interactive solution will be compared to its supervised counterpart in section 5.

Chapter 2

Background

In this section, the background of the study will be explained to provide the appropriate context to enable an understanding of the material discussed in the latter part of the thesis. First, the artificial neural network will be explained as this is used in CLIP (Contrastive Language-Image Pre-Training). Secondly, the convolutional neural network will be explained. Convolution is a popular image preprocessing technique and was used in CLIP before the application of the vision transformer. The vision transformer will also be explained, as this is not only used with the classification with CLIP but the transformed images will also be used with the other algorithms. The other algorithms are supervised support vector machine (SVM) and interactive machine learning.

2.1 Artificial neural network

An artificial neural network (ANN) Hecht-Nielsen (1992) is a simulated network similar to a human brain. An ANN consists of multiple layers: an input layer, (a) hidden layer(s), and an output layer. Each of these layers consists of a certain number of neurons. Every individual neuron holds data: an activation, weights for the next layer and a bias. With a classification problem, the number of neurons corresponds to the number of classes. Every neuron represents a class, with its activation being the probability of the original input belonging to said class. Input data has to be represented in one dimension. Through forward pass, the input data is manipulated with the weights of the neurons in every layer, typically using a dot product. Through backpropagation, weights are adjusted to get to the desired output. Feedforward and backpropagation will be further explained later. A simplistic neural network is visualized in figure 2.1.

The circles represent neurons, each holding an activation value, a bias and weights for the neurons in the next layer. The first layer of a neural network is

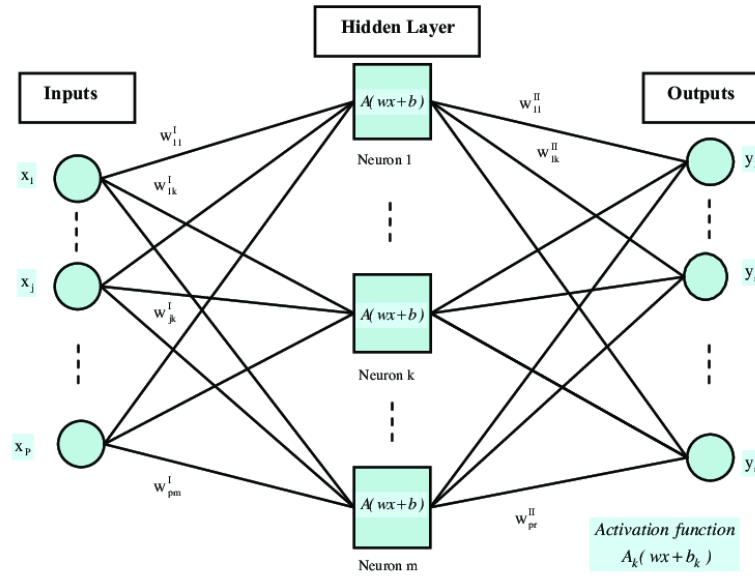


Figure 2.1: Fig. 1. The simple ANN structure with one hidden layer. x being the inputs, w being the weights, b being the biases, y being the output activations. Nonlinear time series forecasting with Bayesian neural networks - Scientific Figure on ResearchGate.

the input layer; every neuron in this layer is assigned a single value from the input data. For example, every neuron is assigned one pixel-value from a grayscale image, with 0 being black and 1 being white with a gradient in between. This data in the neuron is called “the activation”, comparable to a voltage value in a biological human neuron. The layers in between the first and last layers are called the hidden layers. The activation is computed with an activation function. The input of the activation function is calculated by feed forwarding the previous layer’s activations and weights and adding the bias. Weight is a value with which a neural activation will be multiplied to obtain the activation of a neuron in the following layer. The weights and biases can be optimized through backpropagation training of the network. Making the neural network able to learn a specific task.

2.1.1 Feedforward

One neuron has a weight for every neuron in the next layer, typically this is represented in a vector. These vectors may be represented in a matrix with every column having a neuron’s weight vector. $[w_{0,0}, w_{1,0}, \dots, w_{k,0}]$ is the weight vector for a single neuron in the first layer. The weights are the values that will be multiplied by the activation of the neuron $a_0^{(0)}$. To get the activation for $a_0^{(1)}$, $w_{0,0}$

will be used in the calculation, to get the activation for $a_k^{(1)}$, $w_{k,0}$ will be used in the calculation. With n being the number of neurons in the first layer and k being the number of layers in the second layer. These calculations can all be simplified to $a^{(1)} = \sigma(Wa^{(0)} + b)$ with $a^{(1)}$ being all node activations of the second layer, $a^{(0)}$ being all node activations of the first layer. b being all biases for every node in the second layer represented in a vector. σ being the sigmoid function. And W being the weight matrix. A neural network using this process to feed forward the neuron's activations is called a feedforward neural network. Nowadays sigmoid is not widely used anymore. CLIP also works with a ReLU activation function instead of a sigmoid function. An activation function is essentially a mapping to the activation of a neuron to a number that is between 0 and 1. ReLU is a much simpler function improving efficiency in the training process, Agarap (2018).

2.1.2 Backpropagation

When a network is learning, it uses backpropagation to find the right weights and biases for every neuron. A cost function is introduced to give a low cost for a correct prediction and a high cost for an incorrect prediction. The average cost of the network will be minimized through gradient descent, Ruder (2016). Minimizing this function is approaching the best weights and biases. Because computing these gradients takes a long time with a lot of data, stochastic gradient descent is used. With stochastic gradient descent, the training set is added in batches to faster minimize the error function.

2.2 Convolutional neural network

A convolutional neural network (CNN) Albawi et al. (2017) is a widely used image classification technique. Before implementing a vision transformer Dosovitskiy et al. (2020), the CLIP model used convolution to preprocess the images it fed to the neural network. CNN can still be used with CLIP, however, this study employed the Vision Transformer (ViT) Dosovitskiy et al. (2020) trained model. In this part, the preprocessing stage of a CNN will be explained. Firstly, the convolution itself will be explained. Secondly, max-pooling will be explained. And finally, the bigger picture will be portrayed.

In principle, a convolutional neural network consists of two parts: image pre-processing and the artificial neural network. Convolution is used in the image preprocessing of a CNN. The neural network can differ for every classification task, but in essence, it is the same as the ANN previously explained. Convolution is a technique used to improve the data fed to the neural network. The data is preprocessed to extract features useful for classification. This improves an ANN's

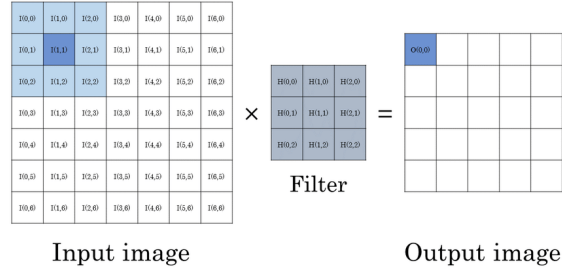


Figure 2.2: Image convolution with an input image of size 7×7 and a filter kernel of size 3×3 . Streaming Architecture for Large-Scale Quantized Neural Networks on an FPGA-Based Dataflow Platform - Scientific Figure on ResearchGate.

Graphic representation of the first step of convolution. The filter being the kernel in this case. The first step and first pixel of the output image is calculated as follows: $O_{(0,0)} = \sum(I_{(0,0)} * H(0,0) + I_{(1,0)} * H(1,0) + I_{(2,0)} * H(2,0) + I_{(0,1)} * H(0,1) + I_{(1,1)} * H(1,1) + I_{(2,1)} * H(2,1) + I_{(0,2)} * H(0,2) + I_{(1,2)} * H(1,2) + I_{(2,2)} * H(2,2))$

performance on this data. This data preprocessing is performed using multiple techniques, convolution, max-pooling, splitting image features such as colour to make a 3d image with different RGB features in every layer.

Convolution requires the implementation of a convolutional kernel. A convolutional kernel can extract features like edges from an image using different kernels. The main function of convolution is transforming the representation of an image into a representation that better suits a neural network input. For example, isolating the edges of an object reduces a lot of background noise from the image. This noise can cause overfitting of the neural network on training data.

A convolutional kernel works as follows: the kernel is a matrix, typically 3 by 3 dimensions. Firstly, the middle of the kernel will be “laid over” every pixel of the image inside a border large enough to accommodate the edges of the kernel. Padding can be used to add an extra border to retain the resolution of the image and to prevent loss of data after convolution. Without padding, potentially essential information of the image around the border can be lost. The size of the padding border is dependent on the size of the kernel. For a 3 by 3 kernel 1 layer is padded to the image. Secondly, when convolving, the output image’s pixel will be at the location the middle of the kernel is at. The value of this pixel in the output image will be the sum of all kernel values multiplied by the pixel-value it is laid over. This is portrayed in figure 2.2. After this calculation, the kernel will be moved a step to the right depending on stride. The size of the output is affected by the stride of the convolution. By default, convolution is done with a stride of 1. But with stride 2 every second node is skipped and thus the output dimensions will be smaller.










Operation	Kernel ω	Image result $g(x,y)$
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3×3 (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian blur 5×5 (approximation)	$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
Unsharp masking 5×5 Based on Gaussian blur with amount as 1 and threshold as 0 (with no image mask)	$-\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	

Figure 2.3: Effect of different convolutional kernels on an image as shown by Albawi et al. (2017). The general form of convolution can be seen in equation 2.1

Convolving can have a variety of functions. The content of the kernel decides its function. A kernel that can detect an object's edges can be seen compared to other convolutional kernels in figure 2.3. For object recognition, the most useful kernels are the edge detection kernels. These kernels work by taking local gradients. Equivalent to taking a gradient of a 2-dimensional function, the gradient is largest where the difference between data of two compared points is the largest. The functionality of these kernels is similar to taking the derivative of the image.

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} * \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{(m-i)(n-j)} y_{(1+i)(1+j)} \quad (2.1)$$

Equation 2.1: General form of convolution. Where x is the convolution kernel and y is an image.

The pooling layer takes on the task of reducing the dimensionality of the image. As explained before, a neural network takes inputs in one-dimensional vectors. Pooling aids to convert a 2-dimensional image into a one-dimensional one. Pooling takes on fragments of an image and reduces every fragment to a single pixel. The pixel value is determined by the values in the fragment and the pooling technique. Two techniques are used in pooling: max-pooling and average pooling. Max-pooling outputs the pixel with the highest value in the pool. Average-pooling computes the average of the pool and outputs this value to the resulting image.

In conclusion, a convolutional neural network is a deep learning neural network that reduces the dimensionality of an image, while reducing input noise to improve computer vision results with artificial neural networks. This is done by using convolution, pooling and an activation function. In an ANN the input values must be a neuron's activation. Therefore, input values must lie between 0 and 1. For this purpose, an activation function is used. Typically a ReLu function but a softmax or sigmoid is also possible, however in contrast to ReLu the latter is less time or computing power efficient.

2.3 Vision transformer

Vision Transformer (ViT), Dosovitskiy et al. (2020) is a state of the art image processing technique developed by Dosovitskiy et al. CLIP Radford et al. (2021) is a vision transformer neural network trained on millions of images and can classify any image and text pairs. CLIP can even describe images in natural language by predicting the content of an image and forming a sentence.

When pre-trained with sufficient images, a vision transformer can approach CNN results while requiring less time and computing power. The vision transformer model proposed by Dosovitskiy et al. involves the following: As seen in figure 2.4, a 2-dimensional image is transformed into a 1-dimensional representation of itself. The resolution is normalized, so all images have an equal amount of dimensions. Positions are saved by the position embedding and the transformed image is fed to the transformer encoder. The transformer encoder consists of two

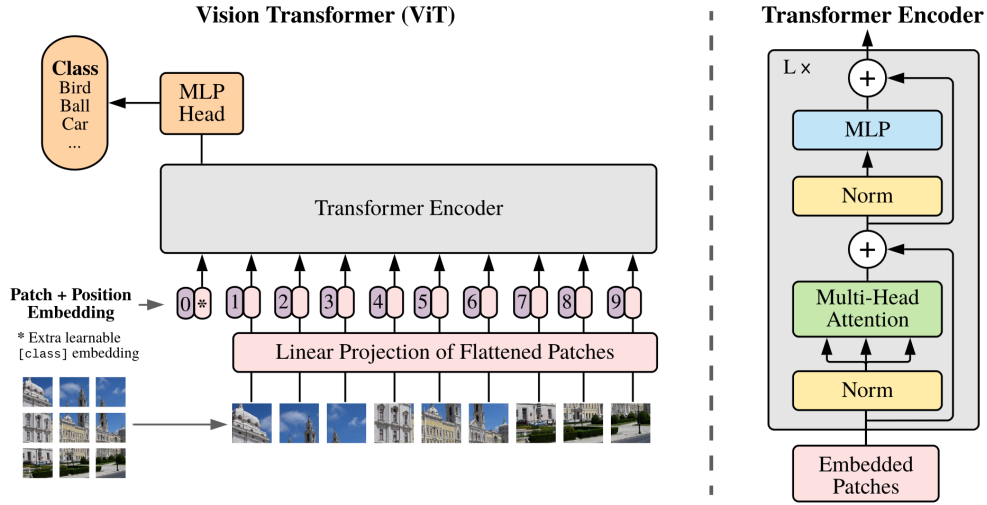


Figure 2.4: Illustration of ViT model, Dosovitskiy et al. (2020)

algorithms: a multi-headed self-attention (MSA) Vaswani et al. (2017) algorithm and an MLP. An MSA is based on a CNN but made to be more efficient and have an output that filters out more noise than a CNN. A self-attention layer in an MSA algorithm has a lower complexity compared to a convolutional layer, as described by Vaswani et al. (2017). The resulting upgrade makes the output better suited as an input for an ANN. It puts out an attention image of an input image. Attention is directed because the model puts more weight on the distinctive parts of the image. In the case of a portrait, for instance, probabilities of importance in the area of a face will be weighted heavier than areas in the rest of the painting with lower probabilities. The final class output is handled by a multi-layer perceptron (MLP) with one hidden layer during training. A multi-layer perceptron is a feedforward neural network with at least three layers using backpropagation for learning.

There are three variants of the ViT model: ViT-Base, ViT-Large and ViT-Huge. The notation is done as such: ViT-B/32 means the “Base” variant with 32×32 input size. ViT-B/32 is the ViT model available in the CLIP python model. Other available models are ResNet models, He et al. (2016)

The ViT of Dosovitskiy et al. (2020) was used in the CLIP algorithm. The ViT has a less image-specific inductive bias, which means that a ViT model is better applicable to zero-shot data. This is useful for the CLIP algorithm as performing well on zero-shot data is one of the purposes of CLIP and specialities in the state of the art. Radford et al. (2021) found the ViT model to have better performance compared to the ResNet.

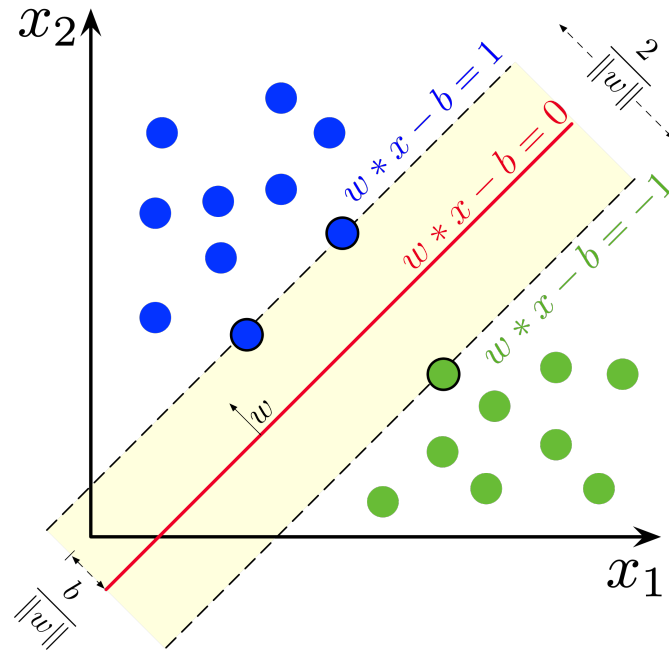


Figure 2.5: Support vector machine hyperplane, data points and margin. By Larhmam - Own work, CC BY-SA 4.0

2.4 Support vector machine

A support vector machine is an artificial intelligence technique used in classification and regression, Noble (2006). Regression differs from classification in having a continuous output variable instead of a discrete output value as a mapping of a function on inputs. A continuous output means predicting a number, for example, a house price. Classification will consist of a probability of data belonging to a certain class. In its simplest form, a support vector machine is a linear, binary classifier. However, linear, radial basis or polynomial classification kernel functions are also available. A radial basis function kernel (RBF) is a discriminatory function, based on a circular decision boundary. This stands in contrast with a linear kernel, which has a linear decision boundary, and the polynomial kernel, which has a polynomial decision boundary. Depending on the problem and the data, an RBF can be a better discriminator than its linear counterpart. With SVM, a hyperplane distinguishes data points into two classes. The dimensions are manipulated to find the dimensions in which the data points can be divided by the hyperplane. The different discriminatory functions are visualized in figure 2.6.

RBF (radial basis function) is similar to the k-nearest neighbours algorithm. RBF finds hyperplanes in infinite dimensions by calculating the relationship between vectors in infinite dimensions. When viewing a simple case in figure 2.6, a

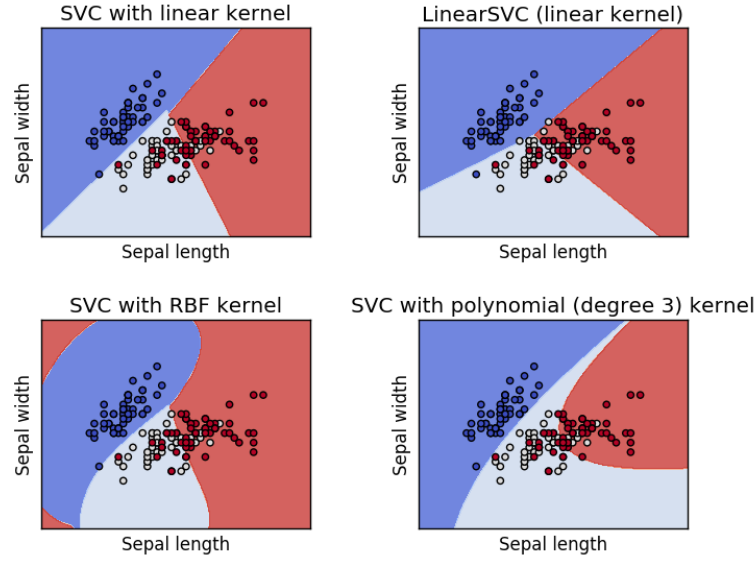


Figure 2.6: Effects of using different kernels with a support vector machine.

linear hyperplane has the advantage of preventing overfitting by design but has a disadvantage of being underfitting in edge cases. Interactive learning could solve the underfitting of edge cases of support vector machines. By asking an oracle to provide labels for edge cases, the chance that edge cases are annotated increases. The edge cases will also be more prevalent in a model's training data, which could prevent underfitting on those edge cases. These advantages of IML will be explained later.

To distinguish between classes, the correct hyperplane has to be found. Around the hyperplane, a margin is formed. The margin is the distance between the hyperplane and the nearest data point from the hyperplane. Data points on the margin are called support vectors. The optimal hyperplane divides data in the middle between most data points so maximizing the margin gives the optimal hyperplane. To calculate the hyperplane, the following formula is applied: $w^T x - b = 0$ with w being the normal vector to the hyperplane, x being the data of a data point and b being the offset of the hyperplane. Maximizing the width of the margin is done with $\max(2/(||w||))$ and optimizes the hyperplane. With non-linear classifiers, the dot product of $w^T x - b = 0$ is replaced with a kernel function. The kernel function of RBF is $k(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$ and $\gamma > 0$ with γ being a parameter for $\gamma = \frac{1}{2\sigma^2}$ with σ being optimized by finding the maximal width of the margin. The polynomial kernel function is $k(x_i, x_j) = (x_i \cdot x_j)^d$.

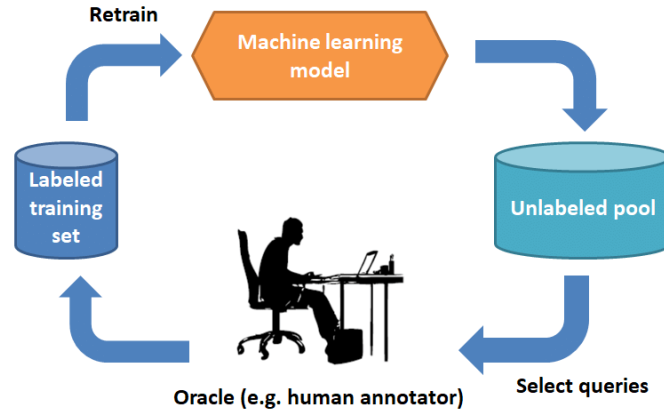


Figure 2.7: The process of interactive machine learning. Ambiguous/uncertain samples are selected for oracle to incrementally augment the training set. Reversed Active Learning based Atrous DenseNet for Pathological Image Classification - Scientific Figure on ResearchGate.

2.5 Interactive machine learning

Interactive learning, interactive machine learning (IML) or active learning is a machine learning technique that utilizes a user labelling data interactively during the training cycle of the algorithm. This user is also called an “oracle”. In this study, the oracle is simulated to improve the validity of comparison with other ML techniques. The purpose IML serves is to make the annotating of data more efficient, saving the user time and effort in the training process of the algorithm. The classification model in IML is also called the “learner”.

Three methods of supplying data (learning scenarios) to an interactive learning active algorithm exist (1) membership query synthesis, (2) pool-based sampling and (3) stream-based sampling. Membership query synthesis is a method where the user provides the learning data. Alabdulmohsin et al. (2015) trained an SVM to learn on data provided by the user. Pool-based sampling is a strategy where the entire training data set is evaluated and the data is presented to the user. This way the training cycle is the most efficient. In Stream-based selective sampling, a stream of data is sent to the learner and the learner has to decide, based on the IML algorithm, whether to ask the oracle to label the data. This is evaluated one data point at a time.

There are various methods to determine which data should be presented to the user (sampling strategies). In this study, three sampling strategies are compared: (1) uncertainty sampling, (2) query by committee (QBC) and (3) random sampling.

Uncertainty sampling is a technique that is based on choosing edge data: data that holds the lowest classifying certainty, predicted by the model. It is hypoth-

esized that labelling the least certain data will help the model have the fastest increase in terms of accuracy, while at the same time decreasing overfitting. Uncertainty sampling is also a very efficient technique to use in conjunction with a support vector machine. It is simple to find the data point with the lowest margin with the hyperplane.

With the query by committee sampling strategy, Freund et al. (1997), other classification models give a probability of the classes of the data. The data where the difference in class probability between the IML and the committee is highest is presented to the user. Class probability meaning the probability an algorithm gives for data belonging to a certain class. The algorithms chosen for the committee are used to evaluate this difference.

Random sampling is used as a baseline for comparison; instead of evaluating which data must be queried to the oracle, random data is selected to be labelled.

There are three subsets of a training dataset in interactive machine learning.

Every training cycle t , the data of the images I in the training set are split into three subsets. I_K, t is the subset of data that is known, this is data that in this research, the support vector machine is already fitted to. Both the data and labels are known in this subset. I_C, t is the subset of the data that is chosen to be presented to the oracle, after the oracle has provided the labels for this data it can be added to I_K, t and the next training cycle can begin. I_U, t is the unknown set from this set. In the next training cycle, new data will be added to the chosen set.

IML allows the user to have insight into the accuracy of the algorithm. As a result, annotating more data than is necessary to achieve the desired accuracy can be prevented. This is an advantage that is unique to IML.

Chapter 3

Related Works

Both the field of interactive machine learning and its application on art are not novel developments. Li & Chen (2009) created a tool to assess the quality of digitisation of paintings. Strezoski et al. (2020) and Frost et al. (2019) created IML systems for art recommendation. Previous research can be used to build a solution to interactively classify art. The domains drawn inspiration from in this study are: computer vision, interactive machine learning and supervised machine learning. This section will cover the previous research in those domains that have inspired and enabled this study.

Omniart is a large database containing images of artworks and related meta-data such as artist, era and iconographic style. This dataset can be and has been used to create a classifier on features such as style and era, Strezoski & Worring (2018). WikiArt, as described by Mohammad & Kiritchenko (2018) is a comparable database of images of visual art. The work of Mohammad & Kiritchenko (2018) was used by Rakovitsky & Knott (2020) to evaluate an algorithm on being able to recognize emotions. The algorithm designed by Rakovitsky & Knott (2020) was trained on the FER-2013 dataset as described by Giannopoulos et al. (2018).

Convolutional neural networks are prevalent in the field of computer vision LeCun et al. (2010). The CLIP algorithm, Radford et al. (2021) uses both a convolutional neural network to classify images and a vision transformer model, which is more time-efficient. Computer vision with convolutional neural networks is put to test in various studies. One of the more popular applications is in self-driving vehicles Ouyang et al. (2019) where convolutional neural networks recognize objects, traffic signs and other vehicles to provide autonomous transport. Important research in the domain of art and computer vision by Tan et al. (2016). The study tackled the problem of conflict situations in abstract painting classification. Tan et al. (2016) trained a deep network to recognize both high-level and low-level features of paintings to classify them correctly. For example, the painting of a pipe



Figure 3.1: A painting of a pipe, which is not an image of an actual pipe

in figure 3.1 is classified as a painting of a pipe instead of an actual pipe. The network was trained on the same WikiArt dataset used in this study.

For the classification of paintings, previous research has provided a framework to classify any image based on a natural language word, Radford et al. (2021). CLIP can also be used to extract features from images to use with other algorithms by using a Vision Transformer as described by Dosovitskiy et al. (2020).

Support vector machines (SVM) can be trained to classify visual, numerical and other data Noble (2006). Computer vision with SVM has been researched by Lin et al. (2011) and Goh et al. (2001). Lin et al. (2011) trained an SVM on a large dataset and achieved an accuracy of 52.9% in classification accuracy. In contrast to convolutional neural networks, SVM is more suited to allow for binary or lower class classification Mathur & Foody (2008).

The application of Interactive machine learning in computer vision is not a novel domain. IML employing an SVM has been used to classify images by Li et al. (2004). This method is capable of utilizing the advantages of interactive machine learning Fails & Olsen Jr (2003) because SVM offers good integration sampling methods like uncertainty sampling. Li et al. (2004) have found IML able to save a data annotator time in the training process. IML has been applied to art to develop an art recommendation system that learns a user's preference Strezoski et al. (2020). On the other hand, Frost et al. (2019) used interactive machine learning to create an interactive tool for a user to find visual art that has

similar low-level features but find images that won't be liked by the user based on metadata but will be liked based on visual data.

As pointed out earlier, many models are available for classifying images. Comparison between these models provides a knowledge gap since it has not been completely realized.

Chapter 4

Methods

To allow for reproducibility, the methods applied to answer the research questions are outlined here. Firstly, data processing is described. Then follows an explanation of the process of embedding all selected images using a vision transformer, arguing that it is necessary to efficiently compare different algorithms on the previously processed data. Thirdly, the algorithm will be trained on the training and tested on the testing dataset and the accuracies of the algorithms will be compared.

The goal of this study is to provide an efficient tool ¹ for classifying visual works of art. This is useful since the annotation of art is a time-consuming task. IML can provide a way to decrease the need for big data. Interactive machine learning has been tested on its capacity to accomplish this task. Classifying a portrait or landscape provides a relatively simple experimental task enabling a controlled comparison of IML to other algorithms. These other algorithms are the CLIP classifier and a supervised support vector machine.

The CLIP classifier was chosen as a baseline because it includes a versatile image recognition algorithm that uses text as its input. It can classify images on labels it has not observed during training (zero-shot learning) Xian et al. (2018). Another advantage of CLIP is its ability to both preprocess and classify images. For image preprocessing, parts of different convolutional neural networks can be used as well. However, these methods do not yield the advantage of providing an easily adjustable model for both preprocessing and classification. The other algorithm used to test IML is a supervised SVM. Training an SVM on the entire training set in one training cycle can provide an insightful comparison with different IML sampling techniques. The presently employed sampling techniques are uncertainty sampling, query by committee and random sampling.

¹The project code can be downloaded at <http://github.com/rubenahrens/ArtThesis/tree/master/IMLart>

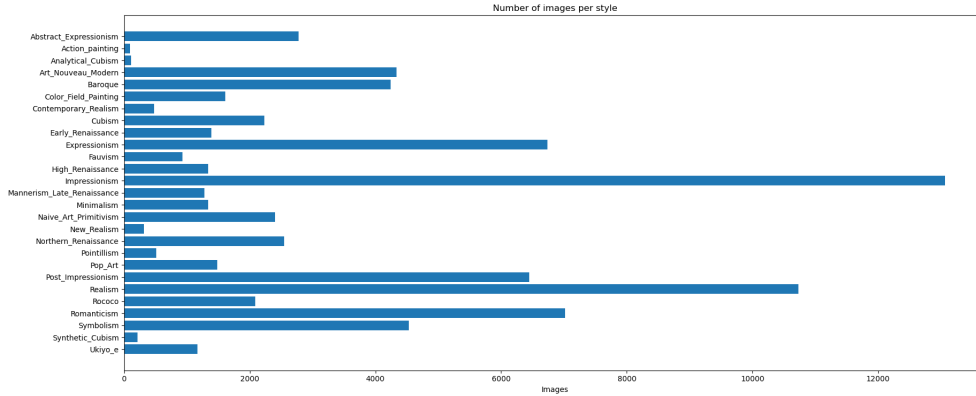


Figure 4.1: Number of images per label.

4.1 Data

The data used in this study are derived from the WikiArt database². The dataset has a size of 25.6 GB and contains a total of 81443 images of visual artworks stored in the .jpg format. These artworks are paintings, photographs and statues in the following styles:

The styles of the paintings could be seen in the folder path of the image, as the images were structured in folders based on style. Apart from the styles, the WikiArt database contains no additional metadata. This lack of metadata necessitated annotation of the images, making an interesting test case for IML as the purpose of IML is to minimize the amount of data to be annotated. Only the portraits and landscapes will be evaluated by the algorithms to simplify the task. The abundance of portrait and landscape paintings makes WikiArt suitable for the goal of classifying portraits and landscapes.

4.2 Data preparation

Firstly, before preparing any of the data, it was in the interest of data referencing to give all images an ID. WikiArt data were simply images in subfolders based on style so first, all 81443 images were given an identification number. The goal of the study is to find a way to improve classification accuracy with a low amount of data by using interactive machine learning. To achieve this goal, interactive learning has been used to distinguish a portrait from a landscape painting in the WikiArt

²The WikiArt database can be downloaded at: <http://github.com/cs-chan/ArtGAN/tree/master/WikiArt>

dataset. To be able to train and test machine learning algorithms at this task, data has to be annotated by a human expert. Since annotation of all 81443 images was unfeasible, a random selection was made. The entire dataset had to consist of at least 500 portraits and 500 landscapes to get a balanced dataset. Because landscape paintings were much less prevalent in the original dataset, at around 1000 images labelled, there were twice as many portraits as landscapes. The data had to be balanced by adding more landscape paintings. The SVM trained on the unbalanced dataset was used to select images that were likely a landscape. These images were presented to the user for labelling, they were added to the labelled dataset until there were at least 500 landscapes to allow for a balanced dataset. A balanced dataset improves machine learning performance because underfitting on sparse data can be prevented.

Random samples were selected from the dataset until the dataset contained 500 portraits and 500 landscapes. 1610 have been annotated with the labels: portrait, landscape or other. These annotations have been made based on definitions of portraits and landscapes. A portrait is a painting where a human being has been portrayed recognizably. Both facial expressions and posture are important to recognizing a portrait. When using this definition, images of ideas like the Virgin Mary or Jesus Christ are portrayals of ideas, not people. An example of this can be seen in figure 4.2.

Therefore these images have not been classified as a portrait. A landscape is defined to be a stretch of land that stands on its own. A cityscape or seascape is not classified as a landscape. 1070 of those 1610 images have been annotated as being a portrait or a landscape image. The other category will not be used in training and testing the algorithms. Even though the data remains available to increase the number of classes in the training set for future work. Out of the 1070 paintings, 570 of them are portraits and 500 of them are landscapes. The 1070 images have been split into a training and test set. 25% of the data being the test set and 75% being the training set. This split ratio is recommended by sklearn as it is the default ratio for its train test split function. After removing the paintings classified as “other”, the training set contains 802 images, the test set contains 268 images. Removing the “other” category simplifies the problem and allows for a better comparison between

To enable an efficient annotation workflow a fit for use graphical user interface was designed and implemented. The interface shows the image with buttons above it. The buttons are “Portrait”, “Landscape” and “Other”. These buttons automate saving the annotations in a .csv file, saving the annotator time and simplifying the annotation process. The user interface can be seen in figure 4.4. The user interface was made using the Tkinter Python module. A module made for making the process of implementing a user interface in python more efficient. The reasons



Figure 4.2: The last supper, not a portrait but the painting of an idea. By Juan de Juanes - [X], Public Domain



Figure 4.3: A self-portrait by Rembrandt - www.mauritshuis.nl : Home : Info : Pic, Public Domain

Sets	portrait	landscape	other	total
Training	422	380	0	802
Test	148	120	0	268
Other removed	570	500	0	1070
Total	570	500	540	1610

Table 4.1: The content of the data of the training set, test set, total and total without other categories except landscapes and portraits.

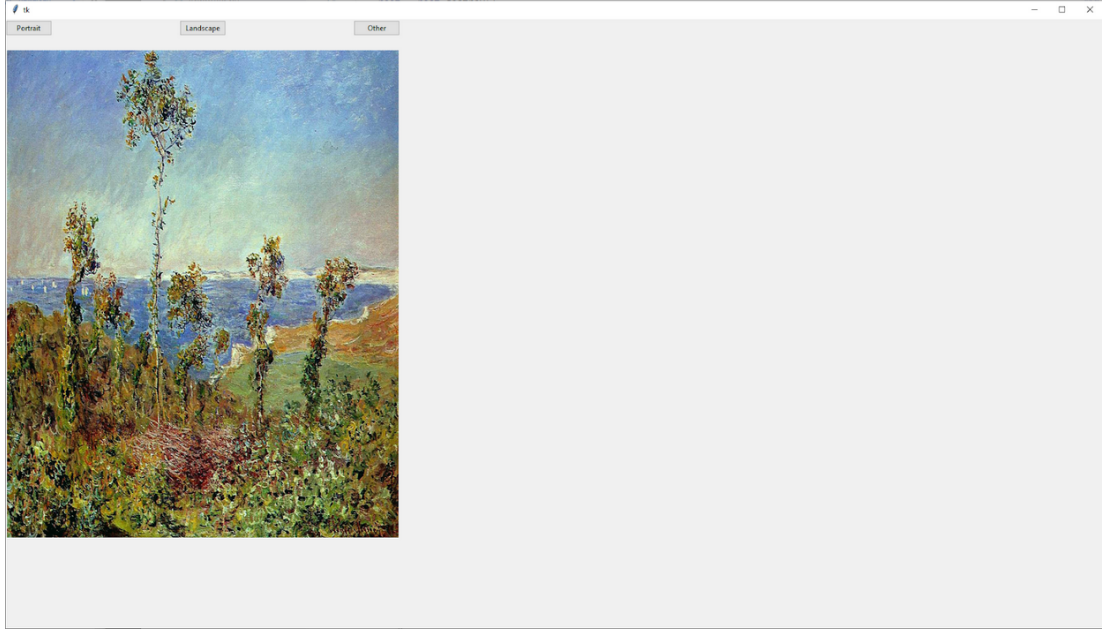


Figure 4.4: Art annotation graphical user interface

for making a graphical user interface is to make annotating faster and to make a program in which could be used for non-simulated interactive machine learning later on. In this program, the queried images will be shown to the user alongside the model’s current accuracy score on a test set.

Annotations made with the user interface are saved in the Annotation.csv file containing the path to the image, its id in the original dataset (id.csv) and the annotation. The annotation labels are -1, 0, 1, 2; -1 stands for not annotated. 0 stands for a Portrait, 1 for Landscape and 2 for “other”. The train and test datasets will not feature either the other or the unannotated category. The code for this user interface is available in the appendix. With this work, the data preparation has been finished. The next step in testing the feasibility of IML is processing the data of the images. Currently, the labels are available, a ML algorithm needs both

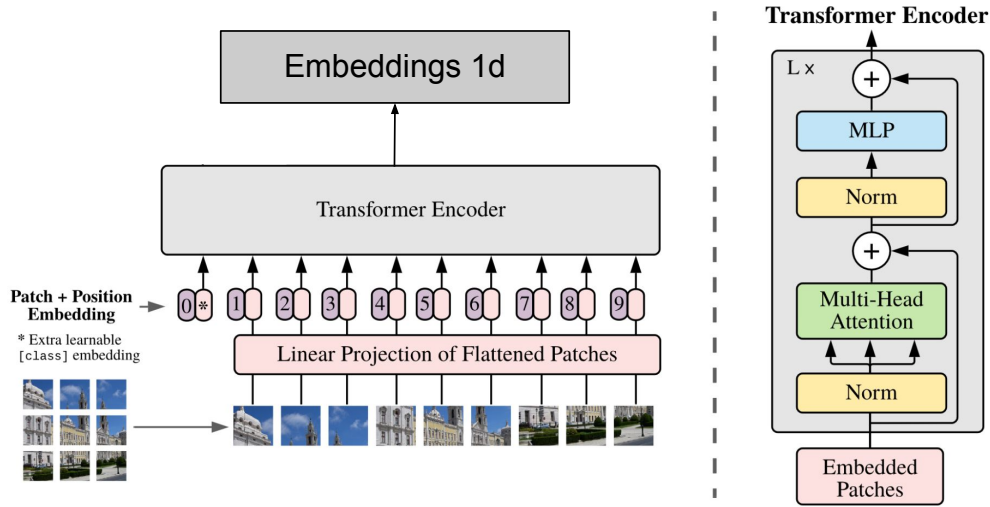


Figure 4.5: How image features are extracted using CLIP.

data and labels.

4.3 Getting image transformations with CLIP

The work of CLIP has been used to obtain features of all images to be later used with support vector machine classification and interactive machine learning. Both the images in the training and test set will be transformed into one dimension using the ViT incorporated in CLIP. The workings of this ViT are explained in the background section and portrayed in figure 4.5 above. CLIP features a function to simplify encoding these images with a ViT. This function works as the ViT described in Dosovitskiy et al. (2020). This function has been used to extract the image features to be used with the other algorithms. To enable this, all images in the training and test set are preprocessed and the features are saved to a .npz file. Both the training and test set has a .npz file with the image embeddings with indices corresponding to the indices of the .csv of the training and test set. The image embeddings were not saved in the .csv files to keep the CSV's readable and because .npz is more suitable for saving a matrix. The content of the .npz files are matrices with the image embeddings for one image in every row. These features will also be used with a support vector machine and an interactive support vector machine along with the CLIP classifier itself.

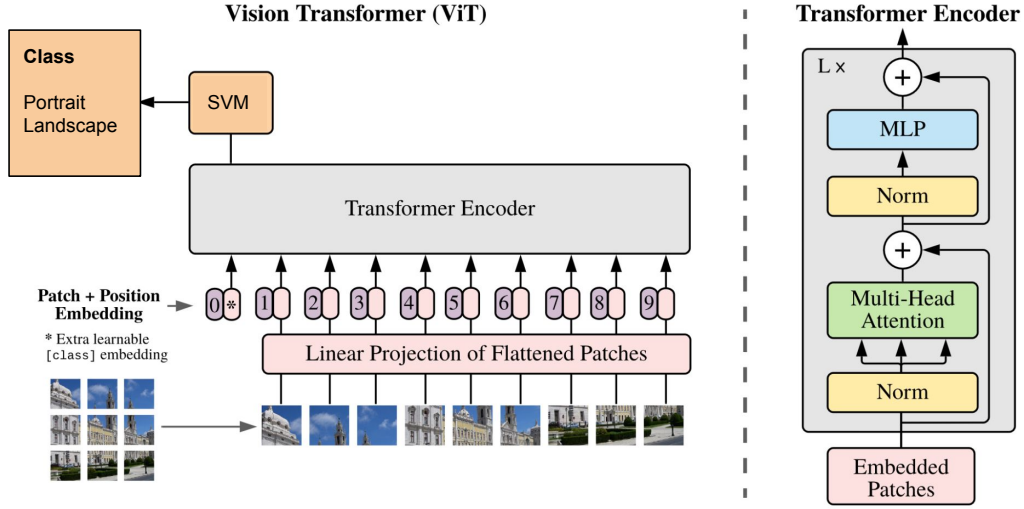


Figure 4.6: SVM pipeline used in this study

4.4 Classifying images with CLIP

The first step in the procedure of evaluating interactive learning is to provide a baseline with an existing method. For this purpose the CLIP model, Radford et al. (2021) will be used. CLIP is a model trained on millions of images and can classify based on any word. It uses text encoding and ViT for its classification. CLIP was used by iterating over all images. The model used is the CLIP “ViT-B/32” model, which is the only ViT model available in the CLIP python module, ResNet models are also available. Radford et al. (2021) found ViT models to have better performance compared to ResNet models. The clip model is used with a softmax activation function to get the probabilities of every class. The class is put in with text, that text is processed to find a similar label in the CLIP model. The output of the softmax function is a list with the probabilities of every class, the index of the highest probability is the label of the predicted class. These predictions of the test set are compared with the annotated features getting an accuracy score.

4.5 Support vector machine classification

The support vector machine classification was made possible by using the sklearn SVM method. Preparing the data properly for the algorithm was important. The SVM has a radial basis function kernel by default which was left unchanged. The

input data has been prepared using CLIP’s preprocessing algorithm. Transforming a 2-dimensional image into a 1-dimensional array. This array is used in the support vector machine. The label array is taken from the training dataset. These labels are the result of annotation that was done by a human expert. Label values are either 0 or 1. Zero being a portrait, 1 being a landscape. Using these two features exclusively serves to simplify the problem and provide a better situation to compare interactive, with non-interactive learning on this problem of classifying visual art. With supervised learning, the model is fit on the training data. The model is then used to predict features of the images in the test set, using the same preprocessing as has been done on the training set. The predictions are binary. 0 for portrait, 1 for landscape. Exactly the same data type as the annotations.

4.6 Interactive learning machine learning

To more efficiently study the effect of interactive learning the results have not been gathered by actually having a human oracle label the data. Instead, the oracle is simulated by making use of the previously annotated data in the training dataset. When the oracle queries an annotation, the annotation will be added to the model from the training dataset. When applying the IML algorithm made in this study to new data and labels, the UI made for annotating could be directly used for non-simulated IML. After choosing how the data should be labelled, a sampling and later a query strategy has to be chosen. The data is sampled with pool-based sampling, which means that the whole training dataset will be evaluated at once by the chosen query strategy. Interactive learning was compared between multiple query strategies; uncertainty sampling with three different offsets, query by committee and random sampling. Every training cycle (iteration), the model is used to predict the labels in the test set to be able to view the progress the model has made in terms of accuracy. The accuracy progress of the different algorithms can be seen in the results section. For interactive learning, there is a subset of the training implemented set called the fit set. This is the set of images presented to the oracle, defined as $I_{(C,t)}$. For the interactive learning algorithm, the training set is a set of which the labels are not known. This can be viewed as the memory of the simulated human: the oracle. The data is not known to the learner, (the learning algorithm). But it is known for the oracle (the larger machine learning algorithm) which is simulated in the study but is normally the user. A problem encountered in earlier versions is that data can be duplicated in the interactive training set also called the fit set in the pseudo-code. This caused the model to overfit onto data that is not representative of the entire dataset. With every sampling strategy, a parameter was added to decide how many images had to be presented to the oracle. The parameter for this is alpha. The model was tested with an alpha of 3

and 5 to compare the effect of adding a lot of data every iteration to adding fewer data.

4.6.1 Uncertainty sampling

The first strategy was uncertainty sampling. The 5 most uncertain samples were presented to the simulated oracle to be annotated and added to the model. Three variations of uncertainty sampling were compared. Each having a different offset, a parameter that decides how many data points to skip when sorting all predictions from the most to the least uncertain. With an offset of 0, the 5 most uncertain predictions are added to the set $I_{(C,t)}$, the set of data points that are chosen to be presented to the oracle. The second strategy is using an offset of 20 with uncertainty sampling. Skipping the 20 most uncertain predictions could land a better result by mitigating the risk of the model overfitting on bad data. An offset of 50 is also used.

4.6.2 Query by committee

Query by committee has been done by using the CLIP algorithm as a committee. Clip classifies based on probabilities of an image belonging to a certain class and will thus output two probabilities per image, converting this output to the same data type as the probabilities the SVM model can output the probabilities can be compared. The 5 images where the probabilities from CLIP and SVM differ the most will be presented to the oracle, as long as they have not been presented yet. When disregarding whether the images have already been annotated the results will be far different for query by committee and the model will end up overfitting on data that may be falsely classified by CLIP.

4.6.3 Random sampling

The sampling techniques have to be compared to a baseline to compare them validly. To be able to prove the validity of using one of the query strategies of IML it has to be compared to a baseline. Random sampling implies the absence of a predefined query strategy. If random sampling were to perform better than either one of the sampling strategies, it would not be necessary to spend computation power on evaluating what images must be queried to the oracle. The learner would be better off by randomly selecting data, querying it to the oracle and letting the user decide when the algorithm has a high enough accuracy and thus enough data to achieve this accuracy. Random sampling was chosen as a baseline for this reason.

Chapter 5

Results

In this section, the results of the experiments described in the methods section are presented. The results were measured in terms of accuracy, which is defined as the percentage of correct predictions $accuracy = \frac{C}{A}$, C is the number of samples classified correctly, A is the total number of samples. All accuracies were tested on the test dataset. A dataset containing 268 paintings, 148 portraits and 120 landscapes. The task for the models was to identify the paintings correctly and to classify them either as portrait or landscape.

Firstly, the results for the accuracy of the CLIP model and a supervised SVM are presented in table 5.1. The CLIP model classified the images in the test set with an accuracy of 86%. The CLIP model was not trained on the training set, making it a zero-shot approach. After having trained a supervised SVM with an RBF kernel on the training set, the support vector machine classified 99.6% of the images in the test set correctly.

Secondly, the results of interactive machine learning are presented in table 5.2. These results can be compared with table 5.1, as the models were tested on the same test set and had the same classification task. The accuracy per training iteration for interactive machine learning is plotted in figure 5.1. Some parameters were added to find an optimal method of using IML. The parameter alpha decided how many samples will be added to the known dataset for the learner, and automatically how many images had to be annotated by the user

algorithm	accuracy
CLIP	86 %
Supervised SVM	99.6 %

Table 5.1: Accuracy of the CLIP classifier and the supervised support vector machine on the test set.

Alpha	Sampling Technique	Offset	Accuracy	Stable Iteration	Stable Images
5	Uncertainty Sampling	0	99.6	35	175
5	Uncertainty Sampling	20	99.6	39	195
5	Uncertainty Sampling	50	99.6	26	130
5	Query By Committee	0	99.6	68	340
5	Random Sampling	0	99.6	74	370
3	Uncertainty Sampling	0	99.6	62	186
3	Uncertainty Sampling	20	99.6	140	420
3	Uncertainty Sampling	50	99.2	263	789
3	Query By Committee	0	99.6	116	348
3	Random Sampling	0	99.6	137	411

Table 5.2: Comparing interactive machine learning query techniques. Stable iteration is the iteration where the accuracy is stable as can be seen in figure 5.1. Stable images is the number of images that have been annotated before the final accuracy has been achieved.

every training cycle. The accuracy in table 5.1 and 5.2 is the accuracy after all images in the training set have been fitted to the model.

In figure 5.1, the accuracy on the test set in every training iteration was plotted for every sampling technique. Two parameters were added to the IML algorithm that changed the model training: alpha and offset. Parameter alpha decided how many images were presented to the user. Equivalent to adding data to subset I_C, t as described in section 2.5. The larger alpha is, the more images were shown to the user in every training cycle. Below the accuracy on the test set was plotted for both alpha 5 and 3. With uncertainty sampling, the parameter offset dictated what images were added to subset I_C, t . An offset of 0 meant that the 5 most uncertain images were shown to the oracle. An offset of 20 meant that from the 20th until the 25th most uncertain images will be shown.

The best performing technique was uncertainty sampling with an alpha of 5 and an offset of 50. After annotating 130 images, the IML algorithm had an accuracy of 99.6% on the test set. This means that the offset 50 approach is more efficient than uncertainty sampling with an offset of 20 or 0. Uncertainty sampling with an offset of 0 and query by committee appears to be the only approaches that were not largely affected by a change from alpha 5 to alpha 3. In general, an alpha of 3 resulted in more images having to be annotated.

With an alpha of 5 in figure 5.1, the only sampling technique displaying a different performance on the test set was query by committee. The "committee" used with the QBC sampling strategy in this study is the CLIP algorithm. The probabilities CLIP put out for the images in the training set belonging to the class

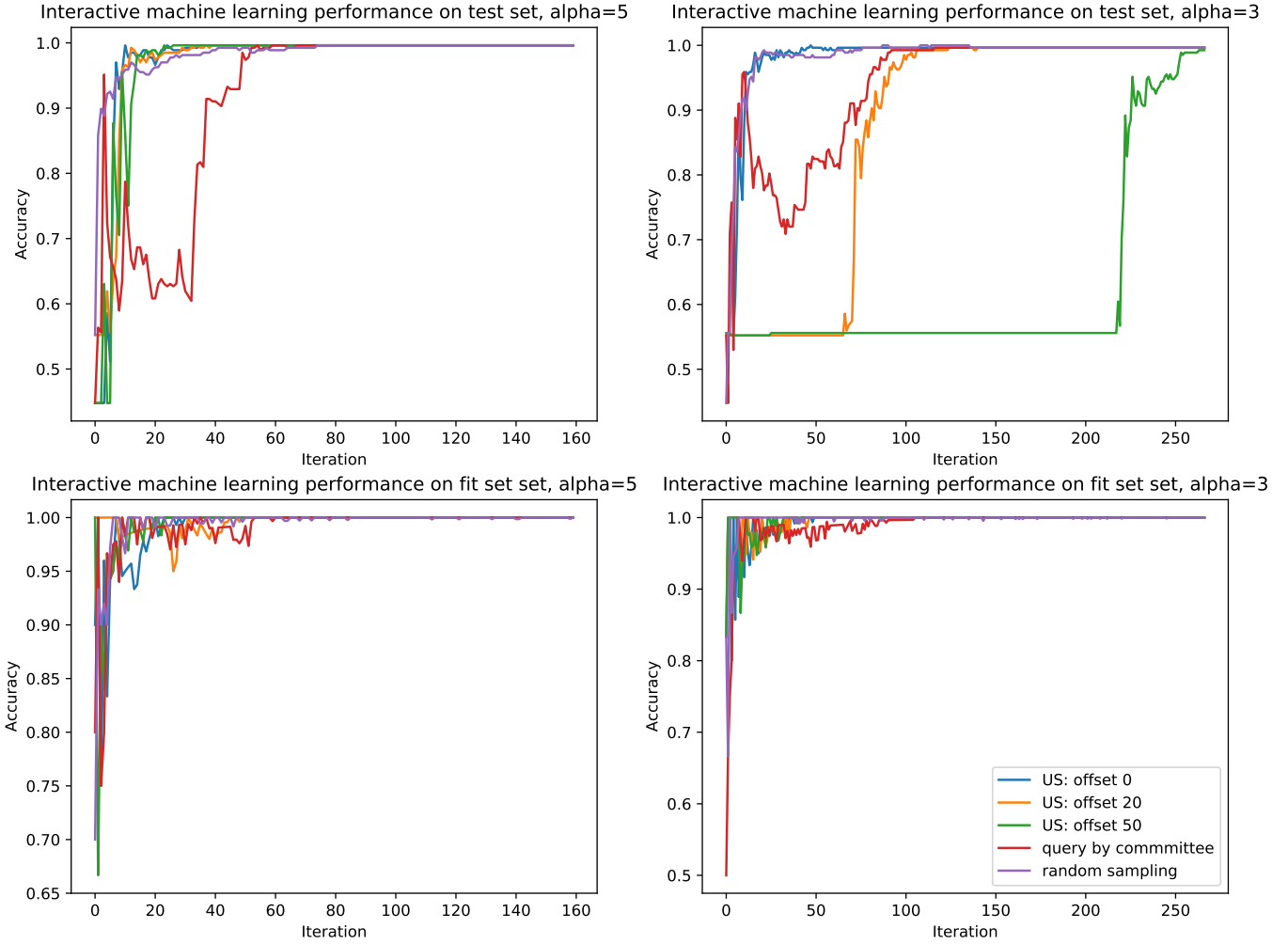


Figure 5.1: accuracy for each training iteration of the different interactive learning methods. Offset 0 being uncertainty sampling using an offset of 0: sampling the 5 most uncertain predictions every iteration. Offset 20 being uncertainty sampling, sampling the 5 most uncertain predictions, skipping the first 20 most uncertain, meaning the 20th until the 25th most uncertain predictions. Query by committee samples the 5 most disagreed predictions as determined by the CLIP algorithm.

of either portrait or landscape were used to decide which image to add to the fit set. Images that showed the largest difference in terms of probability from the SVM and CLIP were added to subset I_C, t and after annotation by the oracle, the images and their labels were added to I_K, t . To better understand how the accuracy of QBC could lag behind the other sampling techniques, the accuracy of each model upon the fit set was plotted for every iteration (the lower two graphs). The final accuracy of QBC on the fit set does not differ significantly from the other algorithms. This could mean that at the beginning of the whole training process, the model overfits on data either incorrectly classified by CLIP, or data that is not well represented in the test set. With an alpha of 5, random sampling and uncertainty sampling appear to follow the same trend when it comes to accuracy increase per iteration on the test set for every offset.

The difference between offset in terms of uncertainty sampling becomes more clear with an alpha of 3. Because only 3 images were shown to the oracle when alpha is 3, an offset of 20 is relatively more than when alpha is 5. With an alpha of 5, relatively less of the most uncertain samples were skipped per iteration. Meaning, with an offset of 20 it took 7 iterations to possibly reach the most uncertain samples with an alpha of 3 while it took 4 iterations to reach this point with an alpha of 5. With an offset of 0, the most uncertain samples can be added immediately which speeds up learning in the beginning phase.

User feedback was incorporated into training by adding the labels and images that were annotated by the oracle to subset I_K, t . The model was fit to this “new” data. In this study, the data was simulated as being new as the oracle of IML was simulated in this study, the images were annotated in advance. What images were presented to the user was determined by the sampling strategy. Uncertainty sampling proved to be the best strategy, the reason will be further explained in the discussion section.

Chapter 6

Conclusion & Discussion

6.1 Conclusion

In this study, the feasibility of IML was researched. To put this feasibility to the test, three sampling strategies of IML were compared to a supervised SVM, a zero-shot image recognition algorithm (CLIP using a ViT). The research questions proposed in the introduction, section 1 will be answered. The main question is: How can an interactive classifier be trained to classify visual art? To answer this question, the following sub-questions have to be answered:

- How can user feedback be incorporated into the training?
- Which images should be presented to the user and how?
- How does an interactive classifier compare to its supervised learning counterpart in terms of accuracy when being trained on the same number of samples?

IML provides better insight into machine learning performance compared to a supervised ANN because it combines the training process with the annotation process, allowing people in need of a classifier a more efficient data annotation process. The performance of IML is better compared to a zero-shot alternative because it allows for a training process with data that is more represented in the test set. When the desired accuracy is achieved, the annotation process can be terminated, preventing annotating more data than necessary.

The interactive classifier was trained by incorporating user feedback into the training process. The user feedback was used as training data for the IML model. Uncertainty sampling proved to be the best approach for selecting images that were presented to the user. When trained on the same number of samples, the IML

classifier achieved the same accuracy on the test set as its supervised counterpart but this accuracy was reached faster with IML.

In conclusion, the study proves that an interactive classifier can be trained to classify visual art, providing an alternative for art historians wanting to save time in the data annotation process.

6.2 Discussion

In this section, the key findings will be discussed. The results will be compared to findings in previous studies. Finally, the limitations of the results will be explained and recommendations for future research are given.

As hypothesised, an interactive machine learning classifier achieves similar or better results as a supervised classifier, given the IML classifier has been trained on the same amount of data.

All accuracies on the test set have been stabilized after about 150 iterations with 3 images per iteration, with uncertainty sampling with an offset of 50 being an exception. With 5 images per iteration, all accuracies on the test set have been stabilized after about 80 iterations, figure 5.1 and 5.2. When unsimulated, interactive machine learning could provide a benefit over other supervised methods by giving insight into the accuracy of the model while annotating the images. Unexpectedly, random sampling trumped uncertainty sampling and query by committee in the early stages before stabilisation.

With random sampling, early overfitting on uncommon data was prevented, giving it a faster learning rate in the first 5 training iterations with an alpha of 5. This is caused by one of the study’s limitations. Because of the binary classification task, the probability of random sampling choosing an image that enables fast learning is higher than with a multi-class problem. The fact that there were only two classes made uncertainty sampling choose paintings that had similar probabilities for both classes. This caused the model to not be trained on and thus underfit on very distinctive portraits or landscapes. Most paintings in the test set are distinctively either a portrait or landscape which caused uncertainty sampling to have a lower learning rate compared to random sampling and query by committee in the first 5 training iterations with an alpha of 5.

In terms of stabilisation of the accuracy, however, uncertainty sampling with an offset of 0 and an alpha of 5 provided the fastest stabilisation. This resulted from a calculated approach. The learning rate was not as high as with random sampling, but the more calculated approach caused an earlier stabilisation as random sampling randomly selected images that caused a negative learning rate after iteration 20.

In this study, on the WikiArt dataset, the accuracy of the CLIP model was

86%. Compared to the zero-shot performance of CLIP on other datasets, this result is in the upper regions. Zero-shot accuracy on aYahoo was 98.4%, accuracy on ImageNet was 76.2%, accuracy on SUN was 58.5%.

Goh et al. (2001) reached an error rate of 19.7 on landscapes and an error rate of 15.8 on people. That is an average accuracy of 82.25 % on paintings and landscapes, a lower accuracy than the achieved 99.6 % in this study. This is caused by the algorithm used in this study being specifically trained on only portraits and landscapes and the algorithm by Goh et al. (2001) being trained on more than two classes.

The results are similar to results found for an artificial database by Li et al. (2004). Compared to the real database used by Li et al. (2004), the results achieved in this study are more accurate. In the work of Li et al. (2004) multilabel images were classified into four similar classes. This caused a lower accuracy compared to the two relatively distinct classes in this study.

In the future, uncertainty sampling is expected to be a superior performer on fine-tuning already trained algorithms looking for more obscure data. The exploitation of uncertainty sampling made it an extensive approach for IML, while random sampling struck a balance between exploration and exploitation. Even though random sampling performed well on this data, uncertainty sampling or a variation on the uncertainty sampling used in this study is expected to truly outperform random sampling on a more complicated dataset with more classes. For future research, it would be interesting to see IML being applied on a dataset of paintings with more than two classes. It is expected that IML will still provide the shown advantages with data annotation. Uncertainty sampling is expected to give even better performance as random sampling would have a lower probability of choosing the correct sample.

Bibliography

- Agarap, A. F. (2018), ‘Deep learning using rectified linear units (relu)’, *arXiv preprint arXiv:1803.08375* .
- Alabdulmohsin, I., Gao, X. & Zhang, X. (2015), Efficient active learning of half-spaces via query synthesis, *in* ‘Proceedings of the AAAI Conference on Artificial Intelligence’, Vol. 29.
- Albawi, S., Mohammed, T. A. & Al-Zawi, S. (2017), Understanding of a convolutional neural network, *in* ‘2017 International Conference on Engineering and Technology (ICET)’, Ieee, pp. 1–6.
- Carneiro, G. (2011), Graph-based methods for the automatic annotation and retrieval of art prints, *in* ‘Proceedings of the 1st ACM International Conference on Multimedia Retrieval’, pp. 1–8.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2020), ‘An image is worth 16x16 words: Transformers for image recognition at scale’, *arXiv preprint arXiv:2010.11929* .
- Fails, J. A. & Olsen Jr, D. R. (2003), Interactive machine learning, *in* ‘Proceedings of the 8th international conference on Intelligent user interfaces’, pp. 39–45.
- Freund, Y., Seung, H. S., Shamir, E. & Tishby, N. (1997), ‘Selective sampling using the query by committee algorithm’, *Machine learning* **28**(2), 133–168.
- Frost, S., Thomas, M. M. & Forbes, A. G. (2019), Art i don’t like: An anti-recommender system for visual art, *in* ‘Proceedings of Museums and the Web’.
- Giannopoulos, P., Perikos, I. & Hatzilygeroudis, I. (2018), Deep learning approaches for facial emotion recognition: A case study on fer-2013, *in* ‘Advances in hybridization of intelligent methods’, Springer, pp. 1–16.

- Goh, K.-S., Chang, E. & Cheng, K.-T. (2001), Svm binary classifier ensembles for image classification, *in* ‘Proceedings of the tenth international conference on Information and knowledge management’, pp. 395–402.
- He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* ‘Proceedings of the IEEE conference on computer vision and pattern recognition’, pp. 770–778.
- Hecht-Nielsen, R. (1992), Theory of the backpropagation neural network, *in* ‘Neural networks for perception’, Elsevier, pp. 65–93.
- Joshi, A. J., Porikli, F. & Papanikolopoulos, N. (2009), Multi-class active learning for image classification, *in* ‘2009 IEEE Conference on Computer Vision and Pattern Recognition’, IEEE, pp. 2372–2379.
- LeCun, Y., Kavukcuoglu, K. & Farabet, F. (2010), Convolutional networks and applications in vision, *in* ‘Proceedings of 2010 IEEE International Symposium on Circuits and Systems’, pp. 253–256.
- Li, C. & Chen, T. (2009), ‘Aesthetic visual quality assessment of paintings’, *IEEE Journal of selected topics in Signal Processing* **3**(2), 236–252.
- Li, X., Wang, L. & Sung, E. (2004), Multilabel svm active learning for image classification, *in* ‘2004 International Conference on Image Processing, 2004. ICIP’04.’, Vol. 4, IEEE, pp. 2207–2210.
- Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., Yu, K., Cao, L. & Huang, T. (2011), Large-scale image classification: fast feature extraction and svm training, *in* ‘CVPR 2011’, IEEE, pp. 1689–1696.
- Mathur, A. & Foody, G. M. (2008), ‘Multiclass and binary svm classification: Implications for training and classification users’, *IEEE Geoscience and remote sensing letters* **5**(2), 241–245.
- Mohammad, S. & Kiritchenko, S. (2018), Wikiart emotions: An annotated dataset of emotions evoked by art, *in* ‘Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)’.
- Noble, W. S. (2006), ‘What is a support vector machine?’, *Nature biotechnology* **24**(12), 1565–1567.
- Ouyang, Z., Niu, J., Liu, Y. & Guizani, M. (2019), ‘Deep cnn-based real-time traffic light detector for self-driving vehicles’, *IEEE transactions on Mobile Computing* **19**(2), 300–313.

- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J. et al. (2021), ‘Learning transferable visual models from natural language supervision’, *arXiv preprint arXiv:2103.00020* .
- Rakovitsky, A. & Knott, J. (2020), ‘Wikiart analysis using facial emotion analysis’.
- Ruder, S. (2016), ‘An overview of gradient descent optimization algorithms’, *arXiv preprint arXiv:1609.04747* .
- Rui, Y., Huang, T. S., Ortega, M. & Mehrotra, S. (1998), ‘Relevance feedback: A power tool for interactive content-based image retrieval’, *IEEE Transactions on circuits and systems for video technology* **8**(5), 644–655.
- Strezoski, G., Fijen, L., Mitnik, J., László, D., Oyens, P. d. M., Schirris, Y. & Worring, M. (2020), Tindart: A personal visual arts recommender, *in* ‘Proceedings of the 28th ACM International Conference on Multimedia’, pp. 4524–4526.
- Strezoski, G. & Worring, M. (2018), ‘Omniart: a large-scale artistic benchmark’, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* **14**(4), 1–21.
- Tan, W. R., Chan, C. S., Aguirre, H. E. & Tanaka, K. (2016), Ceci n’est pas une pipe: A deep convolutional network for fine-art paintings classification, *in* ‘2016 IEEE international conference on image processing (ICIP)’, IEEE, pp. 3703–3707.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017), ‘Attention is all you need’, *arXiv preprint arXiv:1706.03762* .
- Xian, Y., Lampert, C. H., Schiele, B. & Akata, Z. (2018), ‘Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly’, *IEEE transactions on pattern analysis and machine intelligence* **41**(9), 2251–2265.

Appendix A

Setup

The device used for classification is an Nvidia GTX 1070ti using Cuda and PyTorch. The Cuda version is 11.3.58. The PyTorch version is 1.7.1. All code was written in python 3.8 on a Windows 10 (64-bit) PC with 16GB. Other modules used are NumPy, TkInter, sklearn, cv2 and pandas. An overview of the applications used alongside their versions can be found in figure A.1.

Name	Version
Python	3.8
pyTorch	1.7.1
CUDA	11.3
Numpy	1.18.5
TkInter	8.6
Pandas	1.2.4
CLIP	1155CC
Sklearn	0.23.2
OpenCV	4.5.1

Table A.1: The content of the data of the training set, test set, total and total without other categories except landscapes and portraits.