TU WIEN Informatics

Transparency Techniques for Neural Networks trained on Writer Identification and Writer Verification

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieurin

im Rahmen des Studiums

Visual Computing

eingereicht von

Viktoria Pundy, BSc
Matrikelnummer 01633403

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Mitwirkung: Dipl.-Ing. Dr.techn. Florian Kleber
Univ.Ass. Dipl.-Ing. Marco Peer

Wien, 10. November 2023

Viktoria Pundy | Robert Sablatnig

TU WIEN Informatics

Transparency Techniques for Neural Networks trained on Writer Identification and Writer Verification

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieurin

in

Visual Computing

by

Viktoria Pundy, BSc
Registration Number 01633403

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Robert Sablatnig
Assistance: Dipl.-Ing. Dr.techn. Florian Kleber
Univ.Ass. Dipl.-Ing. Marco Peer

Vienna, 10th November, 2023

Viktoria Pundy | Robert Sablatnig

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Viktoria Pundy, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 10. November 2023

Viktoria Pundy

Danksagung

Diese Arbeit wäre ohne die unschätzbare Unterstützung vieler Menschen nicht möglich gewesen.

Ich möchte mich bei meinem Betreuer Robert Sablatnig für die Möglichkeit, diese Arbeit zu schreiben, sowie für seine Unterstützung und Hilfe während dieses Prozesses bedanken.

Besonders bedanken möchte ich mich auch bei meinen Betreuern Florian Kleber und Marco Peer, die mich immerzu unterstützt und in die richtige Richtung gelenkt haben. Euer Wissen, eure Hilfe und euer konstruktives Feedback haben mir den Mut und die Zuversicht gegeben, diese Arbeit abzuschließen. Außerdem möchte ich Manuel Keglevic für seine Hilfe bei den ersten Schritten dieser Arbeit danken.

Diese Arbeit wäre ohne meine Freunde und Familie nicht möglich gewesen. Vielen Dank an meine Thesis-Verbündeten Katja und Hannah - die Diskussionen über unsere gemeinsamen Erfahrungen während des Schreibprozesses motivierten mich stets, wenn es mir einmal nicht so leicht fiel, weiterzuschreiben. Ich möchte mich auch bei Oskar bedanken, der dafür gesorgt hat, dass ich nicht vergesse, dass Pausen zum Prozess dazugehören. Schließlich möchte ich mich bei meiner Mutter bedanken, die mich bedingungslos und kontinuierlich im Leben unterstützt hat, insbesondere auf meinem Weg in der Informatik. Ich danke euch allen, dass ihr diesen Weg mit mir gegangen seid.

Acknowledgements

This thesis would not have been possible without the invaluable support of a lot of people.

I would like to thank my advisor Robert Sablatnig for the possibility to write this thesis and for his support and guidance through this process.
I would also like to especially thank my supervisors Florian Kleber and Marco Peer, who consistently supported me and pushed me in the right direction when there were too many to choose from. The knowledge, help and constructive feedback you provided gave me the courage and confidence to complete this thesis. Additionally, I want to thank Manuel Keglevic for his help with the first steps of this thesis.

Finally, this thesis would not have been possible without my friends and family. Thank you to my fellow thesis companions, Katja and Hannah - our discussion about our shared experiences during the writing of the thesis motivated me when it was not so easy for me to continue writing. I would also like to thank Oskar, who ensured I would not forget that taking breaks is part of the process. Lastly, I want to express my gratitude to my mother, who supported me unconditionally and constantly in life, particularly on my path in computer science. Thank you all for going through this process with me.

Kurzfassung

Neuronale Netze sind der derzeitige Stand der Technik für viele Aufgaben in der Computer Vision [SBM+17, ZTLT21]. Dies inkludiert auch die Bereiche der Identifizierung und Verifizierung von Autoren, welche das Ziel haben, den Autor eines handgeschriebenen Textes zu identifizieren. Eine neue Domäne ist die Interpretierbarkeit neuronaler Netze, die als "Black Box"-Systeme bezeichnet werden, welche sich mit der Erklärung des Entscheidungsprozesses des neuronalen Netzes beschäftigt [SBM+17, HVH22, GMR+18, ZTLT21]. Diese Erklärungen werden verwendet, um die Leistung des Systems zu verbessern, mögliche Artefakte in den Trainingsdaten zu erkennen und die Zuverlässigkeit der Systeme in sicherheitskritischen Domänen zu erhöhen [SM19]. In dieser Arbeit werden zwei Transparenzmethoden auf neuronale Netze, die auf die Identifizierung und Verifizierung von Autoren trainiert wurden, angewendet. Die erste Methode generiert pixelweise Visualisierungen, die jedem Pixel einen Wert an Signifikanz zuweisen. Die zweite Methode generiert zwei Arten von Visualisierungen. Die erste stellt die Ähnlichkeiten zwischen zwei Bildern dar, während die zweite Ähnlichkeiten zwischen einem Punkt im ersten Bild und dem gesamten zweiten Bild darstellt. Das Ziel ist, forensische Experten mithilfe einer Visualisierung zu unterstützen, welche Informationen über Ähnlichkeiten in handschriftlichen Texten bietet. Des Weiteren soll untersucht werden, welche Charakteristiken ein neuronales Netz auswählt, um den Autor eines handgeschriebenen Textes zu identifizieren. Hierzu werden drei Architekturen von neuronalen Netzen, nämlich ResNet18, ResNet20 und ResNet50, ausgewählt, welche auf Methoden aus dem derzeitigen Stand der Technik basieren. Die Transparenzmethoden werden für die Anwendung mit diesen Netzen angepasst und anhand der Lösch- und Einfüge-Metriken evaluiert. Darüber hinaus werden die Techniken qualitativ evaluiert, indem die erstellten Visualisierungen von charakteristischen Bereichen mit den Bereichen, welche forensische Experten zur Identifizierung des Autors analysieren, verglichen werden. Die Ergebnisse der Evaluierung zeigen, dass die pixelweisen Visualisierungen bessere Ergebnisse als die punktspezifischen Visualisierungen bieten, deren hervorgehobenen Bereiche nur schwer einem bestimmten Buchstaben zuzuordnen sind. Die pixelweisen Visualisierungen zeigen ähnliche Muster in den hervorgehobenen Bereichen der verschiedenen Vorkommen des selben Buchstabens, was auf einen ähnlichen Analyseprozess, wie durch einen forensischen Experten durchgeführt, hinweist. Insgesamt besitzen die pixelweisen Visualisierungen Eigenschaften, die für die Unterstützung eines forensischen Experten geeignet sind, während die punktspezifischen Visualisierungen aufgrund ihrer missverständlichen Hervorhebungen nicht geeignet sind.

Abstract

Neural Networks are the state of the art for many tasks in the computer vision domain [SBM+17, ZTLT21]. This also includes the areas of Writer Identification and Writer Verification, where the goal is to identify the author of a handwritten text. A more novel task is the interpretability of neural networks, which are considered "black box" systems, and to provide an explanation for the decision process of the neural network [SBM+17, HVH22, GMR+18, ZTLT21]. These explanations are used to improve the system performance, reveal possible artefacts in the training data and increase the reliability of such systems in safety-critical areas [SM19]. In this thesis, two transparency techniques are applied to neural networks trained on Writer Identification and Writer Verification. The first transparency technique provides pixel-level saliency maps, where a significance value is assigned to each individual pixel. The second transparency technique provides two types of saliency maps, where one type shows overall similarities between two images and the second type displays similarities between one point in the first image and the overall second image. The goal is to support forensic experts with a visualization providing information on similarities in handwritten text inputs. Further, the thesis aims to explore the characteristics selected by a neural network to identify the author of a handwritten text. Three neural network architectures, namely ResNet18, ResNet20 and ResNet50, are selected based on methodologies proposed in the state of the art. The transparency techniques are adjusted for use with these specific networks and are evaluated using the deletion- and insertion score metrics. Furthermore, a qualitative evaluation is conducted, where the visualizations are compared to the areas forensic experts consider during the identification process of an author. The evaluation results show that the pixel-wise saliency map technique performs better than the point-specific saliency map technique, where the displayed highlightings are difficult to allocate to a certain character. The pixel-wise saliency maps display similar highlighting patterns for multiple occurrences of the same character, indicating a similar analysis process as applied by a forensic expert. Overall, the pixel-wise saliency maps display characteristics suitable for the support of a forensic expert, while the point-specific saliency maps are not suitable due to non-intuitive highlightings.

Contents

CHAPTER 1

Introduction

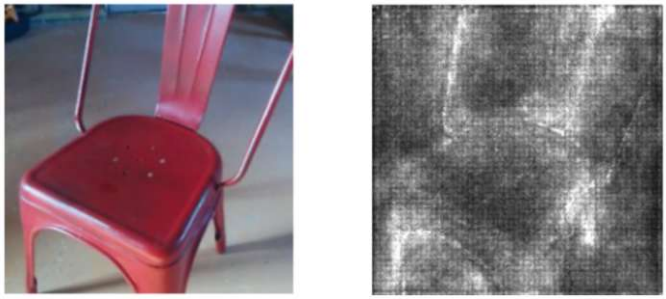
In the last decade, Deep Neural Networks (DNNs) have become a powerful method for various tasks in the computer vision domain, achieving high performances and outperforming previous machine learning methods [SBM+17, ZTLT21]. DNNs are now a standard for computer vision tasks and are becoming more complex regarding their computational effort [ZN20, ZXD17]. The area of computer vision also includes document analysis. One topic of document analysis is Writer Identification (WI) and Writer Verification (WV) [KFS18]. The goal of WI is to identify the author of a query handwritten text from a collection of known authors [TW16]. A WI system is trained to provide a ranking of the most similar handwritten documents, with the goal of listing the documents of the same author at the top [WMC21]. In contrast, the goal of WV is to determine if two given handwritten texts were written by the same author [Sch08]. People display different and unique characteristics in their handwriting, allowing for the identification of a person by their handwriting [BS07]. However, this variation in handwriting makes the identification of the writer a non-trivial task [PP21]. Moreover, the handwriting of one person is influenced by environmental circumstances such as the time taken for the writing process, the writing tool, and the geographic location and upbringing of the writer. Additionally, handwritings change over time. These differences in the handwriting of one person pose a challenge for the identification of a writer as well [ACB19, KFS18, Sch08].

In the past five years, the attention of researchers has been drawn to machine learning interpretability [SBM+17]. Machine learning systems, including DNNs, are considered "black boxes", which do not provide an explanation for the decision process responsible for producing the output for a given input [HVH22]. Although the parameters of a trained neural network are accessible, neural networks contain millions of them, therefore making a manual analysis infeasible [SBM+17, ZTLT21]. The proposed transparency techniques for machine learning interpretability aim to provide a comprehensible representation of the input features a neural network has selected to correctly process the given input and,

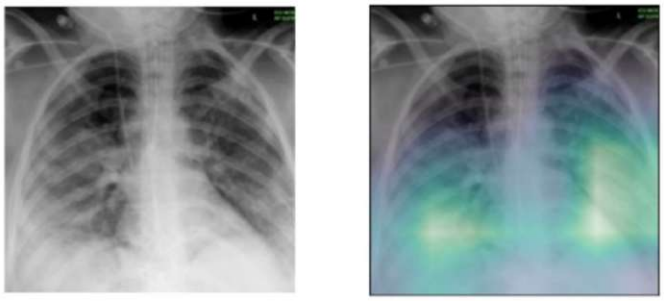
therefore, provide insight into the internal decision process [GMR+18, SBM+17, ZTLT21]. Examples of visual representations created by three different transparency techniques are shown in Figure 1.1. Figure 1.1a displays a pixel-wise highlighting of regions of interest, where bright areas are significant for the output of the underlying neural network. The exemplary visualization of Figure 1.1b shows regions of significance for a medical retrieval neural network, which classifies X-ray images as normal, pneumonia or COVID-19 cases. Figure 1.1c contains visualizations created by a transparency technique, which compares two images and their similarities. The highlights display areas, which are of importance for the similarity. In the literature, the terms *interpretability* and *explainability* are used interchangeably, which is also done in the scope of this thesis [GBY+18]. Furthermore, the term *transparency* is also used as a synonym for explainability.

Depending on the task at hand, the architecture of neural networks and their training configurations are adjusted to achieve the best possible result and performance. This results in various types of neural networks with their own characteristics. For example, one difference in neural networks is the type of loss function used, affecting the type of output a neural network produces. Two types are commonly used, classification- and ranking-based losses [KSDH21]. Neural networks trained with a classification loss provide a label representing a class to which the given input belongs. For example, in the scope of image classification, the label represents a certain category, such as ”cat” or ”boat”, depending on the classes the neural network has been trained on. The number of possible outputs is restricted: if a new class is added to the data, the network must be retrained. Neural networks trained with a ranking-based loss provide an n-dimensional embedding vector as output, which represents a point in n-dimensional space. Two images are considered to be of the same class if the output embeddings from these input images are positioned closely together [KSDH21]. Transparency techniques distinguish between neural networks with different loss types and are not necessarily applicable to all loss types. Depending on their explanation method, they are restricted to a certain type of loss, relying on certain characteristics of a neural network such as a classification module. For example, the Gradient-weighted Class Activation Mapping (Grad-CAM) method proposed by Selvaraju et al. [SCD+17] relies on gradients as weights, limiting its use to classification neural networks but not embedding neural networks [CCHM20]. Adjusting an underlying neural network to apply a certain transparency technique or adjusting a transparency technique to be applicable to all types of neural networks is not ideal. Zheng et al. [ZKC+20] show that adding a classification module to a model for the application of a transparency technique results in unintuitive representations of the contributions of the input features.

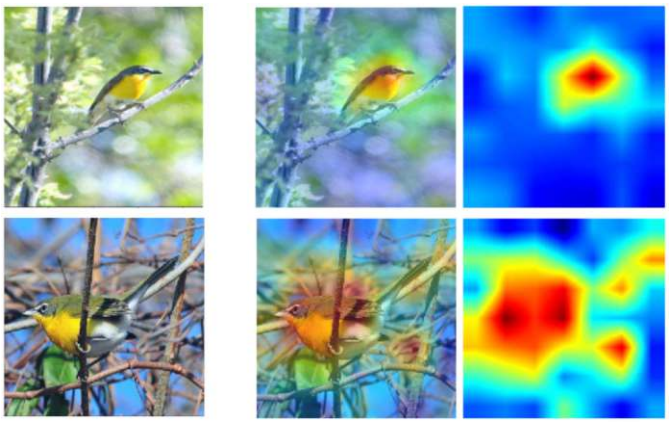
An explanation of a machine learning system is beneficial for several reasons. First, the system accuracy can be improved by using the insight into the system for the removal of biases and the detection of artefacts in the training process [SM19]. Machine learning systems are expected to use a valid strategy to execute their task successfully, i.e. using proper and generalizable features for the decision task [AWN+22, LWB+19]. However,



(a) Pixel-wise highlighting of regions of interest. Bright pixels in the visualization indicate areas which are of high importance for the output of the neural network. Images courtesy of Kobs et al [KSDH21].



(b) Region highlighting of regions of interest. Bright regions in the visualization indicate areas of high importance. Images courtesy of Hu et al [HVH22].



(c) Heatmap highlighting of regions of interest. The transparency technique provides information on the similarity between an image pair learned by the underlying neural network. Red regions indicate areas of high importance for the similarity. Images courtesy of Zhu et al [ZYC21].

Figure 1.1: Exemplary visualizations created by three different transparency techniques to highlight the significance of regions for the output of a neural network.

this behaviour is possibly compromised when spurious artefacts are present in the training process. For example, the method which won the PASCAL VOC challenge [EVGW+10] classified images of horses by a copyright mark, which is present in the horse images of the train set [SM19]. Another example is a system which used notes written on the training data by a radiologist to determine its output [HVH22]. Manual analysis of such datasets, however, is infeasible due to the large amount of data contained within [AWN+22]. Second, insight into a machine learning system is especially relevant for safety-critical areas such as the medical domain, where decisions impact human lives, or autonomous driving, where the technology makes decisions without human interference. Transparency of the used systems is a necessity in these areas [SM19]. Third, the General Data Protection Regulation (GDPR) states that "[...] people have the right not to be subject to an automated decision which would produce legal effects or similar significant effects concerning him or her. The data controller shall safeguard the data owner's right to obtain human intervention, to express his or her point of view and to contest the decision." ([ZTLT21], p.728). Transparency of a machine learning system supports this regulation by providing information on the processing of the given data and the corresponding output of the underlying machine learning system [SM19, WMR17].

In this work, transparency techniques are applied to neural networks trained for WI and WV. One goal is to gain insight into the trained internal decision process of neural networks for these tasks. Another goal is to gain an overview of the expressiveness and validity of the visualizations generated by the selected transparency techniques when using handwritten text images as input. Furthermore, the information provided by these transparency techniques allows to further analyse the results provided by the WI and WV neural networks. For example, an incorrect ranking provided by a WI neural network for a given handwritten text image can be analysed to gain information on the features used by the network. This information possibly exposes an incorrect use of available information, such as the use of additional data, which is not part of the handwritten text. Additionally, neural networks trained for WI and WV are, for example, used for the author identification of handwritten texts such as threat letters [KFS18]. The application of transparency techniques on such neural networks supports experts investigating the given handwritten text by providing information on important areas in the handwritten text and, therefore, expediting the analysis process.

1.1 Contribution

Previous work in the domain of transparency techniques mainly focuses on the applicability of transparency techniques on neural networks trained on computer vision tasks, where the input images are rich with information such as colours, intensities or multiple objects contained within the image. So far, transparency techniques have been used to analyse neural networks trained on tasks including, without limitation, image classification and image retrieval for both natural images [Rud94] and medical scans [HVH22, ZKC+20], as well as face recognition and person re-identification [ZYC21]. The use of transparency

techniques for neural networks trained on WI and WV has not yet been a focus of research.

This thesis explores the applicability of transparency techniques for neural networks trained on distinguishing different writers. The goal is to use transparency techniques to provide visual information on the input features selected by the neural network to successfully execute its task and retrieve information on the internal decision process of the neural network. For this purpose, two transparency techniques proposed by Kobs et al. [KSDH21] and Zhu et al. [ZYC21] have been selected from the state of the art. The performance of these transparency techniques on the WI and WV neural networks is qualitatively and quantitatively evaluated. Further, the goal is to provide forensic experts with a visualization which supports their decision of an authorship for a handwritten text by highlighting the relevant areas.

Furthermore, the following research questions will be addressed:

- What characteristics are selected by a neural network to identify the author of a handwritten text?
- How do the visualizations of feature contribution differ from text areas, which experts consider when identifying the author of a handwritten text?
- How well does a transparency technique perform on neural networks, which take handwritten text images as input?

1.2 Structure of Thesis

Firstly, Chapter 2 introduces the current state of the art for transparency techniques as well as WI and WV methodologies. The transparency techniques are divided into two parts, where the first part introduces transparency techniques for classification networks, and the second part focuses on transparency techniques for embedding networks. Additionally, in the scope of the state of the art for WI and WV methodologies, common concepts for training and evaluation of such methodologies are defined.

Chapter 3 provides an overview of the selected methodology. First, the selected architectures of the neural networks for WI and WV are introduced. Additionally, the hyperparameters and configurations used for the training of these networks are described. Afterwards, the selected transparency techniques and proposed adjustments for the use with WI and WV neural networks are introduced.

Chapter 4 presents the datasets used for the evaluation. Additionally, adjustments made to the original datasets and the splits into train, validation and test set are described here.

Chapter 5 contains the evaluation results. First, experiments performed to improve the performance of the underlying neural networks are outlined. Then, the evaluation of

the transparency techniques, which includes sanity checks, quantitative and qualitative evaluation, are presented, followed by a discussion of the results.

Finally, Chapter 6 concludes the thesis with a summary and an outline of possible future work building upon the results of this thesis.

CHAPTER 2

State of the Art

This chapter gives an overview of the state of the art for transparency techniques and neural networks for WI and WV. For the area of WV, the state of the art is extended with methodologies proposing an approach to signature verification, where the content of the text contained in the snippets to be compared is identical, which is not necessarily the case for WV [Sch08]. Additionally, commonly used concepts for training and evaluation of WI and WV neural networks and for evaluation of transparency techniques are defined and described. The presented methods and techniques are the basis for the selection of the methodology, as described in Section 3.

2.1 Transparency Techniques

The state of the art for transparency techniques can be divided into two groups based on the neural networks the techniques are applicable to. The first group can be applied to neural networks trained with a classification loss, which provide a class label as output. The second group contains transparency techniques for neural networks trained with an embedding loss, which output a vector embedding. Transparency techniques, which are applicable for classification neural networks, are not necessarily applicable for embedding neural networks and vice-versa [KSDH21]. Zheng et al. [ZKC+20] show that extending a neural network with a classification module to apply transparency techniques for classification networks does not result in comprehensible attention maps. Therefore, techniques explicitly using embedding neural networks and techniques for classification neural networks have been developed.

Zhang et al. [ZTLT21] differentiate transparency techniques by three characteristics: passive vs. active approach, type of explanation, and local vs global interpretability. *Passive* approaches provide explanations for already trained networks, while *Active* approaches influence the ongoing training process and network to increase its transparency. The

authors define four types of explanations. *Examples* explanations return similar samples, while *Attribution* explanations provide information on the importance of input features. Additionally, *Hidden Semantics* explanations provide information on one or many hidden neurons or layers of the underlying network and *Rules* explanations collect rules with which the decision for a given input is made. *Local* and *global* interpretability address the input space. For *local* approaches, the explanation provided addresses single samples, while for *global* approaches, an explanation for the whole underlying network is provided [ZTLT21].

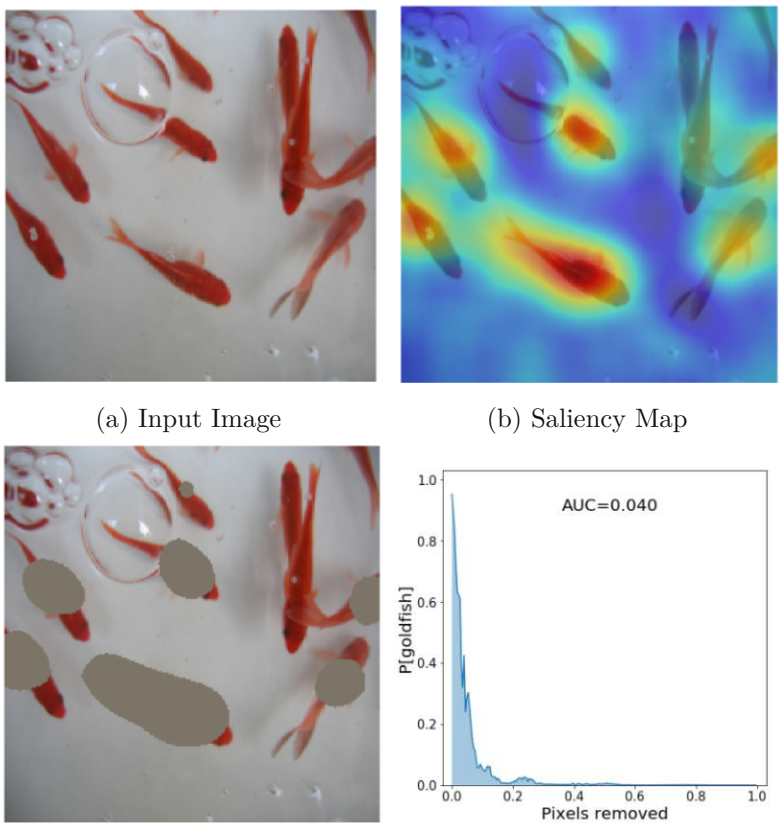
For the application of neural network explainability in the scope of this thesis, the focus will be on passive approaches providing explanations based on the input features.

2.1.1 Concepts for Evaluation

This section defines two concepts proposed for the evaluation of transparency techniques and the accuracy of their created visualizations.

The '*deletion score*' is an evaluation metric for saliency maps generated by a transparency technique. The original metric is proposed by Petsiuk et al. [PDS18] and calculates the change in probability for an output of a classification network if the input image is altered. The pixels in the image are gradually deleted from a given image, starting with the pixels with the highest significance to the neural network output, i.e., the pixels with the highest values in the saliency map. Petsiuk et al. [PDS18] discuss multiple deletion strategies, such as setting the pixels to a constant value or blurring of the pixels. After each step, the change in the probability output is measured and plotted as a curve. An exemplary curve is shown in Figure 2.1. Here, the curve is plotted for an input image classified as "goldfish". The area-under-the-curve (AUC) is then used as the evaluation metric, where a lower value is desirable. Hu et al. [HVH22] propose an adjustment to this metric in order to use it for neural networks providing an embedding vector as output. Instead of measuring a change in the probability output, the change in similarity between a query image and the adapted retrieved image is calculated. Afterwards, the AUC for the change in the similarity score is calculated as measurement [HVH22].

The '*insertion score*' is an evaluation metric, which is similarly calculated to the deletion score and is proposed by Petsiuk et al. [PDS18] as well. It is calculated by gradually inserting pixels into the given image, starting with the pixels with the highest values in the saliency map. Petsiuk et al. [PDS18] propose to use a blurred image initially and adding the original pixel values in each step. Similar to the deletion score, Hu et al. [HVH22] propose to calculate the similarity between the embedding output of the query image and the adapted retrieved image for embedding neural networks. The AUC is calculated as well, with a higher AUC value expected for the insertion score [HVH22].



(a) Input Image (b) Saliency Map

(c) Deletion Step (d) Deletion curve

Figure 2.1: Example for a deletion curve calculated for the probability "goldfish". Images courtesy of Petsiuk et al [PDS18].

2.1.2 Transparency Techniques for Classification Networks

The transparency techniques for classification networks are divided into three groups: local, semi-local and global approaches. Local approaches visualize the features selected by a network for the decision process of one specific input, therefore providing precise information for this individual case. However, this information does not provide insight into the overall logic of the network. In contrast, global approaches generalize and provide information on the general patterns learned by the network. These approaches can provide information on the overall decision process of the network and support the detection of biases within the training data [DVK17, GMR+18, MLB+17, ZTLT21]. Semi-local approaches are located on the spectrum between global and local approaches. For example, an approach providing information on a group of (similar) inputs is considered semi-local [ZTLT21].

Local Approaches

Wang et al. [WZB19] propose to use the bias term for attribution. They suggest a recursive backpropagation algorithm called Bias Backpropagation (BBp), which propagates the bias attribution to the input features, starting on the output layer and computing it layer by layer. Their results show that the bias can contain additional attribution information, which is complementary to information provided by gradient-based methods. Heskes et al. [HSBC20] use causal Shapley values to describe the correlation between an input value and the corresponding output by a network. Using the Shapley values, their approach provides information on important, contributing, and irrelevant features of the input regarding the resulting output of the network. Their proposed algorithm uses causal chain graphs to compute the values.

Global Approaches

Lapuschkin et al. [LWB+19] provide their semiautomatic framework SpRAy (spectral relevance analysis) for the interpretability of neural networks. It provides information on the decision process of a neural network using a whole dataset as input. The individual relevance maps for each input sample are calculated using layer-wise relevance propagation. Afterwards, spectral cluster analysis is applied to the created maps to find structures within the relevance maps representing the according class. Eigengap analysis is then applied to find distinct clusters. The approach by Salman et al. [SPL+20] combines the local attribution of multiple networks using clustering to provide a global interpretation of the networks. They use the Jacobian difference vector to retrieve information about local feature importance. The Jacobian matrices are then clustered using a common clustering algorithm.

Semi-Local Approaches

The approach by Sundararajan et al. [STY17] is called integrated gradients. The methodology uses the gradient operation to calculate attribution in the input. It considers the straight-line path between the baseline and a given input and calculates the gradients along the path. This approach fulfils the axioms of sensitivity and implementation invariance. The semi-local interpretability of the methodology is given as a reference point for the calculation and can be selected by the user. Ramamurthy et al. [NRVZD20] propose a methodology which provides a multi-level explanation using any local explainability technique. The methodology generates an explanation tree where leaves represent local explanations and the root provides an overall and global explanation. The levels in between are generated in a bottom-up fashion where explanations from multiple samples are grouped.

2.1.3 Transparency Techniques for Embedding Networks

In this section, transparency techniques, which can be applied to neural networks providing an embedding for an input image, are described. In contrast to the techniques

described in Section 2.1.2, these techniques do not require a classification module for the neural network to generate the visualization.

Zheng et al. [ZKC+20] propose a transparency technique for neural networks calculating similarities and differences between images using embeddings. The approach uses the output vector of the neural network and determines which feature dimensions are significant for the determination of similarity. The authors explain the functionality of their approach using three different architecture types, namely Triplet, Siamese and Quadruplet architectures. In general, their approach aims to provide insight on the similarities between two images. Using the feature vectors of a tuple of images, the authors calculate weight vectors for each image and generate a single weight vector displaying which feature dimensions contribute the most to the similarity and dissimilarity between these images. Using this weight vector, the attention maps are obtained by calculating sample scores first and using the gradient with the convolutional feature maps of the image to generate an attention map. For their experiments, the authors use a ResNet50 as neural network architecture. The authors experiment with person re-identification, low-shot semantic segmentation and image retrieval. For all three topics, the authors provide qualitative and quantitative evaluations. For the qualitative evaluation, they objectively evaluate if the created attention maps highlight the correct regions of the image and cover the most important features of the according object. For the quantitative evaluation of the person re-identification and image retrieval tasks, the authors use the Recall@K metric. For the low-shot semantic segmentation task, the 1-way and 2-way meanIOU evaluation are used [ZKC+20].

Chen et al. [CCHM20] introduce an adaptation of the Grad-CAM [SCD+17] method, which is constructed for embedding networks, while the original Grad-CAM method is applied to classification networks only. During the training of the transparency technique, a grad-weights dataset is created from the training images. These weights are then transferred during the testing phase to visualize the saliency maps for the test images using forward propagation only. The grad-weights are calculated as described for the Grad-CAM method. The authors use the Triplet Loss as differentiable activation for the grad-weights generation. The grad-weights are averaged over multiple sampled triplets to create a more accurate grad-weights dataset. In order to improve the results, the authors propose to use the top weights by sorting them per channel and utilize the top-M weights of this ranking. The grad-weights from the nearest training image by embedding are selected during testing. For the selection, nearest neighbour search is used, therefore, no back-propagation for the testing image is needed. The authors use the CUB200-2011 dataset with the bounding box score and mask score metrics for evaluation. For the experiments, the Grad-CAM and Grad-CAM++ methods are used as a basis for the calculation. The authors additionally provide a qualitative evaluation by manually comparing the saliency maps generated by models with different settings, such as the number of triplets for the generation of the grad-weights and the change in base methods [CCHM20]. Kuehlkamp et al. [KBC+22] note that this method is more appropriate for

closed-set cases, as the grad-weights for the testing phase are inferred from the training data. Therefore, this method explains why the two training images, which are spatially closest to the test image, are similar instead of comparing two test images with each other [ZKC+20].

Kobs et al. [KSDH21] investigate the differences in visual features selected from the input by models with the same architecture but different loss functions. For this purpose, the authors propose two analysis methods, a comparison of gradient-based saliency maps and a comparison of images on image property-level. 14 different loss functions and three different datasets, namely Cars196, CUB200 and Stanford Online Products (SOP), are used for the experimentation. The first analysis method creates a saliency map by highlighting pixels in the input image based on their contribution to the embedding output of the model. For the calculation of the maps, the authors use the embedding for a base image, i.e. an image consisting of black pixels only. The goal is to find the pixels influencing the base embedding to reduce the distance between the base embedding and an image embedding. The gradients of the loss-specific distance are computed and the Smooth-Grad method is applied to reduce noise. The higher the resulting gradient for a pixel, the more impact a pixel value change has on the embedding vector. The authors propose to compare the saliency maps of the same input image generated by models using different loss functions. For this purpose, the saliency maps are compared using the average Pearson Product-Moment Correlation Coefficient and Jensen-Shannon Divergence. The second analysis method investigates the impact of image properties such as colour and rotation of objects on the resulting embedding. A change in a property, which highly influences the embedding output of the model, should lead to large changes in the embedding output. Additionally, when changing all other properties within the same image while keeping the highly influential property unchanged should lead to small changes in the embedding output. For this purpose, clustering behaviours for image properties are analysed with the R-Precision metric. The authors provide qualitative and quantitative evaluations for the saliency maps. They use the SOP, Cars196, CUB200 and a synthetically generated dataset for evaluation. For the qualitative evaluation, the authors manually compare the highlights of the saliency maps retrieved from models with different loss functions. For the qualitative evaluation, the cross-correlation values are compared [KSDH21]. Kobs and Hotho [KH22] note that the investigation of differences in visual features requires controlled generated datasets in order to access a dataset with small feature changes between images, which can be a time-consuming task.

Chen et al. [CLL+21] propose the method Attribute-guided Metric Distillation for the explanation of person re-identification models. The goal of the method is to generate attention maps which highlight the most important attributes for the distinction between different persons. The transparency method itself learns to interpret the distance between the images of two persons using semantic attributes of the images and providing information on the impact of each attribute on the distance. The technique is trained after the target neural network, on which it should be applied, has been trained. It is structured

such that it has the same architecture as the target network. Both structures share the first Convolutional Neural Network (CNN) stages. The later layers of the transparency technique are trained using information from the target network. For an image pair, the feature maps are extracted from the last convolutional layer of the neural network. Using this output, the feature vectors are calculated by applying Generalized Mean Pooling. By attaching an Attribute Decomposition Head with multiple convolutional layers to the transparency technique, attribute-guided attention maps are generated, which are sliced into matrices, where each matrix represents the attention for one of the previously determined attributes of the images. The authors evaluate their method on the Market-1501 and the DukeMTMC-ReID datasets. The datasets are evaluated on the metric for Distillation and the metric for Attribute Decomposition. ResNet34, ResNet50 and ResNet101 architectures are used as architecture backbones. First, the models are evaluated individually for each dataset. Afterwards, the performances on all datasets are compared [CLL+21].

Zhu et al. [ZYC21] propose a point-to-point activation methodology, which allows the exploration of the relationship between different images and their similarities. The overall saliency maps are calculated by decomposition along the images. Additionally, each pixel in one image can be compared to all pixels in the other image by decomposition of a pixel along the other image, providing information on the similarity between the according pixel and the other image. The authors build their methodology on the concept of Class Activation Map (CAM). However, instead of decomposing along one image, the authors propose to use the feature maps of two images. The point-specific saliency maps can then be calculated by adding up the multiplication of the feature maps for two points (x, y) and (i, j). The authors use a weakly supervised localization method to evaluate their proposed methodology. The saliency maps for four methods (Grad-CAM, Grad-CAM (no norm), Decomposition + Bias and Decomposition) are compared by generating a mask using a threshold for the heatmap. Then, a bounding box is created from the largest connected component of the mask. The comparison of the boxes is done by Intersection over Union (IoU). Additionally, the authors apply a qualitative evaluation using the Amazon Mechanical Turk platform. Reviewers have to compare pairs of images consisting of one saliency map from the decomposition and one saliency map from the Grad-CAM/Grad-CAM (no norm) methods. The reviewers are asked to select the map from each pair that they believe to be more reasonable [ZYC21].

Zhang et al. [ZZZL22] propose an Attributable Visual Similarity Learning framework to explain the similarities between two images. The framework provides a graph structure, which displays coarse similarities on the top nodes, which are divided into fine details on the lower levels, mirroring the comparison process executed by humans. The undirected graph is extracted from a feature extractor such as a neural network, where embeddings are extracted from each layer of the neural network. The authors note that high-level layers contain information about high-level features, which are complementary to the low-level features of lower layers. The nodes of the graph are called similarity nodes.

The feature maps of each convolutional block are extracted and a global pooling and a Fully Connected (FC) layer are applied to receive a corresponding embedding. The similarity node is calculated as the square difference between normalized embeddings of two input images. The edges between two similarity nodes are generated using CAMs. The CAMs are then used to compute the correlation using the inner product. The authors evaluate their proposed method using the CUB-200-2011, Cars196 and SOP datasets. The recall@Ks metric is used as evaluation metric. Additionally, the authors visualize the graph results for a number of samples from the datasets. The ResNet50 is used as neural network. The framework is evaluated with the Margin Loss and the ProxyAnchor Loss [ZZZL22].

2.2 Writer Identification and Writer Verification Methodologies

In the scope of this thesis, transparency techniques are applied to neural networks trained on WI and WV. The state of the art for WI and WV is described in this section. Additionally, common concepts for the training and evaluation of these networks are described here.

2.2.1 Concepts for Training and Evaluation

This section defines notions and concepts which are frequently used for the training and evaluation of neural networks and are mentioned throughout this work.

Concept for Training

The '*x-fold cross-validation*' is a training method for neural networks. The available dataset is divided into x distinct subpartitions. This allows to train x models, where a model i is evaluated on the i^{th} subpartition and trained with the other combined subpartitions [SE10].

Concepts for Evaluation

The '*leave-one-out*' strategy is an evaluation method for WI neural networks. Each image taken from a validation or test set Q is used once as a query image q. For each query image q, the other images are ranked based on the pairwise distance between q and the according image to rank. The distance is calculated using a similarity criterion, such as Euclidean distance or Cosine Similarity. The resulting ranking places images with the smallest distance at the top and images with the highest distance at the bottom. The goal is to have relevant images, i.e. images from the same author as q, at the top of the ranking [LJ19, WTB14].

The '*Mean Average Precision (mAP)*' metric is an evaluation metric for machine learning

models. It provides information on the accuracy a neural network has achieved on a dataset and is calculated as [RB22]

$$mAP = \frac{\sum_{q \in Q} AveP(q)}{|Q|}. \tag{2.1}$$

$AveP$ is the average precision calculated as [RB22]

$$AveP(q) = \frac{\sum_{k=1}^{n} (P(k) \times relevance(k))}{x}, \tag{2.2}$$

where x is the number of relevant documents, i.e. the number of documents written by the same author as q. $P(k)$ is the percentage of relevant documents in the first k documents of the ranking result. $relevance(k)$ is a function defined as [RB22]

$$relevance(k) = \begin{cases} 1, & \text{if relevant document} \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

The mAP value ranges from zero to one, with one indicating all relevant documents are positioned at the top of the ranking for all query images q [KFS18, SMR08].

The '*Top-x*' metric is used for the evaluation of neural networks as well. As with the mAP metric, a ranking of images is generated for each image in a validation or test set. The top x images in the ranking are then inspected regarding the authorship. For the soft criterion, i.e. '*Soft-Top-x*', at least one of the first x images in the ranking must be from the same author as the query image for the ranking to be considered correct. For the hard criterion, i.e. *Hard-Top-x*, all first x images in the ranking must be from the same author. The *Soft-* and *Hard-Top-x* metrics are then the percentage of correct rankings of all query images. For $x = 1$, the hard and soft criterion are equivalent and therefore noted as Top-1 [CGFM17].

The '*False Acceptance Rate (FAR)*' and '*False Rejection Rate (FRR)*' are two evaluation metrics used for verification systems such as WV. The metrics are calculated as the percentage of invalid objects which are accepted and the percentage of valid objects which are rejected by the underlying system, respectively [QZZ+23]. In the case of WV, an invalid object is a pair of images written by different authors, while a valid object is a pair of images written by the same author. Therefore, the '*FAR*' is the percentage of forgery pairs which are determined as genuine pairs and the '*FRR*' is the percentage of genuine pairs which are determined as forged pairs.

The '*Equal Error Rate (EER)*' is an evaluation metric used for systems providing a verification. It is defined as the error rate at the threshold of the similarity metric where the False Positive Rate and True Positive Rate are equal. A lower value for the '*EER*' is desirable [LZL+19, SSGS00].

2.2.2 Writer Identification

Christlein et al. [CBMA15] propose the use of CNNs to learn local activation features, which are then merged to a global feature descriptor for one image using Gaussian Mixture Model (GMM) supervector encoding. Afterwards, the descriptor is normalized. The CNN is used to generate a local feature descriptor by cutting of the last layer after training and using the output of the penultimate layer as descriptor. The last layer consists of 100 Softmax nodes, where each represents the ID of one of the writers in the used train set. The CNN itself contains two blocks, where each block contains a convolutional layer followed by a pooling layer. The output of the second block is then forwarded to a hidden layer, which transforms the output into a 1-dimensional vector. The Rectified Linear Unit (ReLU) is used as activation function for all layers. The CNN takes 32px $\times$ 32px image patches as input. For the aggregation of the local feature descriptors to global descriptors, a GMM is used. This descriptor is then normalized using a kernel derived from the Kullback-Leibler divergence. The authors use the ICDAR2013 and CVL datasets for training and evaluation. For the evaluation, the mAP and the hard Top-X metrics are used [CBMA15].

Fiel and Sablatnig [FS15] propose the use of a CNN for the creation of a feature vector for a handwritten text image. The vector is used as distance measurement to determine the similarity of different handwritten images. In order to retrieve the feature vector, the output of the penultimate FC layer is used. Before the images are forwarded to the CNN, the handwritten text images are binarized with the method of Otsu followed by applying a text line segmentation and a sliding window. The sliding window is used to create fixed-size image patches as input and is applied with a step size of 20 pixels. The architecture of the CNN is based on the caffenet. It contains five convolutional layers and three FC layers. After each convolutional layer, a max pooling layer is inserted. The network is trained using the Softmax Loss function. The final classification layer, which consists of 1000 neurons, is cut off. During training, the resulting feature vectors of all image patches of one handwritten image are averaged to receive a representative feature vector for the image. The feature vectors are compared with the χ_2-distance. The authors use the ICDAR2011, ICDAR2013 and CVL datasets for evaluation. The ranking of the documents for each query image is evaluated using the soft and hard Top-X metrics [FS15].

Xing and Qiao [XQ16] propose a DNN called DeepWriter. It consists of two parallel streams, which are called Half DeepWriter and which share weights between their layers. The architecture of the Half DeepWriter is based on the AlexNet and consists of convolutional, max-pooling, and fully convolutional layers. The output of the two streams after the last fully convolutional layers are summed up element-wise. Finally, a softmax layer is applied to the given result. ReLU is used as activation function and the Softmax Loss is used as loss function. The network takes 113px $\times$ 113px images as input. In order to account for non-quadratic images, the authors propose a patch-scanning strategy. The input image is resized such that the smaller side is 113px wide. Then, image patches with a size of 113px $\times$ 113px are sampled from the resulting image. In order to preserve

the relation between adjacent image patches, the architecture takes two adjacent image patches as input and forwards each to one stream within the DeepWriter. For training of the network, the authors pretrain the Half DeepWriter using the HWDB1.1 dataset. After 400.000 iterations, the DeepWriter architecture is trained on the IAM dataset using the pretrained weights. During training, the authors use the average from the softmax layer output for all image patches taken from one input image to define the final classification. In their experiments, the authors show that the use of adjacent image pairs increases accuracy [XQ16].

Tang et al. [TW16] propose the use of a CNN for feature extraction and a Joint Bayesian for WI. The approach takes whole handwritten images as input and extracts global features instead of local features with the CNN. The authors propose a data augmentation technique to generate multiple images from one handwritten image. These images are then forwarded to the CNN, which extracts a 256-dimensional feature vector for each image. Finally, the Joint Bayesian is used for WI. The authors consider the handwritten text images simpler in comparison to natural scene images and therefore propose a lightweight CNN architecture. The CNN consists of four convolutional layers, two FC layers and a softmax layer as last layer. ReLU is used as activation function. Each convolutional layer is followed by a local response normalization and a max pooling layer. Additionally, one dropout layer is inserted after each FC layer to avoid overfitting. The authors evaluate their approach on the ICDAR2013 and the CVL datasets. For the evaluation, the soft Top-1, soft Top-5, soft Top-10, hard Top-2, and hard Top-3 are used. Additionally, the hard Top-4 is used for evaluation of the CVL dataset [TW16].

Christlein et al. [CGFM17] propose to train a deep CNN in an unsupervised manner for WI and writer retrieval. The train dataset is put into surrogate classes by clustering Scale-invariant Feature Transform (SIFT) descriptors. In a first step, SIFT keypoints are computed for the train dataset. For each keypoint, a descriptor and an image snippet with a size of 32px $\times$ 32px centred around the keypoint are calculated. The authors note that keypoints, which are lying between text lines, are filtered out by using a restricted SIFT method, which keeps only minima in the scale space. The remaining descriptors are normalized and reduced from 128 to 32 dimensions using Principal Component Analysis (PCA). The descriptors are then clustered and the clusters are used as targets for the training of the CNN. The output of the penultimate layer is used as the feature descriptor. Finally, a global image descriptor is created using Vector of Locally Aggregated Descriptors (VLAD) encoding. The authors evaluate their proposed method on the Historical-WI and CLaMM16 datasets. The method is evaluated using the leave-one-out strategy with the Top-1, soft Top-5, soft Top-10, hard Top-2, hard Top-3, hard Top-4, Precision at N and mAP as evaluation metrics. The authors first evaluate the choice of clusters in contrast to writers as surrogate classes and report a decrease in performance when using writers. Further, four different encoding methods are compared with each other: Sum-Pooling, Fisher vectors, GMM supervectors and the proposed VLAD encoding. The authors report the highest performance for the VLAD

encoding. An evaluation of the parameters revealed a peak in the mAP score when using 5000 clusters for training [CGFM17].

Keglevic et al. [KFS18] propose a triplet CNN architecture. This architecture learns a similarity measurement for triplets of image patches. The triplet contains two image patches from the same author, so-called positive samples, and one image patch from a different author, a so-called negative sample. The output of the architecture is used to generate a VLAD, which is then used for the identification of authors of further image patches. The goal is to minimize the distance between the positive samples and to maximize it between the negative samples. The authors propose to extract the image patches from handwritten text images using SIFT keypoints. First, the images are binarized. Then, SIFT features are calculated and used as centres of the 32px × 32px image patches. The features are filtered using clustering with surrogate classes. However, this step is omitted during evaluation. The triplet CNN architecture consists of three DenseNet CNN branches, which share weights between their layers. The CNNs consist of convolutional layers with ReLU as activation function and batch normalization. They learn an embedding of the image patches using the L2 distance. The embeddings of all image patches for one document image form its feature vector by using VLAD encoding. The triplet CNN is evaluated on the ICDAR2013 dataset. For the evaluation, the authors applied the leave-one-out strategy. The mAP is used for the evaluation of the rankings. The authors evaluate their approach using four different strategies for VLAD. Additionally, the number of clusters is varied. The best performance is achieved using five VLADs with 100 cluster centres using the Euclidean distance. The use of multiple vocabularies instead of one leads to a better performance in all experiments [KFS18].

Nguyen et al. [NNI+19] propose an end-to-end method using multiple parallel CNNs for feature extraction. Their approach uses multiple square images randomly sampled from an image containing handwriting from one writer, which are forwarded in parallel to the CNNs. The resulting writer-specific features from the CNNs are then aggregated by a global feature aggregator. The output of the aggregator is considered as the global feature for the given writer. This output is then forwarded to a softmax classifier, which provides a writer prediction. The whole network is trained using Stochastic Gradient Descent (SGD). The authors propose two architectures for the CNNs. The first architecture extracts local features at a sub-region level. It consists of four stages, which contain a convolutional layer and max-pooling layer. This architecture returns a $4 \times 4 \times 1024$-dimensional matrix. The second architecture extracts local features at the character level. It consists of three convolutional layer blocks and returns a $1 \times 1 \times 1024$-dimensional matrix. For the aggregation, the authors propose three methods. Average aggregation computes the average of the values along the depth dimension. Max aggregation selects the maximum value along the depth dimension. The average of k-max aggregation computes the average of the k largest values along the depth dimension. The authors use the JEITA-HP, Firemaker and IAM datasets for evaluation [NNI+19].

Javidi and Jampour [JJ20] propose the use of a residual neural network in combination with a Handwriting thickness descriptor (HTD) as auxiliary feature input. The neural network consists of 18 layers with four residual blocks. Additional batch normalization layers and ReLU activation function are used. The handwriting thickness is a characteristic which varies between different people [JJ20]. The descriptor is calculated by counting fully black rectangular patches within an input image using convolutional processing. The HTD is conjugated with the output from the last flattening layer of the residual network. The network takes 60px × 180px images as input. In order to collect image patches from a larger input, the authors use a scanning strategy similar to Xing and Qiao [XQ16]. Additionally, the constraint of at least 10% black pixels within the image patch is given to ensure a certain amount of information within the image patch. For the evaluation of the network, the authors use the IAM, Firemaker, CVL and CERUG-EN datasets. The authors compare the performance of their approach with and without the use of auxiliary features [JJ20].

Wang et al. [WMC21] propose an end-to-end framework for WI. The framework consists of a preprocessing step and an encoding step consisting of a local feature extraction and a global descriptor computation. The preprocessing step uses a U-Net for binarization of the input images. The local descriptors are created using a ResNet50 architecture, which are forwarded to an optimized encoding layer to compute global descriptors. The architecture is shown in Figure 2.2. The authors use the DIBCO dataset to train the

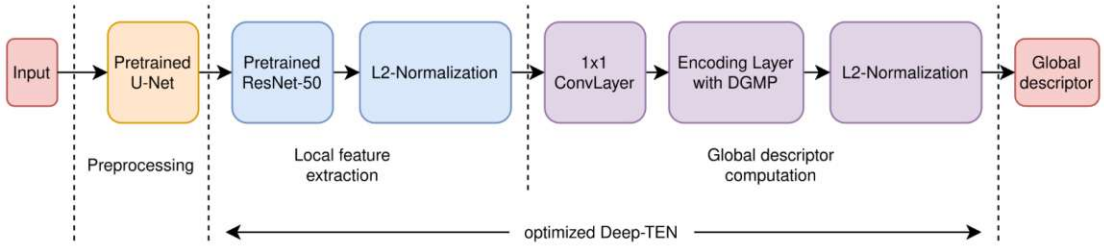


Figure 2.2: Deep-TEN architecture as proposed by Wang et al [WMC21]. Image courtesy of Wang et al [WMC21].

U-Net for binarization. For the creation of local descriptors and aggregation into one global descriptor, the Deep-TEN architecture as proposed by Zhang et al. [ZXD17] with additional Deep Generalized Max Pooling (DGMP) [CSS+19] is used. The authors evaluate their framework with the Historical-WI dataset. The input images are divided into 400px × 400px image patches. A canny edge detector with a threshold of 2000 is used to determine the amount of handwritten text in each image patch. The framework is trained using the Triplet Loss. For the evaluation, the Top-1 and mAP metrics are used. The authors evaluate the performance of each individual step in the framework and experiment with different settings. For the U-Net, the best performance is achieved when pretraining the U-Net without fine-tuning. For the Deep-TEN architecture, the best result is achieved when not using DGMP [WMC21].

Rasoulzadeh and BabaAli [RB22] propose the use of a CNN together with a NetVLAD layer, which is based on the ResNet20 architecture. The network is used to extract local descriptors for image patches. The descriptors are then combined to global descriptors for each image using Generalized Max-Pooling (GMP). PCA is applied before the descriptors are compared with each other and a ranking is constructed. The Triplet Loss is used for training. The architecture is shown in Figure 2.3. The authors use 32px × 32px image patches as input. The patches are constructed using the handwriting contour as centre. The last FC layer of the neural network is discarded and the $1 \times 1 \times 64$-dimensional feature vector is forwarded to a Global Average Pooling (GAP) layer, which is afterwards forwarded to the NetVLAD layer learning the VLAD embeddings. The authors evaluate their approach on the ICDAR2013, the CVL and the KHATT datasets. For the evaluation, they use the Top-1, hard Top-2, hard Top-3 and mAP metrics. For the optimization, Adamax with the Triplet Semi-Hard Loss is used [RB22].

2.2.3 Writer Verification

Shaikh et al. [SDCS20] propose the use of an attention-based methodology which uses cross- and soft-attention. The stem of the Inception-ResNet-v2 or VGG16 network is used for feature extraction in a Siamese architecture, i.e. two networks with shared weights and parameters are used. The output of these networks $f_1, f_2 \in \mathbb{R}^{h \times w \times d}$ is forwarded to the cross-attention module. Here, the calculations are done once with f_1 as key input and f_2 as query input as well as once with f_2 as key input and f_1 as query input. The resulting outputs are concatenated along the channel axis and are forwarded to the soft-attention module. The goal of this module is to identify features, which are significant for the classification output. The soft-attention module uses 3D kernels for convolution, which generates the output $g \in \mathbb{R}^{h' \times w' \times k}$, where k is the number of kernels [SDCS20]. The authors use categorical Cross-Entropy Loss as well as Focal Loss as loss function [SDCS20]. For the evaluation, the authors use segments from the CEDAR dataset, which contain the word "and", as well as the whole dataset. The proposed methodology is evaluated using F-1, Precision, Recall, FAR, FRR and accuracy as evaluation metrics [SDCS20].

Dey et al. [DDIT+17] propose the use of a convolutional Siamese network called SigNet for off-line signature verification. The Siamese network uses two identical CNN architectures as backbone, which are trained to place the given input images containing a signature in an embedding space, where signatures of the same author are placed close to each other while signatures written by different authors are positioned further away from each other. The CNN architecture used is taken from Krizhevsky et al [KSH12]. The two CNNs share their parameters and weights and are joined by a Contrastive Loss function, which uses the Euclidean distance. ReLU is used as activation function [DDIT+17]. The overall architecture is shown in Figure 2.4. Using the Contrastive Loss, a threshold must be determined to distinguish between genuine and forged signature pairs. The authors determine the threshold by calculation of the maximum accuracy over all possible threshold values using the true positive and true negative rates [DDIT+17]. The authors

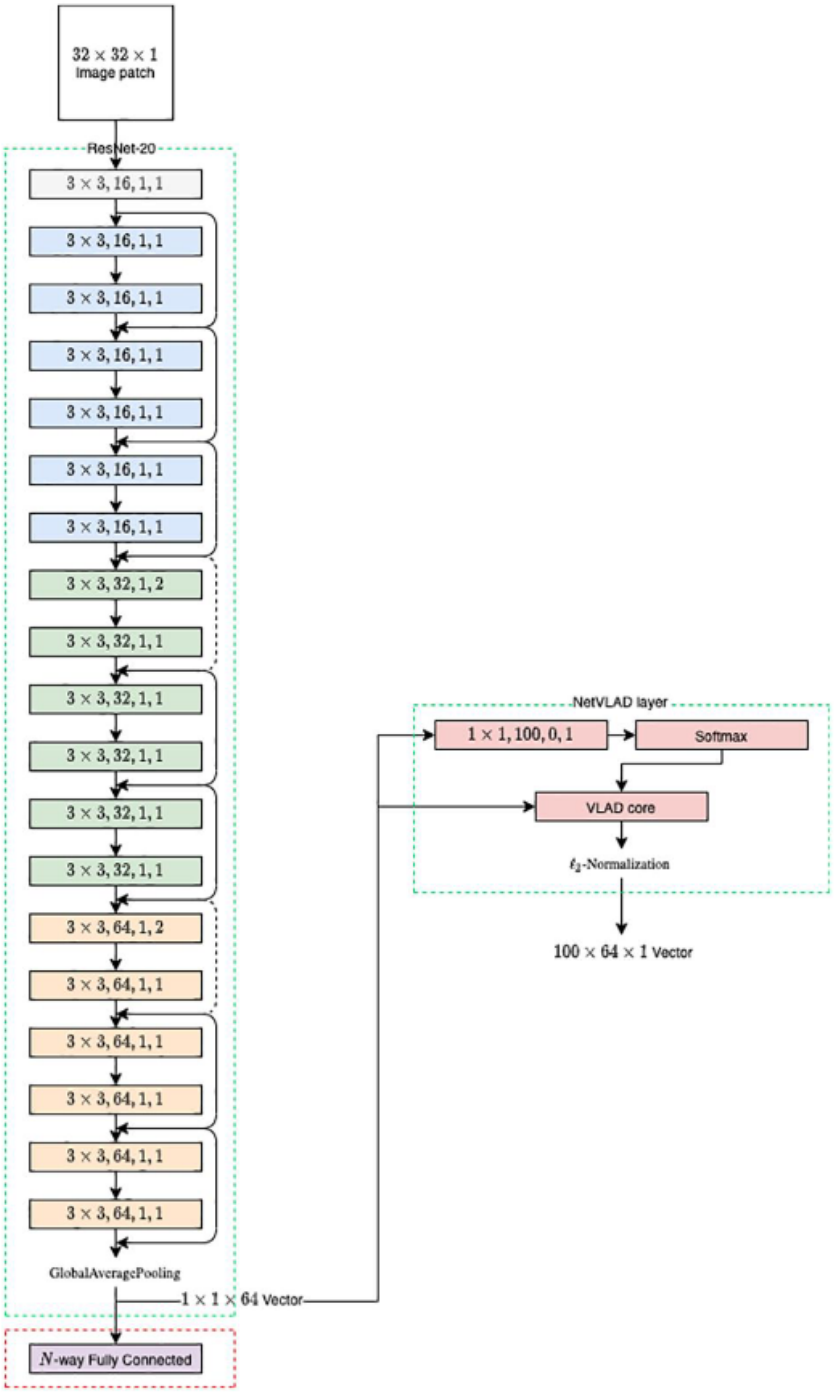


Figure 2.3: ResNet20 architecture as proposed by Rasoulzadeh and BabaAli [RB22]. Image courtesy of Rasoulzadeh and BabaAli [RB22].

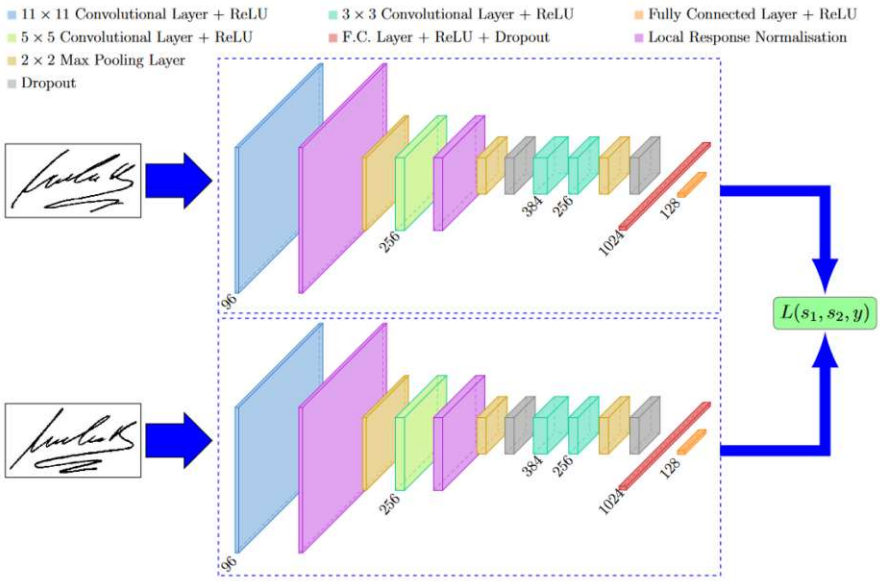


Figure 2.4: Architecture of SigNet as proposed by Dey et al [DDIT+17]. Image courtesy of Dey et al [DDIT+17].

use the CEDAR, GPDS300, GPDS Synthetic Signature and BHSig260 signature datasets and split them into open train and test sets using different split sizes. The created sets are then adjusted to create the same amount of genuine and forged signature pairs for all datasets. Additionally, the authors conduct a cross-dataset evaluation, where the SigNet is trained on one dataset and the test set of another dataset is used for testing [DDIT+17].

Li et al. [LZL+19] propose the use of two bidirectional Recurrent Neural Networks (RNNs) for signature verification of online acquired signatures. The first RNN, the Stroke RNN, is used for feature extraction of the stroke information provided in a signature pair, while the second RNN, which is called Signature RNN, uses the output from the Stroke RNN to extract global features for the given signature pair. Both RNNs consist of two LSTM layers. The output of the Stroke RNN consists of three digits, which represent the probability of the signature pair being a skilled forgery, a random forgery and a genuine pair, respectively. The overall dissimilarity between the signature pair is then calculated as $D = p_{SF} + p_{RF} - p_G$, where p_{SF} is the probability of the pair being a skilled forgery, p_{RF} is the probability of a random forgery and p_G is the probability of a genuine signature pair [LZL+19]. The architecture is shown in Figure 2.5. The authors use the BiosecureID, MCYT-100, SCUT-MMSIG and MOBISIG datasets, which contain online signatures, for their evaluation. For preprocessing, the data is normalized using z-score. Additionally, the signature data is divided into individual pieces at points where the pen was lifted up, i.e. the pen pressure is 0. For the evaluation, three types of signature pairs are created. The first pair contains signatures written by the same person, the second pair contains one true and one forged signature and the third pair contains two

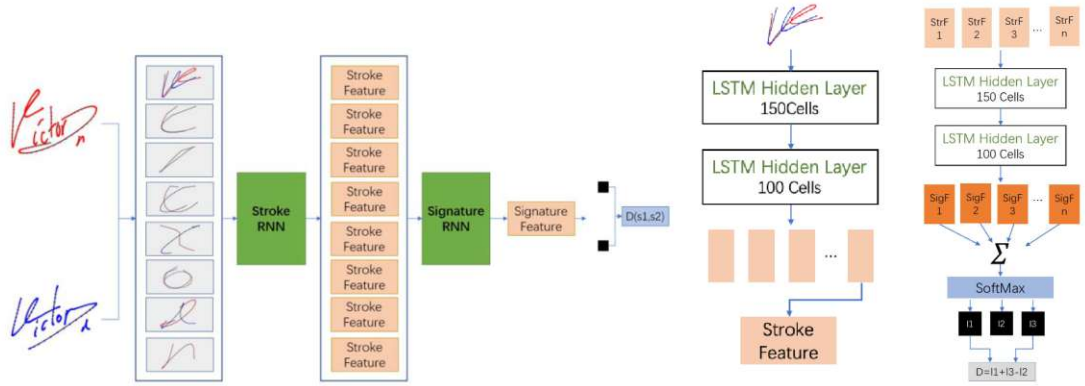


Figure 2.5: Architecture of the methodology as proposed by Li et al [LZL+19]. Image courtesy of Li et al [LZL+19].

genuine signatures taken from two randomly selected authors, which represents the case of a random forgery. The EER is used as evaluation metric [LZL+19].

Cairang et al. [CZY+22] propose a methodology for signature verification. Their framework consist of a Siamese network with Triplet Loss and Cross-Entropy Loss for feature extraction and an Interference Layer Normalization Neck (ILNNeck) for an improved generalization of the framework on different signatures [CZY+22]. The overall architecture is shown in Figure 2.6. The feature extraction of the framework uses a CNN

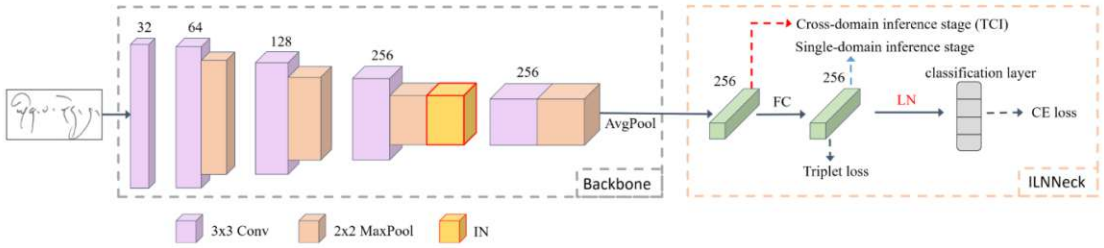


Figure 2.6: Architecture of the methodology as proposed by Cairang et al [CZY+22]. Image courtesy of Cairang et al [CZY+22].

backbone with Instance Normalization (IN), while the ILNNeck uses a combination of Cross-Entropy and Triplet Loss. IN is used as it is suited for cases where the input contains fine-grained information, possibly on a pixel level, and improves the performance between different datasets [CZY+22]. The structure of the ILNNeck is based on the Batch Normalization Neck (BNNeck). However, the authors note that batch normalization removes details in individual samples as it applies normalization over all samples in the given batch. In contrast, Layer normalization (LN) is used to apply sample-wise normalization. Finally, the loss is calculated as the sum of the Cross-Entropy and Triplet Loss, where the Cross-Entropy Loss is scaled with a constant [CZY+22]. The authors

evaluate their method on the CEDAR, BHSig-H and BHSig-B datasets as well as their own dataset named MLSig, which contains signatures written in Chinese, English and Tibetan. The performance is evaluated using single-dataset and cross-dataset evaluation. The AUC and EER are used as evaluation metrics [CZY+22].

CHAPTER 3

Methodology

This section presents the proposed methodology for the use of transparency techniques on WI and WV neural networks. First, in Section 3.1, the neural networks to which the transparency techniques are applied and evaluated on are defined. Two types of neural networks have been selected. The first type is used for WI, while the second type is used for WV. Additionally, the training procedure and configuration for the neural networks are outlined. Then, the transparency techniques selected from the state of the art and their use in the scope of this thesis are presented in Section 3.2. Finally, adjustments made to the metrics used for the evaluation of the transparency techniques are described in Section 3.3.

3.1 Neural Networks

In the scope of this thesis, neural networks are trained on the task of WI and WV. The trained networks are used for the application of the transparency techniques and are utilized to evaluate the functionality of the techniques. The networks used for WI and WV are embedding networks. Therefore, the networks take an image as input and provide an embedding vector as output. The architecture of the networks is selected based on methodologies proposed in the state of the art as presented in Section 2.2. For the area of WV, the state of the art is extended with methodologies for signature verification, where the content of the text contained in the snippets to be compared is identical, which is not necessarily the case for WV [Sch08].

3.1.1 Writer Identification Network

Three neural networks, namely ResNet50, ResNet20 and ResNet18, have been selected for the experiments of this thesis. The selection of the ResNet50 is based on the Deep-TEN architecture proposed by Wang et al. [WMC21], which uses a ResNet50 as feature

Model	Num. Parameters
ResNet18	11.4
ResNet20	0.4
ResNet34	21.5
ResNet50	23.9
ResNet101	42.8
ResNet152	58.5

Table 3.1: Training parameters for the ResNet architectures in millions. The number of parameters for all ResNets except the ResNet20 are provided by Leong et al [LPLL20].

extractor. The ResNet20 is based on the architecture proposed by Rasoulzadeh and BabaAli [RB22], where it is used for feature extraction and generation of an image descriptor. These two methodologies are applied to the raw input images without the need for additional information such as auxiliary features.

ResNet50 The ResNet50 network is used by the Deep-TEN architecture proposed by Wang et al. [WMC21], where the network is used for feature extraction. Since the feature extraction is of interest for the scope of this thesis as the transparency techniques focus on the selection of features from the input image, the ResNet50 network is used as WI neural network. It provides a 1000-dimensional embedding vector as output.

ResNet18 The ResNet18 model is selected based on the ResNet50 architecture. It is a variant of the general ResNet architecture proposed by He et al. [HZRS16], which has fewer parameters to train than the ResNet50 model and, therefore, finishes training faster than the ResNet50 architecture [LPLL20]. Table 3.1 contains the number of parameters for the ResNet variants. As can be seen, the ResNet18 architecture has the second smallest number of parameters to train, with only the ResNet20 network containing fewer parameters. Therefore, the ResNet18 model has been selected in addition to the ResNet50 model for comparison. It provides a 1000-dimensional embedding vector as output.

ResNet20 The methodology proposed by Rasoulzadeh and BabaAli [RB22] uses a ResNet20 with a NetVLAD layer for WI. This architecture is complemented with batch normalization layers, which are placed after each convolutional layer as this improves performance. The model provides a 64-dimensional embedding vector as output.

3.1.2 Writer Verification Network

The architecture for the WV neural network is based on the architecture of SigNet proposed by Dey et al. [DDIT+17] for the task of signature verification. The authors suggest the use of a Siamese network, which uses two CNNs with shared parameters and

weights as backbone. The CNNs are joined using a loss function, i.e. the Contrastive Loss. For the scope of this thesis, the CNNs are replaced. Instead, the ResNet18, ResNet20 and ResNet50 networks are individually used as backbone.

3.1.3 Training

The training configurations are adjusted to achieve the best performance for the WI and WV neural networks. These adjustments are described in this section. The networks are trained using the CVL [KFDS13], Firemaker [BSV03] and ICDAR2013 [LGSP13] datasets, which are described in detail in Chapter 4. Further experiments, which are conducted to improve the implementation and training configuration for the WI and WV networks, are described in the Appendix.

Loss function The WI networks are trained using a Triplet Loss with a margin of 0.3 and the Cosine Similarity as distance measurement. The WV networks are trained with a Contrastive Loss using the Cosine Similarity. The Cosine Similarity is utilized as distance measurement since it is also used for the quantitative evaluation of the transparency techniques as described in Section 5.2.2.

Optimizer The WI and WV networks are trained with an Adam optimizer. The WI networks use a learning rate of 0.01. For the WV networks, the learning rate is reduced to 0.001 as this provides more stable results. Additionally, a scheduler as described in the Appendix is used.

The input forwarded to the WI and WV networks has a size of 400px × 400px. The ICDAR2013 dataset contains pages with a height of less than 400px. Therefore, for this dataset, the input size is set to 200px × 200px. Since the authors of the ResNet20 architecture use an input size of 32px × 32px, the size of the GAP layer of the ResNet20 is adjusted to preserve the output dimension, i.e. a 64-dimensional embedding vector. This is done by reducing the kernel size of the average pooling layer from 100 to 50.

The input images are first transformed to grayscale and then binarized using the Threshold of Otsu method [Ots79]. Afterwards, the images are mapped to the range [-1;1]. The WI and WV networks are trained using 4-fold cross-validation as described in Section 2.2.1. The networks are trained using random snippets from the train dataset. The train dataset contains the full pages as elements. Therefore, one random snippet is taken from each page within the dataset in each epoch. Snippets, which have less than 2% black pixels, are discarded and another random snippet is sampled from the same page. This is done to remove snippets which contain insufficient handwriting information for the author identification. The threshold is determined empirically. After each epoch, the performance is evaluated on the validation dataset, which contains full pages as well. Therefore, each page is divided into a grid of snippets, with each snippet having a size of 400px × 400px. The page is padded with white pixels beforehand to avoid the creation of snippets with smaller sizes. For the WI networks, the mAP is calculated, while for the

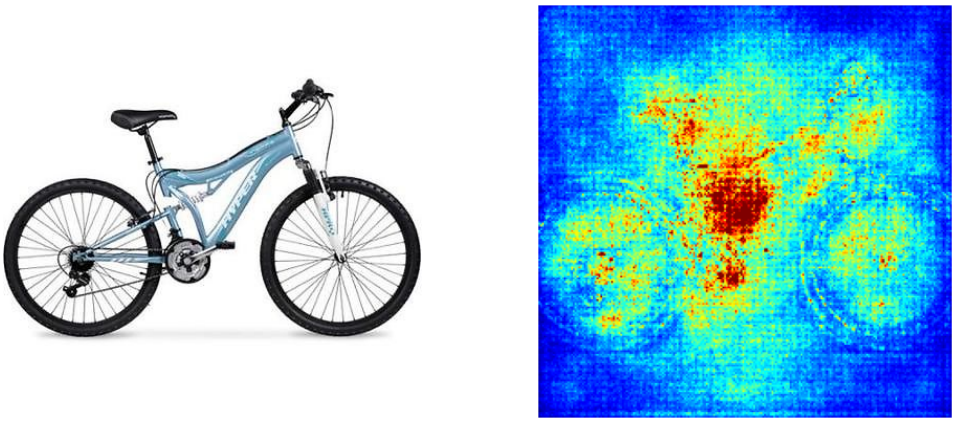
WV networks, the accuracy is calculated. The embedding of a full page is calculated as the mean of the embedding outputs calculated for the according snippets. The mAP is then calculated for all pages for the WI networks. For the WV networks, the fraction of snippets correctly classified as ”same author” and ”different author” is calculated. This is done by creating two pairs of pages for each individual page, where the pages of one pair are written by the same author and the page of the other pair are written by different authors. The embedding outputs of the pages are compared using the Cosine Similarity measurement. For the pair of pages written by the same author, the Cosine Similarity is expected to be greater or equal to 0.5. For the pair of pages written by different authors, the value is expected to be below this threshold. The accuracy is then calculated as the fraction of pairs correctly classified. After the training, the mAP and accuracy for the test sets are calculated as it is done for the validation sets using the trained model.

3.2 Transparency Techniques

The goal of the selected methodology is to visualize the contribution of the different areas in an input image to the output of a neural network using transparency techniques. The transparency techniques provide an encoding of the same size as the input image, where the contribution of each area in an image is depicted.

Two transparency techniques are selected from the state of the art as described in Section 2.1.3. Since the networks in this thesis are trained with a margin-based loss, the transparency techniques have been selected from the state of the art for embedding networks. The first technique provides pixel-level saliency maps and is proposed by Kobs et al. [KSDH21]. An example for a saliency map generated by this technique is shown in Figure 3.1. It shows the saliency map generated for a neural network trained for an image classification task, which receives a bicycle image as input. The saliency map highlights the contribution of the input image areas to the classification of the input image. The second technique generates point-specific saliency maps and is proposed by Zhu et al [ZYC21]. An example for the saliency maps generated by this technique is shown in Figure 3.2. It shows the saliency maps generated by a face verification network, which receives two different images of the same person as input. The overall saliency maps then highlight the areas in the images which are considered similar. Additionally, the point-specific saliency maps highlight similarities between a point in one image in comparison to the other image.

The techniques have been selected due to their detailed visualization of the contribution to the output embedding. The visualization provided by Kobs et al. [KSDH21] produces an encoding by calculating the contribution on a pixel basis. This allows an exact analysis of the significance of the input image areas, which is beneficial for images containing scattered information such as handwritten text. The technique provided by Zhu et al. [ZYC21] further enhances its visualization of the contribution by providing information on similarities between two images. Additionally, it provides information on



(a) Input Image (b) Generated Saliency Map

Figure 3.1: Saliency map generated by the transparency technique proposed by Kobs et al. [KSDH21] for the given input image. The input image is taken from the SOP dataset [OSXJS16].

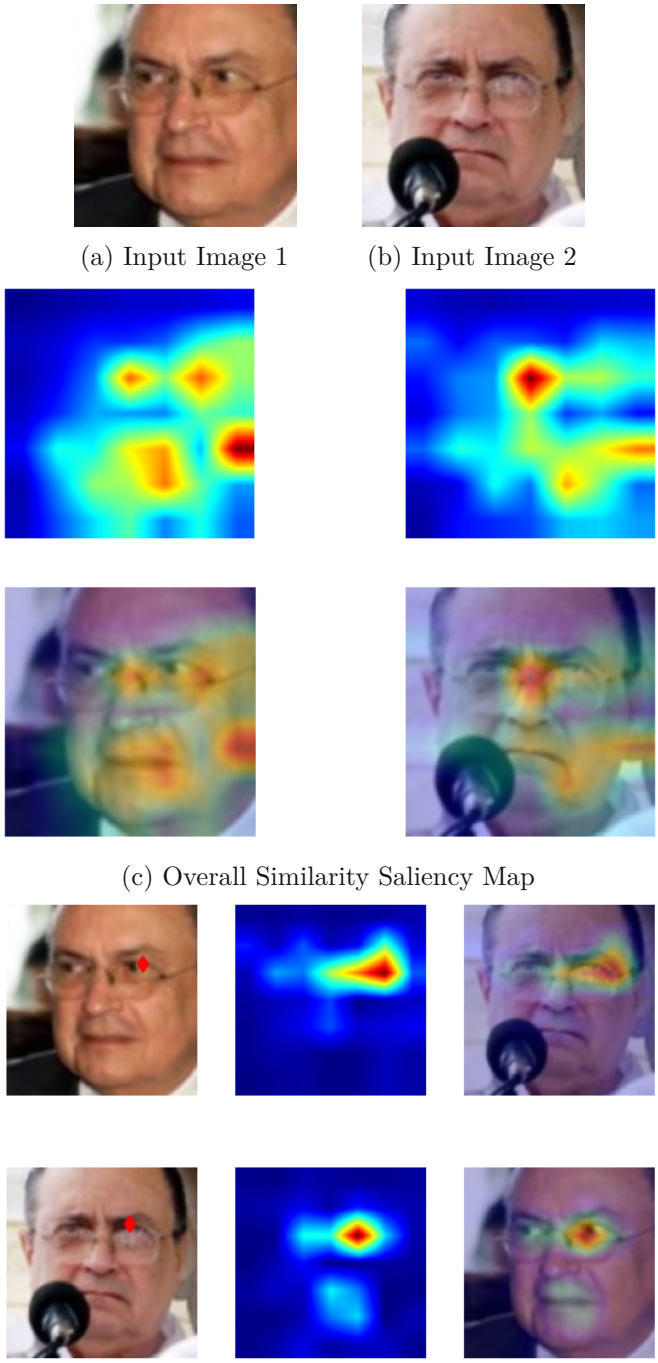
the similarity between one point in an image and the overall comparison image. This can be used for neural networks trained on WI and WV to detect similar patterns in characters, where a point on a character is selected in one image and the visualization provides information on areas with a similar pattern in the other image. Another advantage of the selected methods in comparison to other state-of-the-art techniques is the independence from auxiliary inputs. The selected techniques do not require additional information such as image attributes for the contribution calculation but require the trained model and input images only.

3.2.1 Pixel-Level Saliency Map

The first technique is based on the transparency technique proposed by Kobs et al. [KSDH21]. Their technique provides a visualization at pixel level using a gradient-based approach. Pixels are highlighted based on their contribution to the embedding output of the underlying neural network. If a pixel is important for the embedding output, a change of the pixel should lead to a significant change in the embedding output. The saliency map is generated using the difference between an input image I and a base image. The authors propose the use of an image containing black pixels as base image. For this methodology, an image containing only white pixels is used as it represents a white sheet of paper without information in the form of handwriting. An example of the two image types is shown in Figure 3.3. For the generation of the saliency map, the following gradients are calculated [KSDH21]:

$$s(I) = \partial d(\mathbf{x}_I, \mathbf{x}_{base})/\partial I, \tag{3.1}$$

where $d()$ is a distance function for the two embedding vectors. For this methodology, the Cosine Similarity is used as the underlying neural networks are trained with the

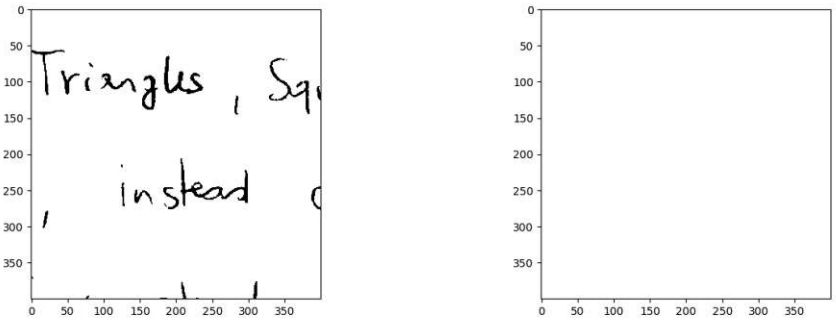


(a) Input Image 1 (b) Input Image 2

(c) Overall Similarity Saliency Map

(d) Point-Specific Saliency Map

Figure 3.2: Saliency maps generated by the transparency technique proposed by Zhu et al. [ZYC21] for the given input images. The overall saliency maps highlight similarities between both images, while the point-specific saliency maps highlight similarities between one image in comparison to one point in the other image (displayed as a red diamond). The input images are taken from the LFW dataset [HRBLM07].



(a) Input Image (b) Base Image

Figure 3.3: The input and base image for the pixel-wise saliency map generation.

Cosine Similarity as well. $\mathbf{x}_I$ and $\mathbf{x}_{base}$ are the embeddings for image I and the base image, respectively. A higher value for the gradients indicates a higher impact of the according pixel value on the embedding output of the neural network. The generated map is then normalized between 0 and 1, where 1 implies a high contribution and 0 implies no contribution to the network output. In order to reduce the noise of the gradients, the authors propose the use of the Smooth-Grad method by creating n image variants by applying Gaussian Noise N to the input image. The resulting gradients are then averaged to receive the final gradient [KSDH21]:

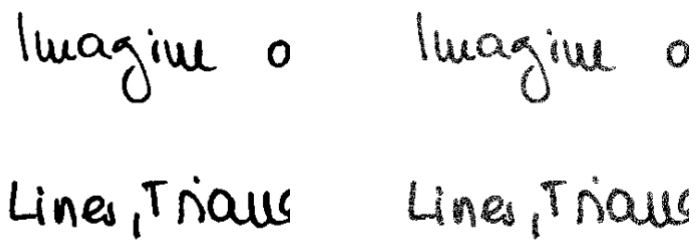
$$s'(I) = \frac{1}{n} \sum_{1}^{n} s(I + N). \tag{3.2}$$

However, for the scope of this thesis, the Gaussian Noise is replaced by random deletion of black pixels from the image since the input images are binarized. The value for n is set to 4. An example of an input image and the applied noise is shown in Figure 3.4.

3.2.2 Point-Specific Saliency Map

The second transparency technique is based on the methodology proposed by Zhu et al. [ZYC21]. This technique takes image pairs as input and creates a visualization, which describes the areas of interest for the similarity between the images using activation decomposition. The technique provides two types of saliency maps. The overall saliency map displays which areas in the input images contribute towards the similarity between the two images and is generated by decomposition of the similarity along the images. The point-specific saliency map provides information on the similarity of one point, i.e. one pixel, in one image to all areas in the other image by decomposition of the according point along the other image.

For a CNN architecture with a GAP and FC layer without bias after the last convolutional layer, the method can be applied as follows [ZYC21]. The GAP layer is a



(a) Input Image (b) Image with deletion of a random amount of black pixels.

Figure 3.4: The original input image and the image with applied noise for the pixel-wise saliency map calculation.

linear component which can be written together with the FC layer as

$$S_c = \sum_k w_{k,c}(\frac{1}{z}\sum_{i,j}\mathbf{A}_{i,j,k}), \tag{3.3}$$

where S_c is the score of class c before the softmax layer and $w_{k,c}$ denotes the weights parameter of the k-th channel for class c for the FC layer. The number z is the normalization term of the GAP layer and $\mathbf{A}_{i,j,k}$ is the output of the last convolutional layer, i.e. its k-th feature map, at position (i, j). z can be calculated by the size of the feature map, i.e. $m \cdot n$ where m and n are the first and second dimension of the matrix $\mathbf{A}\ \epsilon\ \mathbb{R}^{m\times n\times p}$, respectively. Equation 3.3 can be rewritten as [ZYC21]

$$S_c = \frac{1}{z}\sum_{i,j}(\sum_k \omega_{k,c}\mathbf{A}_{i,j,k}), \tag{3.4}$$

where $\sum_k \omega_{k,c}\mathbf{A}_{i,j,k}$ is the decomposition of S_c along (i, j). The method extends this formula for a comparison between two images q and r using their similarity by decomposition along two points (i, j) of q and (x, y) of r, respectively. For this purpose, the Cosine Similarity metric is used. The Cosine Similarity metric is defined as [ZYC21]

$$S = \frac{(\mathbf{e}_q \cdot \mathbf{e}_r)}{|\mathbf{e}_q||\mathbf{e}_r|}, \tag{3.5}$$

where $\mathbf{e}_q$ and $\mathbf{e}_r$ are the embeddings of the two images q and r, respectively, which are generated by the underlying neural network. Equation 3.5 can be rewritten as [ZYC21]

$$\begin{aligned} S &= \frac{\sum_k GAP(\mathbf{A}_k^q) GAP(\mathbf{A}_k^r)}{|E^q||E^r|} \\ &= \frac{1}{z} \sum_k (\sum_{i,j} \mathbf{A}_{i,j,k}^q \sum_{x,y} \mathbf{A}_{x,y,k}^r) \\ &= \frac{1}{z} \sum_{i,j,x,y} (\sum_k \mathbf{A}_{i,j,k}^q \mathbf{A}_{x,y,k}^r), \end{aligned} \tag{3.6}$$

where $\mathbf{A}^q$ and $\mathbf{A}^r$ represent the feature maps of the last convolutional layers for q and r. (i,j) is a point in q and (x,y) is a point in r. The number z is here calculated as $z = m^q \cdot n^q \cdot m^r \cdot n^r \cdot |E^q| \cdot |E^r|$ for $\mathbf{A}^q \, \epsilon \, \mathbb{R}^{m^q \times n^q \times p^q}$ and $\mathbf{A}^r \, \epsilon \, \mathbb{R}^{m^r \times n^r \times p^r}$. The point-specific saliency map for the two points $(i,j),(x,y)$ is then defined as $\sum_k \mathbf{A}_{i,j,k}^q \mathbf{A}_{x,y,k}^r$. The overall saliency map for one image is calculated as $\sum_{i,j}(\sum_k \mathbf{A}_{i,j,k}^q \mathbf{A}_{x,y,k}^r)$, i.e., a summation over all pixels for one image [ZYC21].

These formulations can be used to generalize the methodology for more complex architectures [ZYC21]. Given a flattening layer for the feature output of the last convolutional layer $\mathbf{A} \, \epsilon \, \mathbb{R}^{m \times n \times p}$ with the output of the flattening layer as $\mathbf{A}' \, \epsilon \, \mathbb{R}^{mnp}$, all following linear components can be formulated as one linear transformation [ZYC21]:

$$g(A') = \mathbf{W}'\mathbf{A}' + B = \Sigma_{i,j} \mathbf{W}_{i,j} \mathbf{A}_{i,j} + B, \tag{3.7}$$

where $\mathbf{W}_{i,j}$ is the weights matrix at position (i,j) and B is the bias term. Therefore, Equation 3.6 can be written for this case as

$$\begin{aligned} Sz &= g^q(\mathbf{A}^q) \cdot g^r(\mathbf{A}^r) \\ &= (\sum_{i,j} \mathbf{W}_{i,j}^q \mathbf{A}_{i,j}^q + B^q) \cdot (\sum_{x,y} \mathbf{W}_{x,y}^r \mathbf{A}_{x,y}^r + B^r). \end{aligned} \tag{3.8}$$

This equation can be formulated as

$$\begin{aligned} Sz = \sum_{i,j,x,y} &(\mathbf{W}_{i,j}^q \mathbf{A}_{i,j}^q) \cdot (\mathbf{W}_{x,y}^r \mathbf{A}_{x,y}^r) \\ &+ \sum_{i,j} (\mathbf{W}_{i,j}^q \mathbf{A}_{i,j}^q \cdot B^r) \\ &+ \sum_{x,y} (\mathbf{W}_{x,y}^r \mathbf{A}_{x,y}^r \cdot B^q) \\ &+ B^q \cdot B^r, \end{aligned} \tag{3.9}$$

where S is the Cosine Similarity metric and z is a normalization term. The point specific map is then calculated as $I(x,y) = (\mathbf{W}_{i,j}^q \mathbf{A}_{i,j}^q) \cdot (\mathbf{W}_{x,y}^r \mathbf{A}_{x,y}^r)$, where (i,j) are the coordinates of the selected point [ZYC21].

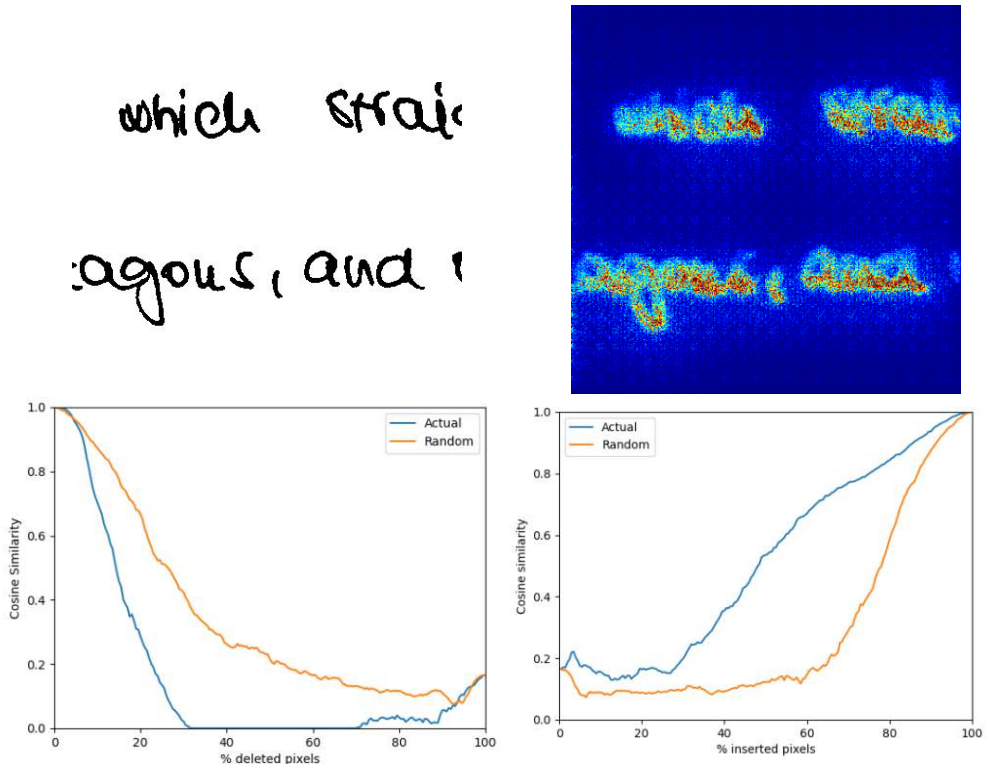


Figure 3.5: Example of deletion and insertion scores calculated for a saliency map.

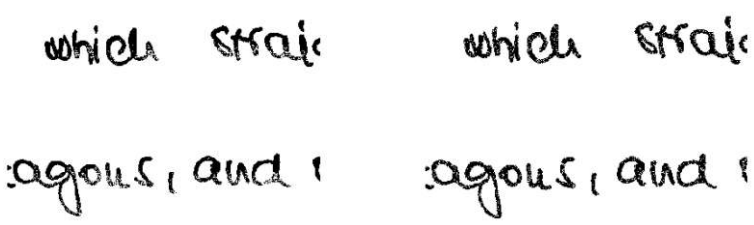
The authors showed in their experiments that the use of the bias term B in the equations decreases the performance of their approach [ZYC21]. Therefore, for the scope of this thesis, the bias term is removed for this technique.

3.3 Metrics

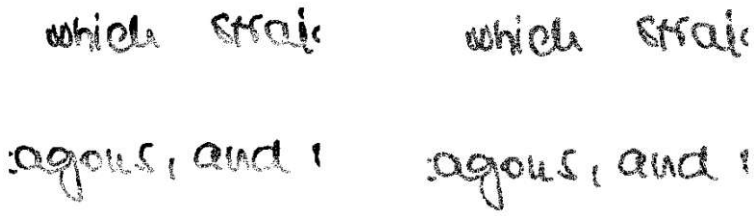
For the evaluation described in Chapter 5, the deletion and insertion scores, as described in Section 2.1.1, have been adapted for use with the selected transparency techniques. These metrics measure the change in similarity between a query and a retrieved image when the retrieved image is altered based on its given saliency map [HVH22]. For use with the selected transparency techniques, the query and retrieval images are the same image. As the input images are binarized for the neural networks, the deletion and insertion scores only affect black pixels, i.e. pixels, which are part of the handwriting. Pixels deleted during the calculation of the score are set to white. This results in a more realistic deletion as the pixels are set to the background colour and the handwriting information is therefore progressively removed from the image. Additionally, the insertion score uses a white image as the base, to which the pixels of the handwriting are gradually added. The Cosine Similarity is used as measurement of similarity between the original

and adjusted image, with the minimum similarity set to zero. An example for the deletion and insertion scores is shown in Figure 3.5. The blue curve displays the change in similarity for the actual saliency map, while the orange curve displays the change in similarity for the random deletion and insertion of pixels. For this input image, the lines diverge for the deletion score, with the similarity for the actual saliency map decreasing quicker as the significant pixels are deleted first, therefore reducing the similarity between the embedding outputs for the original and adjusted images more drastically. After the deletion of approximately 90% of black pixels, the curves converge as the adjusted images for both cases become more similar until all black pixels are deleted and the adjusted images are both completely white. For the insertion score, a similar behaviour is observable. Here, the curve of the actual saliency map displays a higher increase in similarity than for the random insertion. After the insertion of approximately 80% of black pixels, the cosine similarities of both cases become more similar as the adjusted images become more similar to the original image. The change in the original image during this process is shown in Figure 3.6 for the deletion metric and Figure 3.7 for the insertion metric.

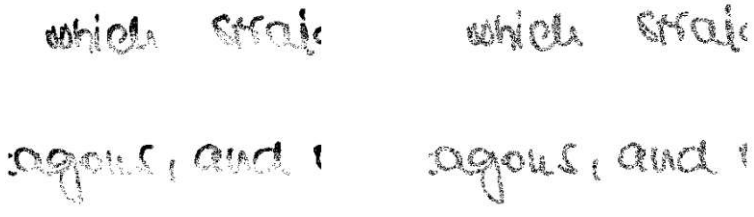
For the quantitative evaluation described in Section 5.2.2, the data is prepared as described in Section 3.1.3, with the pages divided into a grid of snippets. Additionally, snippets with less than 2% black pixels are discarded. For each snippet, the deletion and insertion scores for the generated saliency maps are compared with the scores calculated for a random deletion and insertion of pixels.



(a) Removed pixel: 10%



(b) Removed pixel: 30%

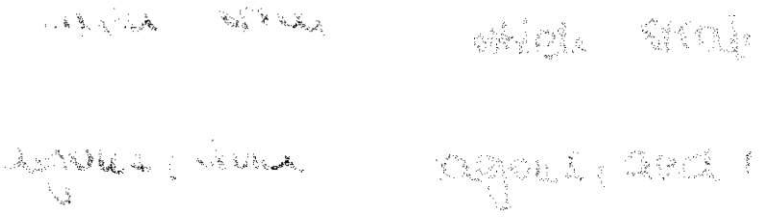


(c) Removed pixel: 50%

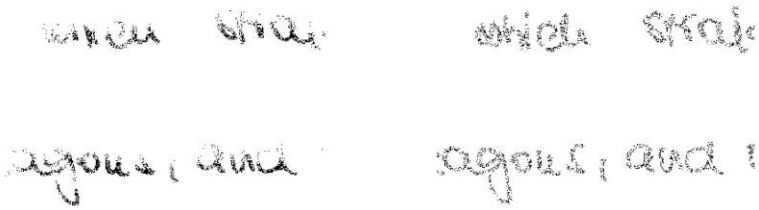
(d) Removed pixel: 70%

(e) Removed pixel: 90%

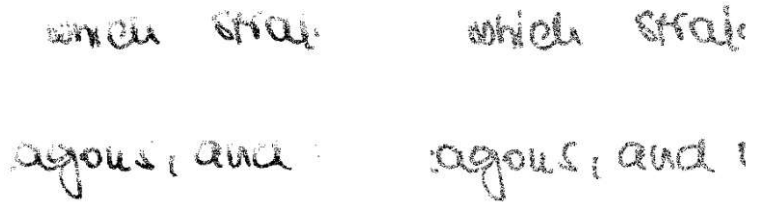
Figure 3.6: Example of deletion progress according to the saliency map given in Figure 3.5 (left) and a random deletion (right).



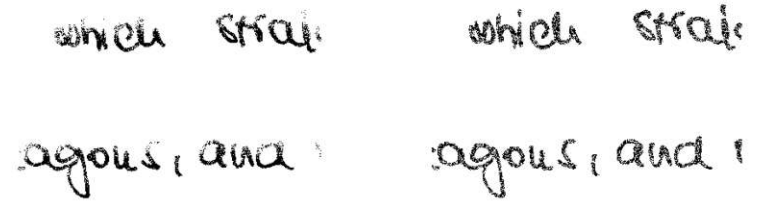
(a) Inserted pixel: 10%



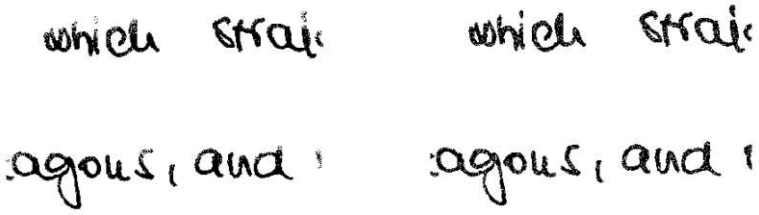
(b) Inserted pixel: 30%



(c) Inserted pixel: 50%



(d) Inserted pixel: 70%



(e) Inserted pixel: 90%

Figure 3.7: Example of insertion progress according to the saliency map given in Figure 3.5 (left) and a random insertion (right).

CHAPTER 4

Datasets

State-of-the-art datasets used in the context of WI and WV provide data on handwritten text. Existing datasets can be split into two groups, offline and online handwritten data. For the creation of online data, digital writing equipment such as a tablet is used. The technology tracks temporal data such as pressure and location of the pen. For offline data, only the handwriting itself is given as an image [CBMA15, Chr19]. This thesis focuses on offline datasets. The existing datasets provide handwritten text in multiple languages, including English, German, Greek, Dutch and Arabic. Moreover, they contain contemporary as well as historical handwritten data. For example, the ICDAR 2017 Historical-WI dataset contains writings created between the 13th and 20th century [Chr19].

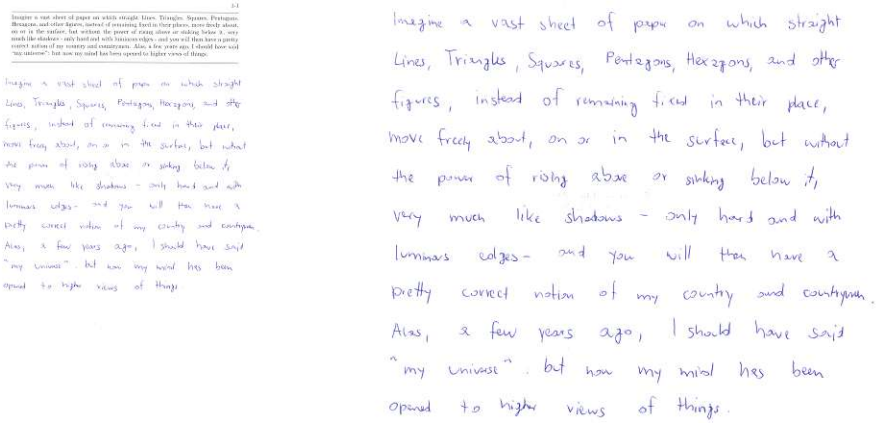
In the scope of this thesis, offline datasets containing contemporary handwriting are used for the training and evaluation of WI and WV neural networks. For this purpose, the CVL [KFDS13], the Firemaker [BSV03] and the ICDAR2013 [LGSP13] datasets have been selected. All three datasets contain modern handwriting and are commonly used for the training and evaluation of WI methodologies. Furthermore, the CVL and ICDAR2013 datasets provide text in two languages each, English and German as well as English and Greek, respectively. All three datasets contain offline handwritten data. A summary of the selected datasets and their characteristics will be presented in the following section. The splits used for the datasets can be found online [1].

4.1 CVL

The CVL dataset [KFDS13] contains 1604 handwritten pages. 27 writers contributed seven handwritten pages each, while 283 writers provided five pages. One page per author

[1] https://github.com/VCPY/dataset_split_cvl_firemaker_icdar2013, accessed 1 November 2023.

is written in German, while the others are written in English. The text is copied from a provided machine-printed text, which is taken from different English and German literary texts. During the acquisition process, the writers were asked to use a ruled undersheet to maintain a straight line for handwriting. The dataset contains full pages consisting of machine-printed and handwritten text, as well as cropped versions containing only the handwritten text [KFDS13]. An example of both the full and cropped version is shown in Figure 4.1. For the purpose of this thesis, only the cropped versions of the



(a) The original page. (b) The cropped version of the page.

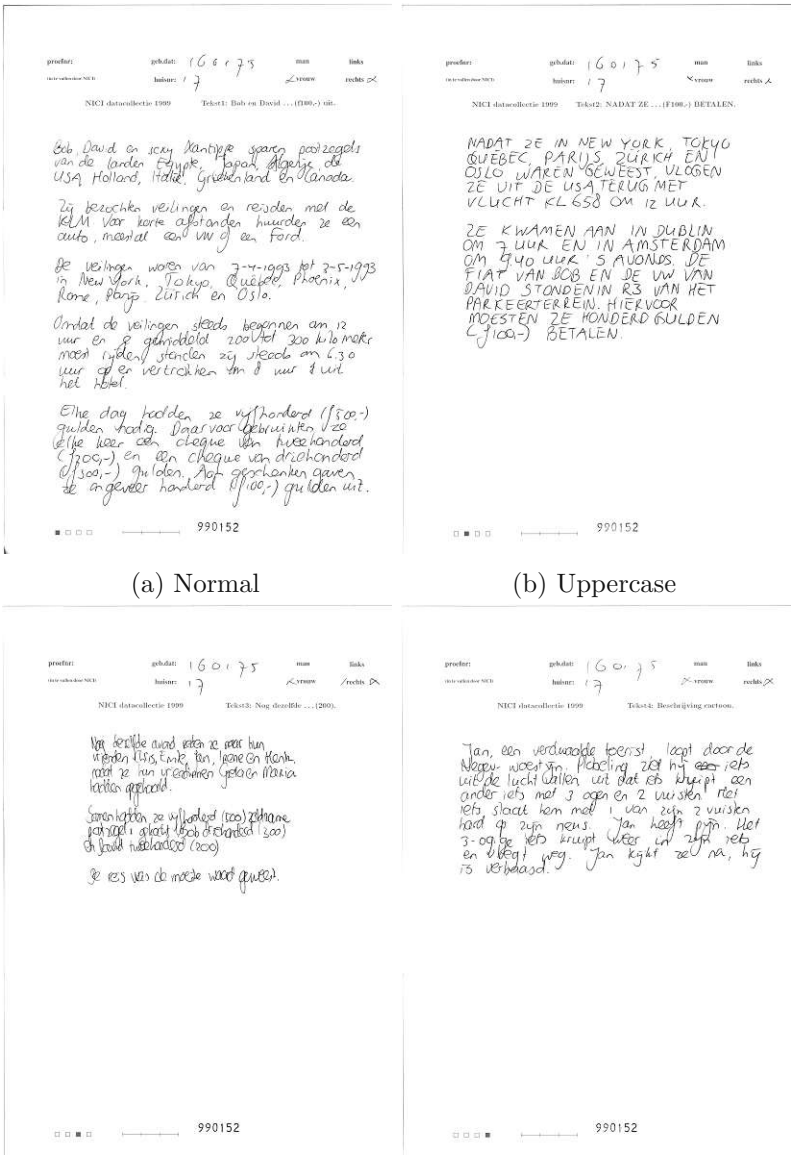
Figure 4.1: Examples of the handwritten pages taken from the CVL dataset [KFDS13].

samples are used. Additionally, the pages for the writer with ID 431 have been removed from the dataset due to two of the pages being completely blank. The original dataset provides a split into train- and test set, where the train set contains the pages of 27 writers. However, in order to have more data available for the training of the networks, the dataset is split into two open sets, with both sets having 50% of the amount of unique writers. Due to an overall uneven number of writers, the first set has one writer more. Additionally, the number of pages in each set is uneven due to the random selection of writers for the split and the uneven number of pages written by each person. The first set is used as open train and validation set. The second set is used as test set.

4.2 Firemaker

The Firemaker dataset [BSV03] contains 1000 handwritten pages from 250 authors, yielding four pages per author. Each page is written with a different characteristic: The text of the first page was copied by the writers from a given machine-printed text. The second page contains text written entirely in uppercase letters. The writers were asked to forge their own handwriting for the third page. Finally, for the fourth page, the writers were asked to describe a given comic in their own words [BSV03, HS20, WTB14]. The recording conditions for all writers are standardized, with all writers using the

same writing equipment. The authors note that therefore differences in the handwriting are unlikely to be caused by variations in the recording conditions but rather reflect differences in handwriting styles [BSV03]. An example of four pages written by one writer is shown in Figure 4.2. For this thesis, the pages of the dataset are cropped to remove



(a) Normal (b) Uppercase

(c) Forged (d) Free Text

Figure 4.2: Examples of the handwritten pages taken from the Firemaker dataset [BSV03].

the machine-printed information on the sheets. Similar to the CVL dataset, the dataset is split in half to create an open train and an open test set. The authors of the dataset removed the pages containing forged text and uppercase letters for their experiments.

This is also done for the experiments in the scope of this thesis. The use of the dataset in this thesis is described in more detail in Section 5.1.

4.3 ICDAR2013

The ICDAR2013 dataset [LGSP13] contains handwritten data used for benchmarking the submitted methods of the ICDAR2013 Competition on Writer Identification. It consists of 1000 pages written by 250 writers. Each writer copied four texts, where two are written in English and two are written in Greek. The number of lines per page varies between two and six lines, and the generated images are binarized [CBH+17, LGSP13]. An example of four pages written by one writer is shown in Figure 4.3. The authors of

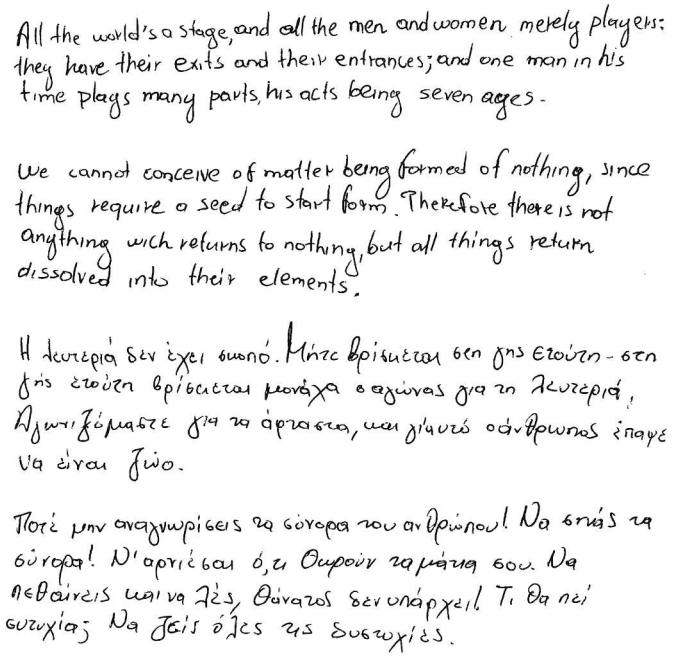


Figure 4.3: Examples of the handwritten pages taken from the ICDAR2013 dataset [LGSP13].

the dataset provide an official split of the data into train and test set, where the train set contains 400 pages and the test set contains 1000 pages. However, in order to have the same percental split into train and test set as used for the CVL dataset, the ICDAR2013 dataset is split equally into open train and test set.

CHAPTER 5

Results and Discussions

The previous chapter introduced the methodology proposed for the use of transparency techniques on WI and WV neural networks. This methodology is evaluated in this chapter. First, an experiment, which is conducted on the underlying ResNet networks to improve their performance, is introduced. The experiment explores the effect of the hyperparameter configuration on the accuracy of the neural network for the CVL, Firemaker and ICDAR2013 datasets. Afterwards, the transparency techniques and their performances are evaluated using sanity checks, quantitative and qualitative evaluation methods. Finally, the achieved results are compared and discussed.

5.1 Hyperparameter Configuration for ResNet Training

Initially, the selected WI ResNet architectures as described in Section 3.1.1 achieve a higher performance on the CVL dataset compared to the Firemaker and ICDAR2013 datasets. For example, the ResNet18 architecture achieves a mAP of 96.09% on the CVL test dataset, but a lower mAP of 40.67% on the Firemaker and 31.37% on the ICDAR2013 test set after 100 epochs. To improve the performance, a comparison of different hyperparameter configurations for the training for a WI task is conducted with both datasets. For each configuration, four models are trained, with the dataset split into open train and test set as described in Chapter 4. 4-fold cross-validation is used for the training of these models. The results shown in this section display the average test mAP of all four models on the test set.

The evaluation is done with a ResNet18 model using Triplet Loss with a margin of 0.1 and the Cosine Similarity as distance measurement. The model uses an Adam Optimizer with a learning rate of 0.01. The models are trained for 100 epochs with input images with a size of 400px $\times$ 400px. The loss function and image size are based on the training process suggested by Wang et al [WMC21].

Pages	mAP
2	**59.40**
3	35.66
4	17.10

Table 5.1: Performance of the ResNet18 model on the test set of the reduced versions of the Firemaker dataset. Values are given in percent.

For the training and evaluation of the models, the Firemaker and ICDAR2013 datasets have been adjusted as follows to achieve a better performance.

Firemaker For the Firemaker dataset, the number of pages per writer is reduced by removing forged and uppercase pages from the dataset. This is done in accordance to Bulacu et al. [BSV03], who removed these two pages for their experiments as well [HS20, WTB14]. The model achieves a mAP of 59.40% on this adjusted dataset as shown in Table 5.1. Here, the model is trained and evaluated with four, three or two pages. For the case of three pages, the forged page has been removed from the dataset, while for two pages, the page containing only uppercase letters has been removed additionally. The table shows that the model performs best when only two pages per writer are used. Therefore, the dataset is adjusted, and the pages containing forged text and uppercase letters are removed.

ICDAR2013 The ICDAR2013 dataset contains pages written in English and Greek. However, the CVL dataset contains multiple pages per author written in English as well, while neither the CVL nor the Firemaker dataset contain pages written in Greek. Therefore, to provide a dataset focused on a different language, the dataset is adjusted by removing the English pages for the remaining training runs and evaluation.

For the hyperparameter search, the batch size and the number of random samples taken from each page per epoch are varied. Additionally, the model is initialized with and without pretrained weights. For the training runs, if not stated otherwise, the batch size is set to 128, the number of random samples per page is set to one and the model is not initialized with pretrained weights.

Batch Size The value chosen for the batch size does impact the accuracy of the model for both datasets. The value is set to 32, 64 and 128 for different training runs. Table 5.2 shows the mAP value for the test sets of both datasets. The test mAP diverges for the models trained with different batch sizes. The model trained on the ICDAR2013 dataset performs best with a batch size of 32, while the model trained on the Firemaker dataset performs best with a batch size of 128.

Batch Size	Firemaker	ICDAR2013
32	52.42	**49.48**
64	57.03	47.27
128	**58.39**	40.62

Table 5.2: Performance of the ResNet18 model trained with different batch sizes. The table displays the average mAP score of the four models for the test set. Values are given in percent.

Weights	Firemaker	ICDAR2013
Pretrained	**76.56**	**66.90**
Not Pretrained	54.86	37.71

Table 5.3: Performance of the ResNet18 model initialized with and without pretrained weights. The table displays the average mAP score for the four trained models for the test set. Values are given in percent.

Pretrained Weights The performance of the model also increases when the model is initialized with pretrained weights provided by PyTorch [PGM+19]. This is shown in Table 5.3. Here, a comparison of two models, where one is initialized with pretrained weights, while the other model is trained without pretrained weights, can be seen. The average mAP score shows that for both datasets, the models initialized with pretrained weights perform better than the models without pretrained weights. For the Firemaker dataset, the mAP score increases by 22 percentage points, while the score increases by 29 percentage points for the ICDAR2013 dataset.

Number of Samples The performance of the model can be increased by raising the number of random samples taken from one page per epoch during training. This does not affect the number of samples taken from one page for the validation and test set. Table 5.4 shows the difference in accuracy when increasing the number of samples per page. Here, the achieved mAP score increases with an increased number of samples, with the exception of seven samples for the Firemaker dataset, where the accuracy slightly decreases in comparison to five samples, and five samples for the ICDAR2013 dataset, where the accuracy decreases by four percentage points in comparison to three samples.

Overall, the highest performance for the Firemaker dataset is achieved by a pretrained model, which is trained with a batch size of 128 and nine snippets per page. This model achieves a test mAP of 88.77%. For the ICDAR2013 dataset, the highest performance is achieved by a pretrained model, which is trained with a batch size of 32 and nine samples per page. This model achieves a test mAP of 80.96%. These configurations are used for the training of the WI and WV networks and subsequently used for the evaluation of the transparency techniques.

Num Samples	Firemaker	ICDAR2013
1	60.50	39.62
3	61.36	48.33
5	69.88	44.49
7	68.29	51.29
9	**72.74**	**56.21**

Table 5.4: Performance of the ResNet18 model trained with different numbers of samples. The table displays the average mAP score of the four models for the test set. Values are given in percent.

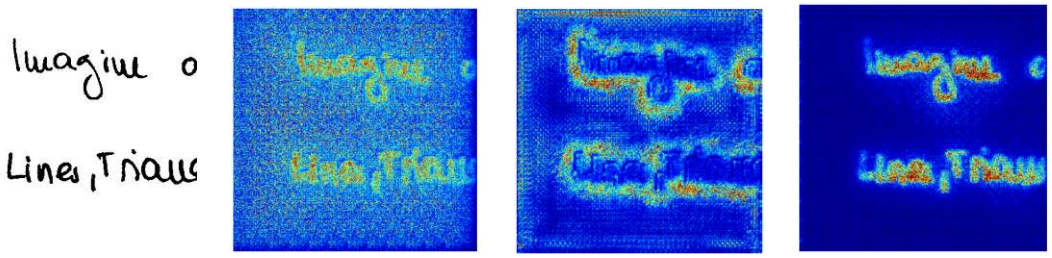
5.2 Evaluation of Transparency Techniques

In this section, the evaluation of the transparency techniques is described. First, a sanity check with a ResNet18 model and different types of weights is performed, which is described in Section 5.2.1. Then, the evaluation metrics for the quantitative evaluation and the results of the evaluation process are described in Section 5.2.2. Afterwards, a qualitative evaluation is conducted by comparing highlights of the occurrences of the same characters. The results are presented in Section 5.2.3. Finally, the results of the evaluation are discussed, and the applicability of the selected transparency techniques on neural networks trained on WI and WV assessed, which is described in Section 5.2.4.

5.2.1 Sanity Check

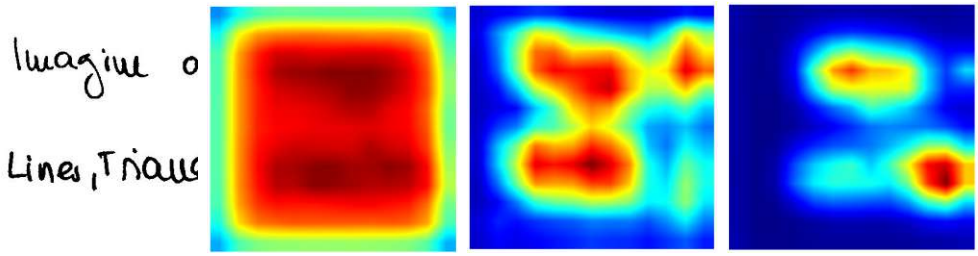
In order to validate the responsiveness of the created saliency maps to the trained model, a sanity check, as proposed by Arras et al. [AOS22], is conducted. This evaluation allows to analyse if the explanations created by a transparency technique correlate with the parameters of a model and differ between a trained and an untrained model. The results of this analysis do not provide any information on the correctness of the visualization but are considered an additional verification [AOS22].

The sanity check is conducted for both transparency techniques using a ResNet18 model. The saliency maps are calculated for an untrained model, a model initialized with pretrained weights provided by PyTorch [PGM+19] and a model trained on WI. The results for the pixel-wise saliency maps are shown in Figure 5.1. Here, the calculated saliency maps for the given input image are displayed. As can be seen, the saliency maps differ for each model type. The untrained model does not focus on certain patterns in the image but creates its output based on random pixels and image areas. The pretrained model, in contrast, focuses on the contour of the handwriting. The trained WI model uses the pixels of the handwriting for the creation of its output. This shows that the pixel-wise saliency map technique is responsive to the parameters of the model. The result for the point-specific saliency maps are shown in Figure 5.2. The differences in the saliency maps show the responsiveness of the technique to the parameters of the model.



(a) Input image (b) Saliency map for untrained model (c) Saliency map for pretrained model (d) Saliency map for model trained on WI

Figure 5.1: Pixel-wise saliency maps calculated for three ResNet18 models with different weights.



(a) Input image (b) Saliency map for untrained model (c) Saliency map for pretrained model (d) Saliency map for model trained on WI

Figure 5.2: Point-specific saliency maps calculated for three ResNet18 models with different weights.

The saliency map for the untrained model places the highest significance at the middle of the image. The pretrained model focuses more on certain areas of the handwritten texts. In contrast, the trained WI model focuses on specific and, in comparison to the pretrained model, small areas of the handwritten text.

These experiments show that both selected transparency techniques are responsive to the parameters of the underlying neural network and create corresponding visualizations.

5.2.2 Quantitative Evaluation

For the quantitative evaluation of the transparency techniques, the insertion and deletion metrics as described in Section 3.3 are used. For the evaluation, the underlying neural networks are trained with the train sets of the CVL, Firemaker and ICDAR2013 datasets as described in Section 3.1.3. Additionally, the training configurations and adjustments to the datasets, as described in Section 5.1, are applied. However, the neural networks for this evaluation are not initialized with pretrained weights, although the experiments show that the neural network performance on the Firemaker and ICDAR2013 datasets can be improved by using pretrained weights. The sanity checks described in Section 5.2.1 show

		WI (mAP)	WV (accuracy)
CVL	ResNet18	**86.41**	**90.48**
	ResNet20	56.43	67.58
	ResNet50	82.41	88.52
Firemaker	ResNet18	**70.73**	**93.4**
	ResNet20	39.66	68.4
	ResNet50	57.64	86.0
ICDAR2013	ResNet18	**62.08**	**91.57**
	ResNet20	47.10	76.14
	ResNet50	53.98	90.43

Table 5.5: Scores achieved on the test sets of the CVL, Firemaker and ICDAR2013 datasets by the model with the highest test mAP in the case of WI neural networks and highest accuracy in the case of WV neural networks. Values are given in percent.

that the model with pretrained weights highlights regions around the handwriting present in the image, which is displayed in Figure 5.1c. Models trained on WI and initialized with these weights consider the regions around the handwriting as well. Due to the characteristic of the insertion and deletion score, namely the insertion and deletion of black pixels, pixels which are part of the background are not considered. Therefore, the saliency maps created when using pretrained weights are not suitable for these evaluation metrics. In order to create saliency maps, which can be evaluated with these metrics, the neural networks are trained without pretrained weights. The training procedure uses 4-fold cross-validation. For the evaluation, the model with the highest test mAP in the case of WI neural networks and the highest accuracy in the case of WV neural networks is used. The test mAP scores and accuracies achieved by the neural networks are shown in Table 5.5. The results show that all WV neural networks perform better than their WI equivalents. Moreover, the WI and WV ResNet18 networks provide the best results for all datasets, while the WI and WV ResNet20 networks provide the lowest results for all datasets.

The results of the quantitative evaluation for the deletion and insertion score calculations are shown in Table 5.6 and Table 5.7, which display the percentage of snippets for which the deletion and insertion scores are better in comparison to a random insertion and deletion, i.e. where the AUC value for the saliency map is lower for the deletion score and higher for the insertion score. The transparency technique as proposed by Zhu et al. [ZYC21] provides two types of saliency maps as described in Section 3.2.2. However, the point-specific saliency map requires a manual selection of a point in the image. A manual selection is not feasible due to the large amount of snippets. Therefore, only the pixel-level saliency map proposed by Kobs et al. [KSDH21] and the overall saliency map proposed by Zhu et al. [ZYC21] is used for the quantitative evaluation.

		Pixel-Level		Point-Specific	
		Del.	Ins.	Del.	Ins.
CVL	ResNet18	89.66	95.27	**25.57**	97.66
	ResNet20	12.87	95.00	8.61	**100.00**
	ResNet50	55.57	98.30	8.30	98.76
Firemaker	ResNet18	**97.23**	**100.00**	8.30	99.85
	ResNet20	15.64	95.69	22.77	99.77
	ResNet50	51.33	89.29	21.42	99.25
ICDAR2013	ResNet18	82.31	99.88	8.01	99.64
	ResNet20	15.08	99.64	7.22	**100.00**
	ResNet50	71.97	99.68	7.30	99.84

Table 5.6: Percentage of calculated deletion and insertion scores, where the saliency map performs better than random deletion and insertion, i.e. the deletion AUC score is lower and the insertion AUC score is higher for the saliency map. Values are calculated for neural networks trained on WI.

WI networks The scores in Table 5.6 show the quantitative evaluation results for the WI neural networks. They indicate that the performance of the pixel-level saliency maps is dependent on the underlying neural network architecture. This can be seen for the deletion scores of all three networks. While the mAP scores of the three networks vary for all three datasets, the deletion scores for the pixel-level saliency maps do not show this amount of variation. For example, the ResNet50 network achieved a mAP of 82.41% on the CVL dataset and mAP of 57.64% on the Firemaker dataset. However, the pixel-level deletion score percentages for both datasets are similar, with 55.57% for the CVL dataset and 51.33% on the Firemaker dataset. The percentages for the insertion scores for both types of saliency maps are high for the WI neural networks, with all but one value being above 90%. This suggests that the neural networks are able to correctly classify a snippet of handwritten text with little information, i.e. when only a few pixels or areas are inserted into the image and therefore available to the neural network. The results also show that the pixel-level saliency maps perform better than the point-specific saliency maps regarding the deletion score for the WI neural networks. One possible explanation is that the deletion of individual pixels alters important handwriting information, which the network uses to identify the author, more drastically than the deletion of multiple pixels from one area of text, where the network can use other unaltered areas with less significance within the snippet to identify the author.

WV networks The scores in Table 5.7 display the quantitative evaluation results for the WV neural networks. Similar to the WI neural networks, a higher performance on the insertion scores than on the deletion scores is displayed for both saliency maps. All insertion score percentages except two are above 90%. In contrast, all deletion score percentages except one fall below the threshold of 70%. This also indicates that

		Pixel-Level		Point-Specific	
		Del.	Ins.	Del.	Ins.
CVL	ResNet18	67.38	92.28	18.10	97.12
	ResNet20	11.33	84.26	**34.50**	**100.00**
	ResNet50	61.08	96.74	28.61	89.83
Firemaker	ResNet18	57.46	92.19	8.36	99.36
	ResNet20	43.63	98.91	15.28	99.98
	ResNet50	69.15	97.67	13.93	99.02
ICDAR2013	ResNet18	**75.64**	**99.68**	11.61	99.64
	ResNet20	6.24	98.30	10.26	**100.00**
	ResNet50	51.13	95.54	16.27	99.84

Table 5.7: Percentage of calculated deletion and insertion scores, where the saliency map performs better than random deletion and insertion, i.e. the deletion AUC score is lower and the insertion AUC score is higher for the saliency map. Values are calculated for neural networks trained on WV.

the neural networks are able to classify a snippet of handwritten text with only little information available. The results for the WV neural networks show a similar behaviour to the WI neural networks. They indicate that the deletion of individual pixels alters the characteristic of the handwritten text more than the point-specific saliency maps, therefore resulting in a lower deletion score percentage for the point-specific saliency maps. In contrast, the deletion of patches for the point-specific saliency maps removes parts of the information, while other handwritten text, which contains information as well, remains unchanged and can be used for the creation of the embedding output of the neural network. In contrast to the WI neural networks, the performance of the pixel-level saliency maps on the WV neural networks does not display a pattern regarding the underlying neural network. The deletion score percentage for the ResNet20 network is the lowest for all three datasets. However, the insertion score percentage achieved by the ResNet20 is the highest for the Firemaker and ICDAR2013 datasets.

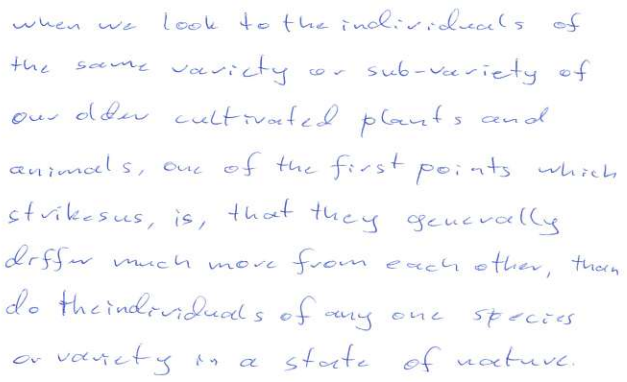
A comparison of the results shows a variation in the results for neural network models trained on WI and WV. For example, the performance of the ResNet18 model shows a difference in the achieved score for the two types of neural networks. While the WI network achieved 89.66%, 97.23% and 82.31% on the CVL, Firemaker and ICDAR2013 datasets, respectively, the WV neural network achieved a lower score for all three datasets, namely 67.38% on the CVL, 57.46% on the Firemaker and 75.64% on the ICDAR2013 dataset. This indicates an influence of the loss function and training procedure on the creation of the saliency map, as both types of networks use the same architectures as backbones but differ in their loss functions and training procedures.

5.2.3 Qualitative Evaluation

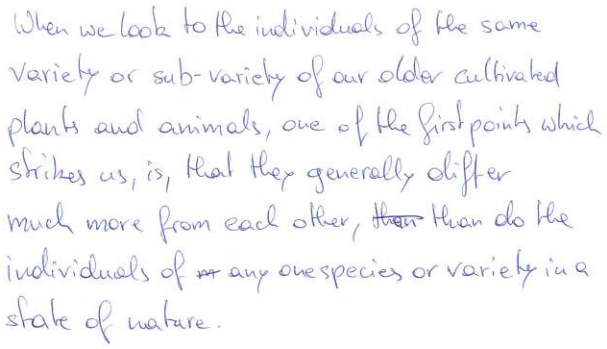
Forensic experts consider characteristics such as spelling idiosyncrasies, ink deposition of a handwriting and cultural influences to determine the author of a handwritten text [Sch08]. Additionally, these experts apply their individual approach by using their personal experience, making the identification of an author a subjective task [Sch08]. Schomaker [Sch08] notes that this knowledge "[...] is partly perceptual and is difficult to verbalize, partly cognitive and explainable to others: colleagues in the forensic domain, criminal investigators, lawyers and judges." ([Sch08], p.12).

This aggravates the comparison of explanations created by a transparency technique with the analysis done by a human expert. Christlein [Chr19] notes that the identification process can include a comparison of occurrences of the same character at different positions in the handwritten text. Therefore, this section analyses the saliency maps on similarities in highlightings of the same character in the same and different image snippets. The intensity of a given highlight is only meaningful in the scope of its snippet. The selected transparency techniques normalize the calculated values of the saliency maps, leading to the existence of at least one area with the highest possible significance highlight in each snippet. Therefore, the highlights do not provide information on absolute values but display a relative significance between areas of the same snippet.

Two images with handwritten text from two different authors are selected from the test set of the CVL dataset for the evaluation of the saliency maps generated by WI and WV neural networks, since all three neural networks achieved a high performance on this dataset as shown in Table 5.5. The selected pages are shown in Figures 5.3a and 5.3b. For the page in Figure 5.3a, the WI ResNet18 and ResNet50 networks return the other four pages written by the same author on top of the ranking, i.e. the page has an average precision of 1, while the ResNet20 ranks three pages of the same author at the top and the page has an average precision of 0.9167. For the page in Figure 5.3b, the retrieval does not work, as the pages written by the same author are ranked outside the top 150 pages. For the WI ResNet18, ResNet20 and ResNet50 networks, the page has an average precision of 0.0408, 0.0246 and 0.0413, respectively. For the WV ResNet18, ResNet20 and ResNet50 networks, the other pages written by the same author as the first page are returned as matches. For the second page, the ResNet20 returns three pages as matches and one page as a non-match. However, for the ResNet18 and ResNet50 networks, all other pages for the second page are returned as non-match. The neural networks have been trained with the CVL dataset beforehand, as described in Section 3.1.3. Therefore, the same models are used as in Section 5.2.2. The saliency maps for the selected images are then calculated for the ResNet18 and ResNet20 networks, since the saliency maps for the ResNet50 network are similar to the saliency maps for the ResNet18 network.



(a)

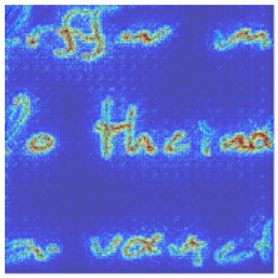


(b)

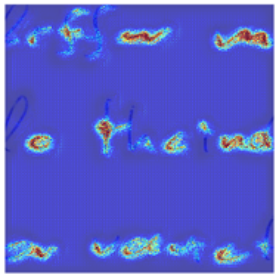
Figure 5.3: Pages taken from the test set of the CVL dataset [KFDS13] as described in Section 4.1 for the evaluation of the transparency techniques.

Pixel-Wise Saliency Map

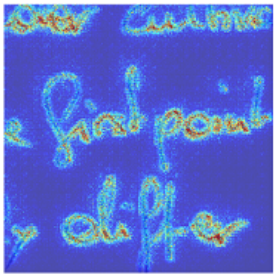
Examples for the saliency maps generated for the WI networks for the first and second page are shown in Figure 5.4. The saliency maps for the first page display similar highlighting patterns for multiple occurrences of the same character. An example is the character "y". Its occurrences are displayed in Figure 5.5. The highlighting pattern differs from network to network but stays consistent for the saliency maps created for one neural network type. For the ResNet18, the right side of the arch as well as the bottom part display a peak highlighting in all cases, with the left side of the arch highlighted less in comparison. For the ResNet20, the upper arch of the character is displayed with a peak highlight, while the bottom arch is highlighted less in comparison. A similar highlighting pattern is also present for the character "f". Examples of occurrences are shown in Figure 5.6. For this character, the highlighting pattern is similar for both networks. The highlighting peaks in the area where the vertical and horizontal lines cross. Additionally, the highlight changes when the horizontal line is positioned close to the



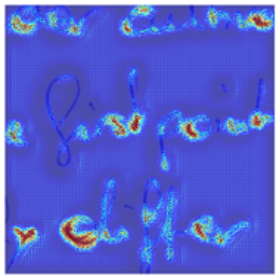
(a) ResNet18 First Page



(b) ResNet20 First Page

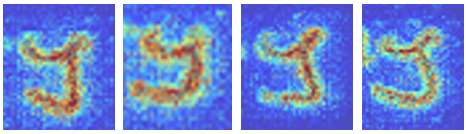


(c) ResNet18 Second Page

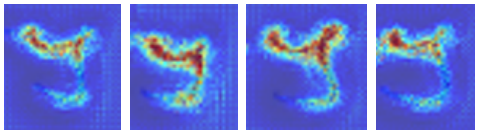


(d) ResNet20 Second Page

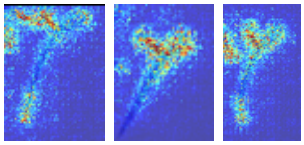
Figure 5.4: Pixel-wise saliency maps for the handwritten text of the first and second page for the WI ResNet18 and ResNet20 networks.



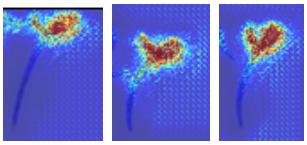
(a) ResNet18 First Page



(b) ResNet20 First Page

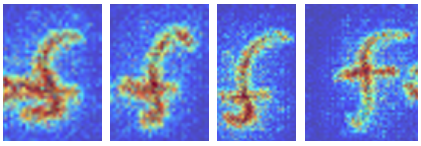


(c) ResNet18 Second Page

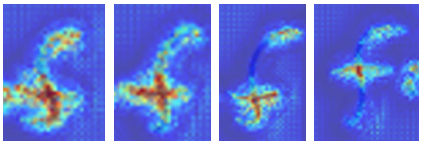


(d) ResNet20 Second Page

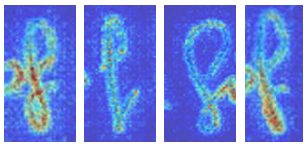
Figure 5.5: Occurrences of the character ”y” in the saliency maps for the first and second page generated for the WI networks.



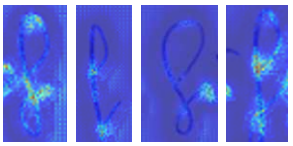
(a) ResNet18 First Page



(b) ResNet20 First Page

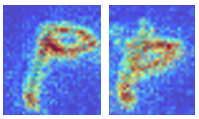


(c) ResNet18 Second Page

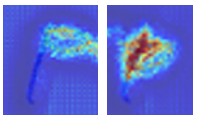


(d) ResNet20 Second Page

Figure 5.6: Occurrences of the character ”f” in the saliency maps generated for the WI networks for the first and second page.



(a) ResNet18



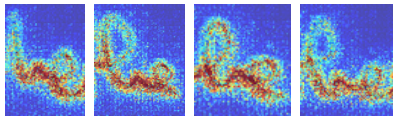
(b) ResNet20

Figure 5.7: Occurrences of the character ”p” in the saliency maps generated for the WI networks for the first page.

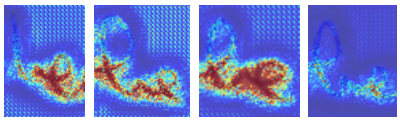
bottom part of the vertical line, where the bottom part of the character is highlighted as well. However, both networks also display deviating highlightings for their occurrences of characters such as ”p”, which occurs twice on the page. The occurrences are shown in Figure 5.7. For the ResNet18, the first occurrence displays the highest significance on the bottom left of the circle and the bottom of the vertical line, while the second occurrence displays the peak highlightings at the top right of the circle and the bottom of the vertical line. For the ResNet20, the first occurrence is highlighted at the top and bottom right of the circle, while the second occurrence is highlighted on the left of the circle.

In contrast to the first page, the saliency maps for the second page display large differences in their highlighting patterns for multiple occurrences of the same character. For example, the character ”f” does not display a similar highlighting, which is shown in Figure 5.6. For the ResNet18 and ResNet20, the highlights differ for each occurrence. Similarities in highlighting, however, are present for the character ”y” as shown in Figure 5.5 and the character sequence ”he” for the ResNet18 as shown in Figure 5.8, where the connection between the characters is highlighted in combination with the right part of the arch of ”h”. However, for the ResNet20, the highlighting for the same character sequence differs for the occurrences on the page.

Examples for the saliency maps generated for the WV networks for the first and second



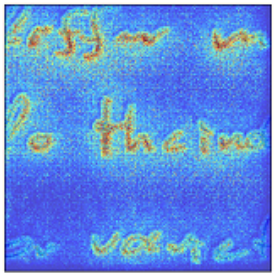
(a) ResNet18



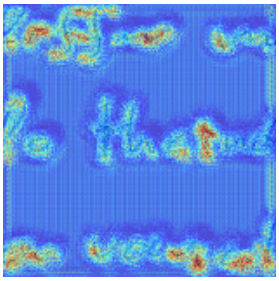
(b) ResNet20

Figure 5.8: Occurrences of the character sequence ”he” in the saliency maps generated for the WI networks for the second page.

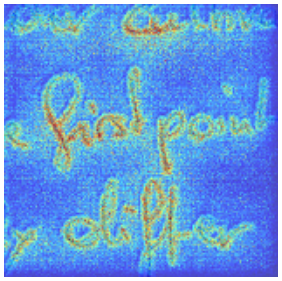
page are shown in Figure 5.9. For both WV networks, the saliency maps display scattered



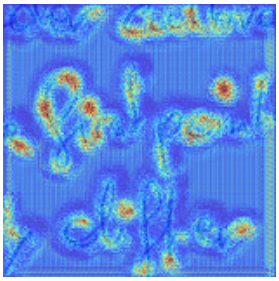
(a) ResNet18 First Page



(b) ResNet20 First Page



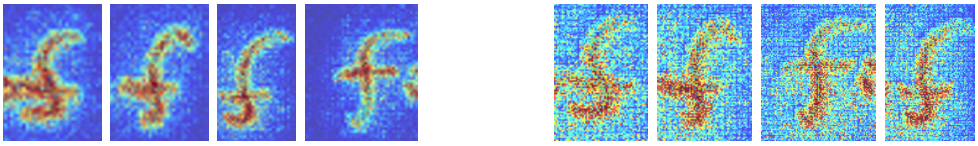
(c) ResNet18 Second Page



(d) ResNet20 Second Page

Figure 5.9: Pixel-wise saliency maps for the handwritten text of the first and second page on the WV ResNet18 and ResNet20 neural networks.

highlights, with the saliency maps for the ResNet18 displaying distributed intensities between the characters of the handwritten text. Additionally, the highlighting patterns for characters differ between the networks trained on WI and the networks trained on WV. For example, the WI ResNet18 network highlights the crossing of the lines of the character ”f” with high significance for the first page, while the WV ResNet18 networks places high significance on the bottom of the vertical line. Examples of occurrences are shown in Figure 5.10.



(a) WI ResNet18 (b) WV ResNet18

Figure 5.10: Occurrences of the character ”f” in the saliency maps generated for the WI and WV ResNet18 networks for the first page.

(a) WI ResNet20 (b) WV ResNet20

Figure 5.11: Occurrences of the character ”i” in the saliency maps generated for the WI and WV ResNet20 networks for the second page.

The saliency maps generated for the second page for the WV ResNet20 network differ from the saliency maps generated for the WI ResNet20 network. For example, the WV network places a high intensity highlight on the dot of the character ”i”. This, however, is not present for the WI network, where the dot of the character is displayed with the lowest significance in the snippet. Examples are shown in Figure 5.11.

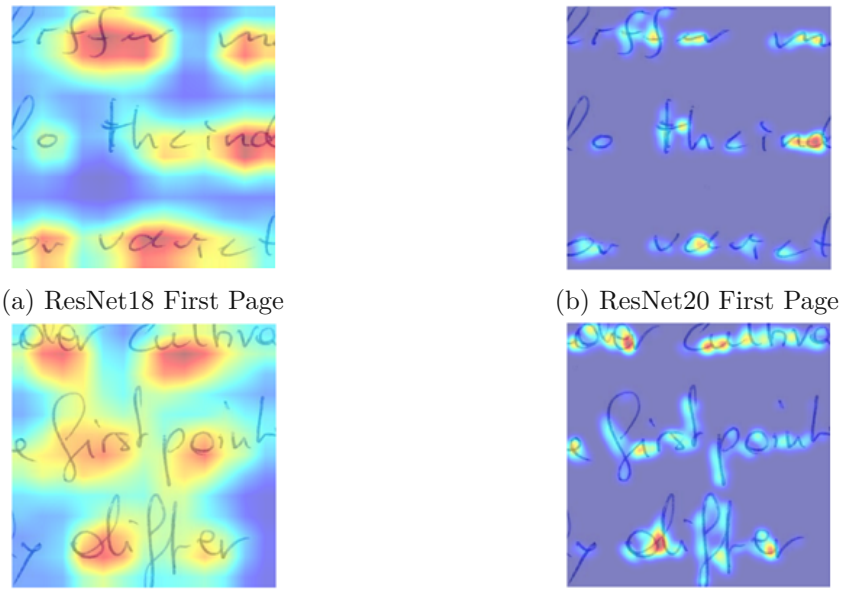
Overall, the saliency maps display similarities in the highlighting of characters for the saliency maps of one network, but deviating highlightings for saliency maps generated from different networks. This indicates that the networks take different areas of the characters into account when identifying the author of the handwritten text. A comparison of highlights between the saliency maps of Figure 5.3a and Figure 5.3b shows that the maps for the snippets of Figure 5.3a contain more single areas with high intensity in comparison to the snippets in Figure 5.3b, where multiple areas and characters are displayed with high intensity. This indicates that the snippets of the second page do not have salient characteristics, which the neural networks can used to identify the author. Instead, all the available information is used to determine the output of the neural network. Moreover, the saliency maps for the first page contain more consistent highlighting patterns for the characters, while the character highlightings vary more in saliency maps for the second page.

Comparing the generated saliency maps of the two networks shows that overlapping areas with high intensity occur for the networks especially in snippets with few characters. However, for snippets with multiple characters, i.e. where the snippets contain a word with more than five characters, the highlightings differ from network to network. This indicates that the networks focus on different aspects of the handwritten text if multiple characters and, therefore, more information is present in the snippet.

The noisy highlights displayed by the saliency maps generated for the WV networks indicate an influence of the loss selected for the underlying neural network, since the WV are trained with the Contrastive Loss and the WI networks are trained with the Triplet Margin Loss.

Point-Specific Saliency Map

Examples for the point-specific saliency maps generated for the WI networks for the first and second page are shown in Figure 5.12. The saliency maps for the ResNet20



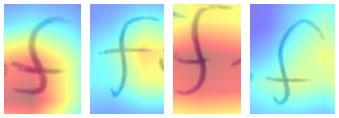
(a) ResNet18 First Page (b) ResNet20 First Page

(c) ResNet18 Second Page (d) ResNet20 Second Page

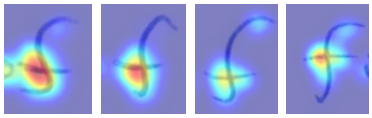
Figure 5.12: Point-specific saliency maps for the handwritten text of the first and second page on the WI ResNet18 and ResNet20 neural networks.

display a similar highlighting pattern for the character ”f” for the first page. The peak highlighting for this character is placed at the centre of the character, where the vertical and horizontal lines cross. However, for the second page, no highlighting pattern emerges. Examples for the highlights are shown in Figure 5.13. The saliency maps for the ResNet20 network display a similar highlighting pattern for the character ”d” as well, where the circular area is displayed with a peak highlight for the first page. For the second page, a similar highlighting pattern emerges, where the area between the circle and the vertical line is highlighted. This, however, is not the case for the saliency maps for the ResNet18 network, where no highlighting pattern emerges for this character. Examples for the occurrences in the saliency maps are shown in Figure 5.14.

Examples for the point-specific saliency maps generated for the WV networks for the

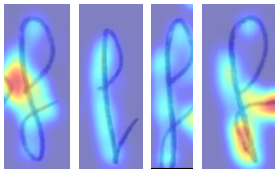


(a) ResNet18 First Page



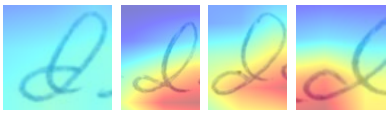
(b) ResNet20 First Page

(c) ResNet18 Second Page

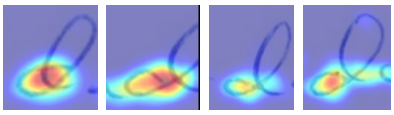


(d) ResNet20 Second Page

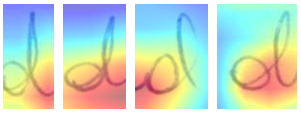
Figure 5.13: Occurrences of the character ”f” in the saliency maps generated for the WI networks for the first and second page.



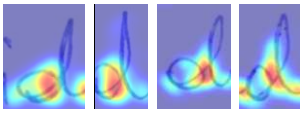
(a) ResNet18 First Page



(b) ResNet20 First Page



(c) ResNet18 Second Page

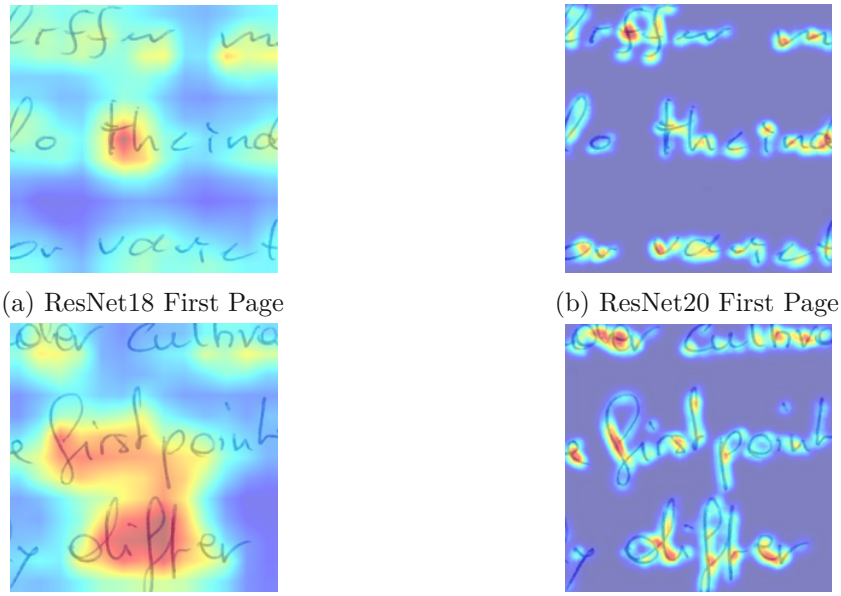


(d) ResNet20 Second Page

Figure 5.14: Occurrences of the character ”d” in the saliency maps generated for the WI networks for the first and second page.

first and second page are shown in Figure 5.15. Both networks place peak highlights at different locations than the WI networks. For example, the WI ResNet18 network places peak highlights for the first page on the character sequences ”ffer”, ”nd” and ”ar”, which is shown in Figure 5.12a, while the WV ResNet18 network places one peak highlight on the character sequence ”th” as shown in Figure 5.15a. The saliency maps for the first page for the ResNet18 display multiple occurrences of the character ”o” with the same highlighting pattern, where a peak highlight is placed at the bottom left of the character. However, this pattern is not displayed for all occurrences of this character. The ResNet20 network also displays a highlighting pattern for the same character, where one end of the line, which constructs the character, is highlighted. Examples are shown in Figure 5.16.

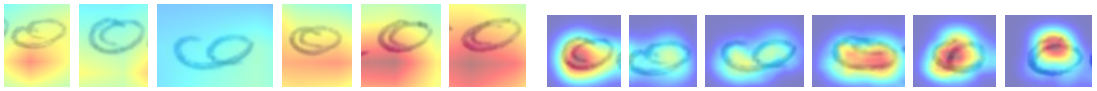
Overall, the point-specific saliency maps for the ResNet20 network display similarities in highlightings for the same character, however, the saliency maps for the ResNet18 network contain highlighting patterns with large deviations between the occurrences. Additionally, the exact position of a highlight is difficult to determine for this network, as it covers large areas of the page and, in some cases, overlaps multiple characters. This aggravates the allocation of a highlight to a certain character. In contrast, the ResNet20



(a) ResNet18 First Page (b) ResNet20 First Page

(c) ResNet18 Second Page (d) ResNet20 Second Page

Figure 5.15: Point-specific saliency maps for the handwritten text of the first and second page for the WV ResNet18 and ResNet20 neural networks.



(a) ResNet18 (b) ResNet20

Figure 5.16: Occurrences of the character ”o” in the saliency maps generated for the WV networks for the first page.

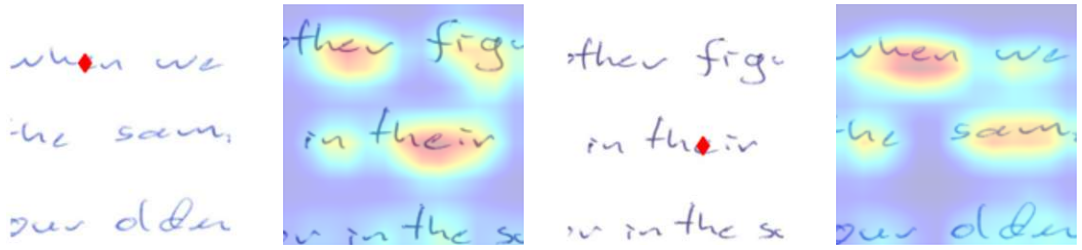
network displays concise highlighting areas, which are located closely to handwritten text and can thus be assigned to a character. This shows the influence the feature map size of the last convolutional layer has on the accuracy of the generated highlighting regarding its allocation, as the feature maps for the ResNet20 network are significantly larger than the feature maps for the ResNet18 network. In multiple cases for the ResNet18 network, the peak highlight is not placed on top of a character but in the vicinity of them, such as below the character. This indicates that the area around the characters is also analyzed by the neural network.

Point-to-Image Comparison

The point-specific saliency map technique, as described in Section 3.2.2, provides a second type of saliency map, where a point in an image can be selected for an analysis of its similarity to a second image. For this section, the analysis is conducted between a point in an image snippet and a second snippet taken from another page written

by the same author as the first snippet. A ResNet18 model is used as the underlying neural network. One snippet is selected from each page as shown in Figures 5.3a and 5.3b.

The saliency maps generated for the snippets taken from two pages written by the first author are shown in Figure 5.17. For the first snippet, a point on the character ”e”



(a) (b) (c) (d)

Figure 5.17: Point-specific saliency maps for the first author with a point selected within one snippet and a comparison to the overall other image. The selected points are displayed as red diamonds.

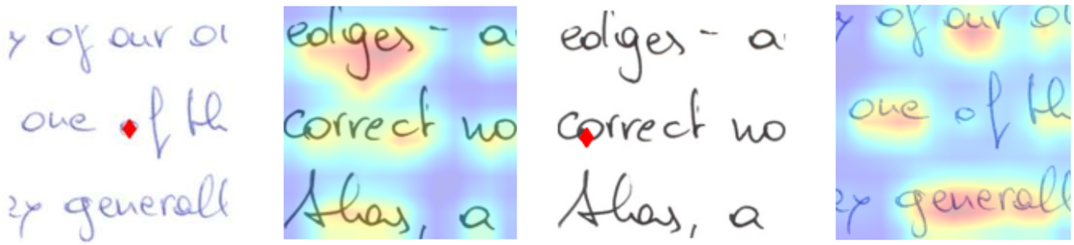
has been selected. A peak highlight is correctly placed on an occurrence of the same character within the character sequence ”their”. The highlight with the second-highest significance is also correctly placed on another occurrence of this character in the sequence ”ther”. However, the third-highest significance is placed on the character sequence ”fig”, where the character does not occur. Additionally, another occurrence of the character in the character sequence ”the” is highlighted with weak intensity. For the second snippet, the point has been placed on the character ”e” as well. Here, a peak highlight is positioned below an occurrence of the same character in the character sequence ”when”. The second and third-highest significance values, however, are assigned to the characters ”a” and ”m” within the character sequence ”sam”. It is noteworthy, however, that the right part of the character ”a”, resembles the characteristics of the character ”e”. This similarity is shown in Figure 5.18. Three other occurrences of the character ”e” in the sequences

(a) (b)

Figure 5.18: Character parts taken from the page shown in Figure 5.3a. The left snippet displays the right part of the character ”a” while the right snippet displays an occurrence of the character ”e”.

”we”, ”he” and ”older” are shown with a weak-intensity highlight.

The saliency maps generated for the snippets taken from two pages written by the second author are shown in Figure 5.19. For the first snippet, a point at the bottom



(a) (b) (c) (d)

Figure 5.19: Point-specific saliency maps for the second author with a point selected within one snippet and a comparison to the overall other image. The selected points are displayed as red diamonds.

of the character ”o” has been selected. The second snippet contains two occurrences of this character, however, they are not highlighted with a peak highlight. Instead, the character sequence ”edges” is highlighted strongly with a peak below the second occurrence of the character ”e”. Additionally, strong highlights are placed in the middle of the character sequence ”Alas” and below the second occurrence of the character ”c” in the character sequence ”correct”. Both characters, ”e” as well as ”c”, contain a round bottom part, similar to the character ”o”. However, a similar characteristic is present for the character ”a”, which is minimally highlighted in the saliency map. For the second snippet, a point at the bottom of the character ”o” has been selected as well. Here, the saliency map displays a peak highlight below the character sequence ”en”, which is part of the sequence ”generall”. Another peak highlight is placed below the character ”u” in the sequence ”our”. The third strongest highlight is displayed below the character ”e” in the sequence ”one”. For all three cases, the highlighted characters display a round bottom part, similar to the selected character ”o”. However, the five actual occurrences of the character ”o” are not displayed with peak highlights, with two being shown with a medium-intensity highlight. In comparison to the saliency maps for the first page, the highlights for the saliency maps of the second page are more distributed and overlap multiple characters, which aggravates the allocation of the highlights to a certain character.

The saliency maps display characteristics that indicate a comparison of patterns in characters executed by the underlying neural network. These patterns are not restricted to certain characters, as has been shown for the characters ”e” and ”o”, where other characters with similar features are highlighted strongly as well. However, the actual occurrences of the same character are not reliably highlighted, with multiple occurrences being weakly highlighted. This shows that the point-specific saliency map does display similar patterns and features but does not select all occurrences of characters with the same features.

5.2.4 Discussion

The pixel-wise saliency maps frequently display similar highlightings for multiple occurrences of the same character. The point-specific saliency maps, however, do not display similar highlightings in a reliable frequency. Additionally, the position of a peak highlight moves around in the vicinity of a character for this type of saliency map. The pixel-wise saliency maps are able to display more precise values regarding the significance of an area in an image, since the values are calculated for each pixel individually. In contrast, the point-specific saliency maps are calculated based on the feature maps of the last convolutional layer and are upscaled to the size of the input image, with interpolation used to create values for each individual pixel. This results in inaccuracies regarding the exact position of a highlight and the values for the areas between the originally calculated values. As a consequence, the allocation of a highlight to a certain character or character part is possible for the pixel-wise saliency maps but is aggravated for the point-specific saliency maps. This is also supported by the quantitative evaluation results, which show that the pixel-level saliency maps perform better than the point-specific saliency maps for the deletion score. The difference in performance indicates that the pixel-wise saliency maps provide more accurately allocated highlightings than the point-specific saliency maps. Moreover, due to the positioning of the highlights next to the characters for the point-specific saliency maps, the deletion and insertion scores cannot use the overall information of the highlight as it is not fully placed on black pixels. This indicates one reason why the deletion score is worse for the point-specific saliency maps than for the pixel-level saliency maps, where the highlights are allocated on the handwritten text itself. The upscaling of the calculated values to create a point-specific saliency map indicates that this transparency technique performs well when only little information in concise locations is present. This suggests that the use of the point-specific saliency maps with character-based input images, i.e. images containing only one handwritten character, could improve the level of detail the point-specific saliency maps can provide.

A disadvantage of both types of maps is the normalization of the calculated values for the creation of the visualization. This results in incomparability of peak highlights between saliency maps created by the same technique, since the actual value of significance might be different. Instead, the intensity of the highlights can be compared within one snippet only.

The qualitative evaluation results indicate that the neural networks analyse occurrences of the same character, similar to a human investigator. Additionally, the neural networks consider few characters with high significance if multiple characters and, therefore, more information is available, indicating a focus on salient characteristics. In contrast, if few characters are available, more characters and areas in the image are considered for the creation of an embedding output.

As described in Section 5.2.2, the quantitative evaluation results suggest a robustness of the underlying neural networks to the amount of information available. For both types

of neural networks, WI and WV, the insertion score for the point-specific saliency maps is higher or equal and the deletion score lower or equal in comparison to the pixel-wise saliency maps. An exception for the deletion score is present for the WI ResNet20 model on the Firemaker dataset and the WV ResNet20 model on the CVL and ICDAR2013 datasets. For the insertion score, an exception is present for the ResNet50 WV model on the CVL dataset. This indicates that the underlying neural network is able to correctly classify an input image if batches of the handwritten text are removed or inserted, while problems occur when the handwritten text is altered by a few pixels in comparison to the original handwritten text.

Overall, the similar highlightings for different occurrences of the same character for the pixel-wise saliency maps and the similar processing of the input data to a human investigator shows that this transparency technique can support the analysis of a handwritten text by an investigator by highlighting similar patterns in characters. This transparency technique suggests locations containing interesting information for the identification process and, therefore, facilitates the analysis process. Further, due to the calculation of significance on a pixel-based level, this transparency technique is suitable for neural networks, which take handwritten text as input and contain significant information at multiple locations in the input image. However, the point-specific saliency maps display non-intuitive highlightings regarding the allocation of a highlight to a certain character and are not suitable for the support of the analysis process. The characteristics of the generated saliency maps indicate that the use of these maps for neural networks, which take images containing compact information such as natural images, is more suitable than for neural networks, which take handwritten text images as input.

CHAPTER 6

Conclusion

6.1 Summary

In this thesis, the applicability of transparency techniques on neural networks trained on WI and WV is evaluated. DNNs are used in many areas in the computer vision domain, where they have become the state of the art for such tasks [SBM+17, ZTLT21]. This includes the area of WI and WV, where the DNNs are used to identify the author of a handwritten text. This is, for example, used in forensic evaluations, where handwritten texts are compared for similarities [KFS18]. The topic of machine learning interpretability has become a focus topic in the past five years [SBM+17]. The proposed transparency techniques, which provide insight into the decision process of a neural network, are used to remove bias from the training data, detect artefacts present in the training process and to increase the trust and reliability of such systems in safety-critical areas [SM19, ZTLT21].

This thesis is a first step into the topic of transparency for neural networks trained on WI and WV. The goal is to gain insight into the decision process of the underlying neural network and support investigators with a visualization, which provides information on similarities in the given handwritten text. For this purpose, two transparency techniques, namely the pixel-wise saliency maps proposed by Kobs et al. [KSDH21] and the point-specific saliency maps proposed by Zhu et al. [ZYC21], are selected from the state of the art for embedding neural networks. Additionally, three neural network architectures, namely ResNet18, ResNet20 and ResNet50, are selected based on the state of the art for WI and WV. The neural networks are trained and evaluated using the CVL, Firemaker and ICDAR2013 datasets. The transparency techniques are quantitatively evaluated using an adjusted version of the deletion and insertion score as proposed by Hu et al [HVH22]. For the qualitative evaluation, the saliency maps calculated for two pages taken from the CVL dataset are compared for similarities in highlightings for multiple occurrences of the same characters.

The results of the evaluation show large differences in the performance of the deletion and insertion score. The deletion scores are comparably lower than the insertion scores. Moreover, the deletion scores for the pixel-wise saliency maps are overall better in comparison to the deletion scores of the point-specific maps. This suggests a robustness of the underlying neural network to the amount of available information but indicates problems occurring in the neural network when the available handwriting information is altered by removing pixels. Moreover, the qualitative evaluation shows that the point-specific saliency maps contain highlights which are vague regarding the allocation to a certain character, while the allocation is simpler for the pixel-wise saliency maps. Similarities in highlightings for the same character at different occurrences are present for the pixel-wise saliency maps, while the highlights show great variations for the point-specific saliency maps. Overall, the pixel-wise saliency maps are suitable for the support of forensic experts, while the point-specific saliency maps display non-intuitive highlightings regarding the location of a highlight and are therefore not suitable.

In Section 1.1, three research questions are defined. The first question is defined as *"What characteristics are selected by a neural network to identify the author of a handwritten text?"*. The evaluation results described in Section 5.2 have shown that the trained neural networks select parts of characters or whole characters from the given handwritten text, depending on the amount of information available in the input snippet and the characteristics of the given characters. Additionally, areas such as connections between successive characters are considered if they contain significant information. Moreover, the sanity checks in Section 5.2.1 and the qualitative evaluation of the point-specific saliency maps show that neural networks also consider the area surrounding the characters, such as contours and areas inbetween characters. The second question is defined as *"How do the visualizations of feature contribution differ from text areas, which experts consider when identifying the author of a handwritten text?"*. The results of the qualitative evaluation as described in Section 5.2.3 show that similar highlights are present in the pixel-wise saliency maps for different occurrences of the same character. As Christlein [Chr19] notes, the identification process of a forensic expert can include the comparison of characteristics of the same characters at different positions in a given handwritten text. Comparing this procedure to the highlightings created by the pixel-wise saliency maps shows a similarity in the approach to the identification of an author, where the underlying neural network considers similar areas and features of a character to identify the author. Finally, the third question is defined as *"How well does a transparency technique perform on neural networks, which take handwritten text images as input?"*. The quantitative evaluation shows that both of the selected transparency techniques display issues, as the deletion scores are generally lower than the insertion scores. Especially for the point-specific saliency maps, the deletion scores are considerably low, with none of the WI and WV neural networks achieving a score above 35%. This indicates that the transparency techniques have performance problems with neural networks, which take handwritten text images as input.

6.2 Future Work

This thesis provides a first step towards the explainability of WI and WV neural networks. Therefore, the proposed approach and evaluation can be further enhanced to receive more insight into this topic.

The quantitative evaluation uses the deletion and insertion of black pixels in the input images for the neural networks. This adjustment was made due to the binarization of the input images and, therefore, the non-existence of grey values, which were originally proposed by Hu et al. [HVH22] for this evaluation metric. However, the restriction of the deletion onto black pixels in the input image restricts the use of all information the saliency maps provide. This becomes visible in the sanity checks described in Section 5.2.1. Here, the saliency maps produced by pretrained neural networks use the information contained on the contour of the handwritten text, which includes underlying white pixels. This information, however, is not used during the deletion and insertion score as it lies outside the boundaries of the handwritten characters. In future work, a quantitative evaluation method suitable for cases, where white and black pixels should be considered, could be explored.

Further work could include the evaluation of changes in the distance between embeddings of the snippets of one page and how the removal of information from the according image snippets influences this distance. This would provide a more detailed insight into the importance of certain areas in the image and the behaviour of different types of snippets regarding the amount of information they contain. This would additionally allow to determine the effect different characters and their characteristics have on the embedding output. Moreover, unique characteristics such as a crossed-out text and their impact on the embedding output could be explored.

The qualitative evaluation of the transparency techniques on WI neural networks displayed similar highlightings for characters in different image snippets. However, due to the normalization of the values to generate the saliency map, the intensity of the highlightings may vary, depending on other information available in the snippet and its significance for the determination of the authorship. Therefore, experiments could be conducted on the similarity of highlightings for characters for a character-based neural network, i.e. a network which takes input images containing a single character. Due to the absence of other information, the intensity of the highlighting can be taken into account for the evaluation in this case.

List of Figures

List of Tables

Acronyms

AUC area-under-the-curve. 8, 24, 48–50, 73

BNNeck Batch Normalization Neck. 23

CAM Class Activation Map. 13, 14

CNN Convolutional Neural Network. 13, 16–18, 20, 23, 26, 27, 31

DGMP Deep Generalized Max Pooling. 19

DNN Deep Neural Network. 1, 16, 65

EER Equal Error Rate. 15, 23, 24

FAR False Acceptance Rate. 15, 20

FC Fully Connected. 14, 16, 17, 20, 31, 32

FRR False Rejection Rate. 15, 20

GAP Global Average Pooling. 20, 27, 31, 32

GDPR General Data Protection Regulation. 4

GMM Gaussian Mixture Model. 16, 17

GMP Generalized Max-Pooling. 20

Grad-CAM Gradient-weighted Class Activation Mapping. 2, 11, 13

HTD Handwriting thickness descriptor. 19

ILNNeck Interference Layer Normalization Neck. 23

IN Instance Normalization. 23

IoU Intersection over Union. 13

LN Layer normalization. 23

mAP Mean Average Precision. 14–20, 27, 28, 43–46, 48, 49, 73

PCA Principal Component Analysis. 17, 20

ReLU Rectified Linear Unit. 16–20

RNN Recurrent Neural Network. 22

SGD Stochastic Gradient Descent. 18, 78

SIFT Scale-invariant Feature Transform. 17, 18

SOP Stanford Online Products. 12, 14, 29, 69

VLAD Vector of Locally Aggregated Descriptors. 17, 18, 20

WI Writer Identification. 1, 4, 5, 7, 14, 17, 19, 25–29, 39, 43, 45–58, 63, 65–67, 70, 73, 78

WV Writer Verification. 1, 4, 5, 7, 14, 15, 25–29, 39, 43, 45, 46, 48–51, 54–59, 63, 65–67, 70, 73, 78

Appendix

ResNet Architecture

In order to verify the implementation of the chosen ResNet models and to identify possible problems in the implementation, a comparison of the chosen implementation with existing implementations is conducted. For this purpose, the ResNet50 neural network is compared with a ResNet50 neural network provided by Phan [Pha21]. Their architecture and training process achieves a test accuracy of 93.65% on the Cifar-10 dataset [Pha21]. For this purpose, the performances of the two implementations are compared by adjusting different parameters of the training process and observing the changes in the achieved accuracies. The results are shown in Table 1. The split into training and test set provided by the authors of the dataset is used for all evaluations. The weights of the models are not initialized and the models are trained for 100 Epochs.

The *Architecture* column displays which ResNet50 architecture is used. "Base" refers to the PyTorch implementation of the ResNet50 architecture, while "Phan" refers to the implementation provided by Phan [Pha21]. The *Output* column refers to the dimension of the output of the network. The *Transformation* column notes if transformations have been applied to the input images during training. The transformations consist of

Architecture	Output	Transform.	Scheduler	Eval.	Loss	Acc.
Base	1×1000	No	No	Class	Cross-Entropy	76.84%
Base	1×1000	Yes	No	Class	Cross-Entropy	72.17%
Base	1×10	Yes	No	Class	Cross-Entropy	77.37%
Base	1×10	Yes	Yes	Class	Cross-Entropy	89.23%
Base	1×1000	Yes	Yes	Class	Cross-Entropy	89.56%
Base	1×1000	Yes	Yes	KNN	Triplet-Margin	86.49%
Base	1×1000	Yes	No	KNN	Triplet-Margin	74.59%
Phan	1×1000	Yes	Yes	KNN	Triplet-Margin	91.01%
Phan	1×1000	Yes	No	KNN	Triplet-Margin	80.20%

Table 1: Models and parameters chosen for the evaluation of the ResNet50 performance on the Cifar-10 dataset.

```
SGD:
    dampening: 0
    lr: 0.01
    maximize: False
    momentum: 0.9
    nesterov: True
    weight_decay: 0.01
```

Listing 1: Parameters used for the SGD optimizer.

random cropping with padding set to 4, random horizontal flip and normalization by mean and standard deviation for the training data. These transformations are selected based on the implementation provided by Phan [Pha21], where the same configuration for the transformations is used. For the test data, only normalization is applied. The *Scheduler* column displays if a scheduler has been used for the learning rate of the training. The scheduler implementation of the Linear Warmup Cosine Annealing by Phan [Pha21] is used for all runs with scheduler. Two types of evaluations are displayed in the *Evaluation* column. The first evaluation, the class evaluation, compares the output of the neural network with the true class label. In order to calculate the accuracy, the percentage of images labelled correctly by the network is taken. For this evaluation, the Cross-Entropy Loss is used, which is a commonly used loss function for classification tasks [GRLGPC20]. The second evaluation uses a KNN algorithm. The images of the test dataset are forwarded to the network, which provides an embedding vector for each image. For each query image in the test set, the three nearest neighbours are calculated and the most often occurring label of these neighbours is taken as the label for the query image. The test accuracy is again calculated as the percentage of correctly labelled images. For this evaluation, the Triplet Margin Loss with a margin of 0.1 is used. The SGD is used as optimizer. If not stated otherwise, the parameters for the optimizer are set as seen in Listing 1. For each evaluation, the test accuracy is given in the table.

The results show that the scheduler improves the achieved test accuracy. The accuracy of a neural network architecture can be improved by 12 percentage points when a scheduler is used. Therefore, the scheduler is selected for the training process of the WI and WV neural networks. Additionally, the architecture proposed by Phan performs better on the dataset than the PyTorch implementation, where an increase of five percentage points can be observed. However, in order to use an architecture comparable to the ResNet50 architectures used in the literature, the PyTorch implementation is further used in this thesis.

Bibliography

[ACB19] Chandranath Adak, Bidyut B. Chaudhuri, and Michael Blumenstein. An Empirical Study on Writer Identification and Verification From Intra-Variable Individual Handwriting. *IEEE Access*, 7:24738–24758, 2019.

[AOS22] Leila Arras, Ahmed Osman, and Wojciech Samek. CLEVR-XAI: a benchmark dataset for the ground truth evaluation of neural network explanations. *Information Fusion*, 81:14–40, 2022.

[AWN+22] Christopher J Anders, Leander Weber, David Neumann, Wojciech Samek, Klaus-Robert Müller, and Sebastian Lapuschkin. Finding and removing Clever Hans: using explanation methods to debug and improve deep models. *Information Fusion*, 77:261–295, 2022.

[BS07] Marius Bulacu and Lambert Schomaker. Text-Independent Writer Identification and Verification Using Textural and Allographic Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):701–717, 2007.

[BSV03] Marius Bulacu, Lambert Schomaker, and Louis Vuurpijl. Writer Identification Using Edge-Based Directional Features. In *ICDAR '03: Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 937–941. IEEE Computer Society, 2003.

[CBH+17] Vincent Christlein, David Bernecker, Florian Hönig, Andreas Maier, and Elli Angelopoulou. Writer Identification Using GMM Supervectors and Exemplar-SVMs. *Pattern Recognition*, 63:258–267, 2017.

[CBMA15] Vincent Christlein, David Bernecker, Andreas Maier, and Elli Angelopoulou. Offline Writer Identification Using Convolutional Neural Network Activation Features. In *German Conference on Pattern Recognition*, pages 540–552. Springer, 2015.

[CCHM20] Lei Chen, Jianhui Chen, Hossein Hajimirsadeghi, and Greg Mori. Adapting Grad-CAM for Embedding Networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2794–2803, 2020.

[CGFM17] Vincent Christlein, Martin Gropp, Stefan Fiel, and Andreas Maier. Unsupervised Feature Learning for Writer Identification and Writer Retrieval. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–997. IEEE, 2017.

[Chr19] Vincent Christlein. *Handwriting Analysis with Focus on Writer Identification and Writer Retrieval*. Ph.D. dissertation, Friedrich-Alexander-Universitaet Erlangen-Nuernberg (Germany), 2019.

[CLL+21] Xiaodong Chen, Xinchen Liu, Wu Liu, Xiao-Ping Zhang, Yongdong Zhang, and Tao Mei. Explainable Person Re-Identification With Attribute-Guided Metric Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11813–11822, 2021.

[CSS+19] Vincent Christlein, Lukas Spranger, Mathias Seuret, Anguelos Nicolaou, Pavel Král, and Andreas Maier. Deep Generalized Max Pooling. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1090–1096. IEEE, 2019.

[CZY+22] Xianmu Cairang, Duoji Zhaxi, Xiaolong Yang, Yan Hou, Qijun Zhao, Dingguo Gao, Danzeng Pubu, and Dorji Gesang. Learning Generalisable Representations for Offline Signature Verification. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2022.

[DDIT+17] Sounak Dey, Anjan Dutta, Juan Ignacio Toledo, Suman K. Ghosh, Josep Lladós, and Umapada Pal. SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification. *arXiv preprint arXiv:1707.02131*, 2017.

[DVK17] Finale Doshi-Velez and Been Kim. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv preprint arXiv:1702.08608*, 2017.

[EVGW+10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88:303–338, 2010.

[FS15] Stefan Fiel and Robert Sablatnig. Writer Identification and Retrieval Using a Convolutional Neural Network. In *Computer Analysis of Images and Patterns*, pages 26–37. Springer, 2015.

[GBY+18] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89. IEEE, 2018.

[GMR+18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[GRLGPC20] Elliott Gordon-Rodriguez, Gabriel Loaiza-Ganem, Geoff Pleiss, and John Patrick Cunningham. Uses and Abuses of the Cross-Entropy Loss: Case Studies in Modern Deep Learning. In *Proceedings on "I Can't Believe It's Not Better!" at NeurIPS Workshops*, pages 1–10. PMLR, 2020.

[HRBLM07] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[HS20] Sheng He and Lambert Schomaker. FragNet: Writer Identification Using Deep Fragment Networks. *IEEE Transactions on Information Forensics and Security*, 15:3013–3022, 2020.

[HSBC20] Tom Heskes, Evi Sijben, Ioan Gabriel Bucur, and Tom Claassen. Causal Shapley Values: Exploiting Causal Knowledge to Explain Individual Predictions of Complex Models. *Advances in Neural Information Processing Systems*, 33:4778–4789, 2020.

[HVH22] Brian Hu, Bhavan Vasu, and Anthony Hoogs. X-MIR: EXplainable Medical Image Retrieval. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 440–450, 2022.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[JJ20] Malihe Javidi and Mahdi Jampour. A deep learning framework for text-independent writer identification. *Engineering Applications of Artificial Intelligence*, 95:103912, 2020.

[KBC+22] Andrey Kuehlkamp, Aidan Boyd, Adam Czajka, Kevin Bowyer, Patrick Flynn, Dennis Chute, and Eric Benjamin. Interpretable Deep Learning-Based Forensic Iris Segmentation and Recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pages 359–368, 2022.

[KFDS13] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. CVL-Database: An Off-line Database for Writer Retrieval, Writer Identification and Word Spotting. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 560–564. IEEE, 2013.

[KFS18] Manuel Keglevic, Stefan Fiel, and Robert Sablatnig. Learning Features for Writer Retrieval and Identification using Triplet CNNs. In *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 211–216. IEEE, 2018.

[KH22] Konstantin Kobs and Andreas Hotho. On Background Bias in Deep Metric Learning. In *Fifteenth International Conference on Machine Vision (ICMV 2022)*, page 1270114. International Society for Optics and Photonics, 2022.

[KSDH21] Konstantin Kobs, Michael Steininger, Andrzej Dulny, and Andreas Hotho. Do Different Deep Metric Learning Losses Lead to Similar Learned Features? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10644–10654, 2021.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[LGSP13] Georgios Louloudis, Basilios Gatos, Nikolaos Stamatopoulos, and A Papandreou. ICDAR 2013 Competition on Writer Identification. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1397–1401. IEEE, 2013.

[LJ19] Songxuan Lai and Lianwen Jin. Offline Writer Identification Based on the Path Signature Feature. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1137–1142. IEEE, 2019.

[LPLL20] Mei Chee Leong, Dilip K Prasad, Yong Tsui Lee, and Feng Lin. Semi-CNN Architecture for Effective Spatio-Temporal Learning in Action Recognition. *Applied Sciences*, 10(2):557, 2020.

[LWB+19] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1–8, 2019.

[LZL+19] Chuang Li, Xing Zhang, Feng Lin, Zhiyong Wang, Jun'E Liu, Rui Zhang, and Haiqiang Wang. A Stroke-Based RNN for Writer-Independent Online Signature Verification. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 526–532, 2019.

[MLB+17] Gregoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Muller. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern recognition*, 65:211–222, 2017.

[NNI+19] Hung Tuan Nguyen, Cuong Tuan Nguyen, Takeya Ino, Bipin Indurkhya, and Masaki Nakagawa. Text-independent writer identification using convolutional neural network. *Pattern Recognition Letters*, 121:104–112, 2019.

[NRVZD20] Karthikeyan Natesan Ramamurthy, Bhanukiran Vinzamuri, Yunfeng Zhang, and Amit Dhurandhar. Model Agnostic Multilevel Explanations. *Advances in Neural Information Processing Systems*, 33:5968–5979, 2020.

[OSXJS16] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4004–4012, 2016.

[Ots79] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[PDS18] Vitali Petsiuk, Abir Das, and Kate Saenko. RISE: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[Pha21] Huy Phan. huyvnphan/pytorch_cifar10. `https://doi.org/10.5281/zenodo.4431043`, Last accessed 31 July 2023, January 2021.

[PP21] Naresh Purohit and Subhash Panwar. State-of-the-Art: Offline Writer Identification Methodologies. In *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–8. IEEE, 2021.

[QZZ+23] Zhen Qin, Pengbiao Zhao, Tianming Zhuang, Fuhu Deng, Yi Ding, and Dajiang Chen. A survey of identity recognition via data fusion and feature learning. *Information Fusion*, 91:694–712, 2023.

[RB22] Shervin Rasoulzadeh and Bagher BabaAli. Writer identification and writer retrieval based on NetVLAD with Re-ranking. *IET Biometrics*, 11(1):10–22, 2022.

[Rud94] Daniel L. Ruderman. The statistics of natural images. *Network: Computation in Neural Systems*, 5(4):517, 1994.

[SBM+17] Wojciech Samek, Alexander Binder, Grégoire Montavon, Sebastian Lapuschkin, and Klaus-Robert Müller. Evaluating the Visualization of What a Deep Neural Network Has Learned. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2660–2673, 2017.

[SCD+17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In

Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 618–626, 2017.

[Sch08] Lambert Schomaker. Writer Identification and Verification. In *Advances in Biometrics: Sensors, Algorithms and Systems*, pages 247–264. Springer, 2008.

[SDCS20] Mohammad Abuzar Shaikh, Tiehang Duan, Mihir Chauhan, and Sargur N. Srihari. Attention based Writer Independent Verification. In *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 373–379, 2020.

[SE10] Giovanni Seni and John Elder. *Ensemble Methods in Data Mining*. Morgan & Claypool Publishers, 2010.

[SM19] Wojciech Samek and Klaus-Robert Müller. Towards Explainable Artificial Intelligence. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 5–22, 2019.

[SMR08] Hinrich Schütze, Christopher D. Manning, and Prabhakar Raghavan. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[SPL+20] Shaeke Salman, Seyedeh Neelufar Payrovnaziri, Xiuwen Liu, Pablo Rengifo-Moreno, and Zhe He. DeepConsensus: Consensus-based Interpretable Deep Neural Networks with Application to Mortality Prediction. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[SSGS00] Dirk Scheuermann, Scarlet Schwiderski-Grosche, and Bruno Struif. *Usability of Biometrics in Relation to Electronic Signatures*. GMD-Forschungszentrum Informationstechnik Sankt Augustin, 2000.

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR, 2017.

[TW16] Youbao Tang and Xiangqian Wu. Text-Independent Writer Identification via CNN Features and Joint Bayesian. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 566–571. IEEE, 2016.

[WMC21] Zhenghua Wang, Andreas Maier, and Vincent Christlein. Towards End-to-End Deep Learning-based Writer Identification. *INFORMATIK 2020*, 2021.

[WMR17] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *Harv. J.L. & Tech.*, 31(2):841, 2017.

[WTB14] Xiangqian Wu, Youbao Tang, and Wei Bu. Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform. *IEEE Transactions on Information Forensics and Security*, 9(3):526–536, 2014.

[WZB19] Shengjie Wang, Tianyi Zhou, and Jeff Bilmes. Bias also matters: Bias attribution for deep neural network explanation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6659–6667. PMLR, 2019.

[XQ16] Linjie Xing and Yu Qiao. DeepWriter: A Multi-stream Deep CNN for Text-Independent Writer Identification. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 584–589. IEEE, 2016.

[ZKC+20] Meng Zheng, Srikrishna Karanam, Terrence Chen, Richard J Radke, and Ziyan Wu. Towards Visually Explaining Similarity Models. *arXiv preprint arXiv:2008.06035*, 2020.

[ZN20] Martin Zurowietz and Tim W. Nattkemper. An Interactive Visualization for Feature Localization in Deep Neural Networks. *Frontiers in Artificial Intelligence*, 3:49, 2020.

[ZTLT21] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A Survey on Neural Network Interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

[ZXD17] Hang Zhang, Jia Xue, and Kristin Dana. Deep TEN: Texture Encoding Network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 708–717, 2017.

[ZYC21] Sijie Zhu, Taojiannan Yang, and Chen Chen. Visual Explanation for Deep Metric Learning. *IEEE Transactions on Image Processing*, 30:7593–7607, 2021.

[ZZZL22] Borui Zhang, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Attributable Visual Similarity Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7532–7541, 2022.