BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

EXPLAINABLE FACE LIVENESS CLASSIFICATION

VYSVĚTLITELNOST KLASIFIKACE ŽIVOSTI TVÁŘÍ

MASTER'S THESIS
DIPLOMOVÁ PRÁCE

AUTHOR **Bc. PETR MIČULEK**
AUTOR PRÁCE

SUPERVISOR **Ing. JAKUB ŠPAŇHEL**
VEDOUCÍ PRÁCE

BRNO 2023

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Master's Thesis Assignment

144772

Institut:	Department of Computer Graphics and Multimedia (UPGM)
Student:	**Mičulek Petr, Bc.**
Programme:	Information Technology and Artificial Intelligence
Specialization:	Sound, Speech and Natural Language Processing
Title:	**Explainable Face Liveness Classification**
Category:	Image Processing
Academic year:	2022/23

Assignment:

1. Study the basics of image processing. Focus mainly on neural networks.
2. Study the available materials on neural networks for face liveness classification.
3. Explore current methods for explaining neural networks.
4. Select appropriate methods and propose a way to compare their applicability for face liveness classification.
5. Experiment with your implementation and, if necessary, propose your own modifications to the classification model to improve its explainability.
6. Compare your results and discuss possibilities for future developments.
7. Create a brief poster and video presenting your work, its goals, and results.

Literature:

- YIN, Bangjie, et al. Towards interpretable face recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019. p. 9348-9357.
- LIN, Yu-Sheng, et al. xCos: An Explainable Cosine Metric for Face Verification Task. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2021, 17.3s: 1-16.
- SEQUEIRA, Ana F., et al. An exploratory study of interpretability for face presentation attack detection. *IET Biometrics*, 2021, 10.4: 441-455.
- KRISHNA, Satyapriya, et al. The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective. *arXiv preprint arXiv:2202.01602*, 2022.
- Dále dle pokynů vedoucího

Requirements for the semestral defence:

- Completion of the first three points of the assignment
- Considerable work on the fourth point of the assignment

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor:	**Špaňhel Jakub, Ing.**
Consultant:	Beszédeš Marián, Ph.D. Ing.
Head of Department:	Černocký Jan, prof. Dr. Ing.
Beginning of work:	1.11.2022
Submission deadline:	31.7.2023
Approval date:	31.10.2022

Abstract

The goal of this thesis is to explore, develop, and evaluate explainable face presentation attack detection (PAD) systems. PAD systems act as security filters for face recognition, preventing spoofed faces from reaching the identification phase. These systems are a necessary component enabling the recent rise of biometric systems used in smartphones and security cameras. While neural networks are the standard method for this task, they are commonly a black-box method providing no explanation. To provide a better understanding of the detection process, input attribution methods are applied. Their suitability is studied and various variants are compared. Of the seven methods compared, GradCAM using test-time augmentation is evaluated as the best, achieving a deletion metric AUC of 0.658 and an insertion metric AUC of 0.908. Experiments with the explanations show their limited capability at helping understand the model, but provide hints at how the predictive accuracy of the PAD system can be verified, and possibly improved.

Abstrakt

Cílem této práce je analýza, vývoj a vyhodnocení vysvětlitelných systémů klasifikace živosti obličeje. Klasifikace živosti obličeje slouží jako bezpečnostní filtr vyřazující podvrhy před provedením rozpoznávání obličeje. Klasifikátory živosti obličeje proto nezbytně stojí v pozadí vzestupu biometrických systémů v chytrých telefonech a bezpečnostních kamerách. Standardní neuronové sítě pro tuto úlohu však neposkytují vysvětlení pro svá rozhodnutí. V této práci jsou tedy prozkoumány a použity metody vysvětlitelnosti, konkrétně atribuce vstupních dat. V rámci těchto metod jsou provedeny experimenty na zhodnocení jejich vhodnosti a jako nejvhodnější je vybrán GradCAM používající dodatečné augmentace. Dosahuje hodnoty AUC metriky při umazávání snímku o hodnotě 0,658 a AUC při vkládání do snímku 0,908. Ačkoli další experimenty ukazují limity věrnosti těchto vysvětlení a jejich ne vždy jasnou srozumitelnost, je ukázáno, že poskytují vodítka pro vylepšení prediktivních schopností systémů pro vyhodnocení živosti obličejů.

Keywords

Machine Learning (ML), Interpretable ML, ML Explainability, Convolutional Neural Networks, Face Liveness, Face Presentation Attack Detection

Klíčová slova

Strojové učení, interpretovatelné strojové učení, vysvětlitelnost strojového učení, klasifikace živosti tváří, konvoluční neuronové sítě

Reference

MIČULEK, Petr. *Explainable Face Liveness Classification.* Brno, 2023. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Ing. Jakub Špaňhel

Explainable Face Liveness Classification

Declaration

I hereby declare that this Master's thesis was prepared as an original work by the author under the supervision of Ing. Jakub Špaňhel. Supplementary information was provided by Ing. Marián Beszédeš Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....
Petr Mičulek
July 31, 2023

Acknowledgements

I would like to thank my supervisor, Ing. Jakub Špaňhel, for his understanding, advice, and patience. I would also like to extend my thanks to the consultant, Ing. Marián Beszédeš Ph.D.; I would like to thank Tibor Kubík for his perseverance in helping me, and for his feedback on the thesis text. Finally, my deepest gratitude goes to my family, friends, girlfriend, and everyone else who supported me throughout my studies. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic.

Contents

List of Figures

Chapter 1

Introduction

Biometry and machine learning have become ubiquitous in everyday life over the past decade. For many, face recognition has become the standard way to unlock their smartphones. In addition, face recognition is being increasingly used for entering buildings, at airports, and in many other day-to-day situations. However, the security of these systems is often questioned due to the possible abuse of spoofed faces, such as paper prints or on-screen video. To detect these spoofs, Presentation Attack Detection (PAD) systems are used as a first layer of security.

With recent advances in machine learning, neural networks are the standard approach for almost any problem in the field of computer vision. They have achieved superior performance in many tasks and fields, and are transforming the world as we know it. However, their high complexity makes them difficult to interpret.

Explainable artificial intelligence (XAI) attempts to interpret neural networks' behaviour and make them more reliable and trustworthy in high-stake decisions. Recently, even European legislation has been making efforts to enact the „right to explanation“, so explainability presents a crucial goal of the research field.

This thesis aims to investigate the methods of explainable AI and apply them to the task of face presentation attack detection. Applicable methods will then be employed to better understand the used models and data, and thus will aid in the development of the PAD solution.

The following chapters first present the fields investigated and then the proposed solution is presented in the remaining chapters. Chapter 2 gives a brief overview of face presentation attack detection, relevant methods, and available data. In Chapter 3, explainable machine learning is presented. In Chapters 4 and 5, the chosen explainability methods are applied to the PAD task, and their evaluation is discussed in Chapter 7.

Chapter 2

Face Liveness

Face liveness classification, also known as face presentation attack detection (PAD), or face anti-spoofing, is the task of verifying the authenticity of face biometric data. A photo or video of a human face is classified as either genuine (bona-fide, negative) or an attack (spoof, positive). Face liveness detection aims to prevent malicious users from attacking biometrics systems through identity theft or fake identity generation. Whereas face verification works with existing enrolled identities, face liveness is commonly used in scenarios where the person's identity is not yet registered in the system. Face liveness detection can be performed actively or passively. In active PAD, the user must perform a given action, such as moving their head or smiling. On the other hand, passive liveness is determined using only a single photo, which requires no user collaboration and can be performed instantly. While active liveness detection provides more input data, the requirements may be too cumbersome for users. The input data can take various modalities spanning RGB, depth, IR, LIDAR, audio, and other sensor data (e.g. rPPG heart rate measurement). Here, focus is put only on using RGB data from commonly used consumer device cameras.

2.1 Face Presentation Attack Categories

Various categories of spoofing attacks can be used to evade PAD systems. These include printed photos, replayed videos, make-up attacks, and others (see the list below). This work focuses on physical attacks, so faking the camera capture directly is not considered. Attacks, which digitally modify or generate faces, such as deepfakes, do not form a separate category, as their presentation will still use any of the existing categories. The list of categories is naturally never complete, as novel attacks can appear anytime.

Face Presentation Attacks List

- Printed photos – still photos, flat and bent photos, and photos using various printing materials
- Cutouts – paper photos or other masks partially covering the face
- Replay – photos shown on various displays
- Masks – from simple paper 2D masks to expensive 3D silicone masks, mannequins
- Makeup attacks – altering one's appearance to impersonate someone, or to obfuscate one's visage

2.2 Datasets

There are many existing datasets for the face PAD task. Due to the sensitive nature of sharing personal face data, their availability is often limited. Here, only datasets focusing on image (RGB) data are presented.

2.2.1 RoseYoutu

RoseYoutu [23] (2018) is a public academic dataset of spoofed faces in static video shots in an indoor setting. There are 3497 videos with 20 subjects. The videos are captured using common consumer smartphones released around 2013 (e.g. iPhone 5s). The standoff distance is 30 – 50 cm, and the subject is shown at a frontal angle. The main attack types are various forms of paper photos and video replays. Paper spoofs are cut out from A4 papers. See Figure 2.1 for the full list of attack categories and example images.

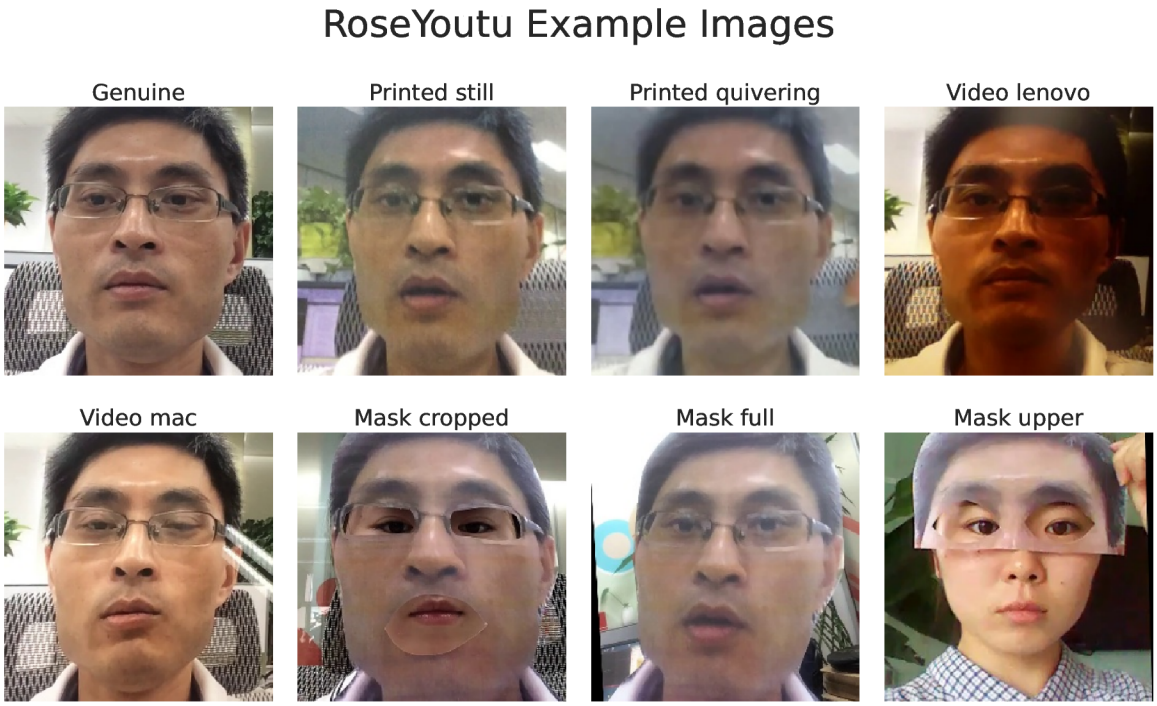


Figure 2.1: Example images showing every class in the RoseYoutu dataset [23].

2.2.2 SiW-M

The SiW-M dataset (Spoofing in the Wild with Multiple Attacks, 2018) [25] is a dataset of 1628 videos of 493 subjects, spanning 13 types of spoof attacks. The attack categories can be grouped into print, replay, 3D mask, makeup, and partial. Static and dynamic camera shots show the subjects at a close distance. Some data variability includes changes in head pose up to 90° and also changes in expression and illumination.

Although the SiW-M dataset is public, it is no longer available from the dataset authors (as of July 2023). A new version of the dataset, SiW-Mv2 [50], was released in 2022, which drops the makeup attacks, as these were taken from YouTube videos in the first version.

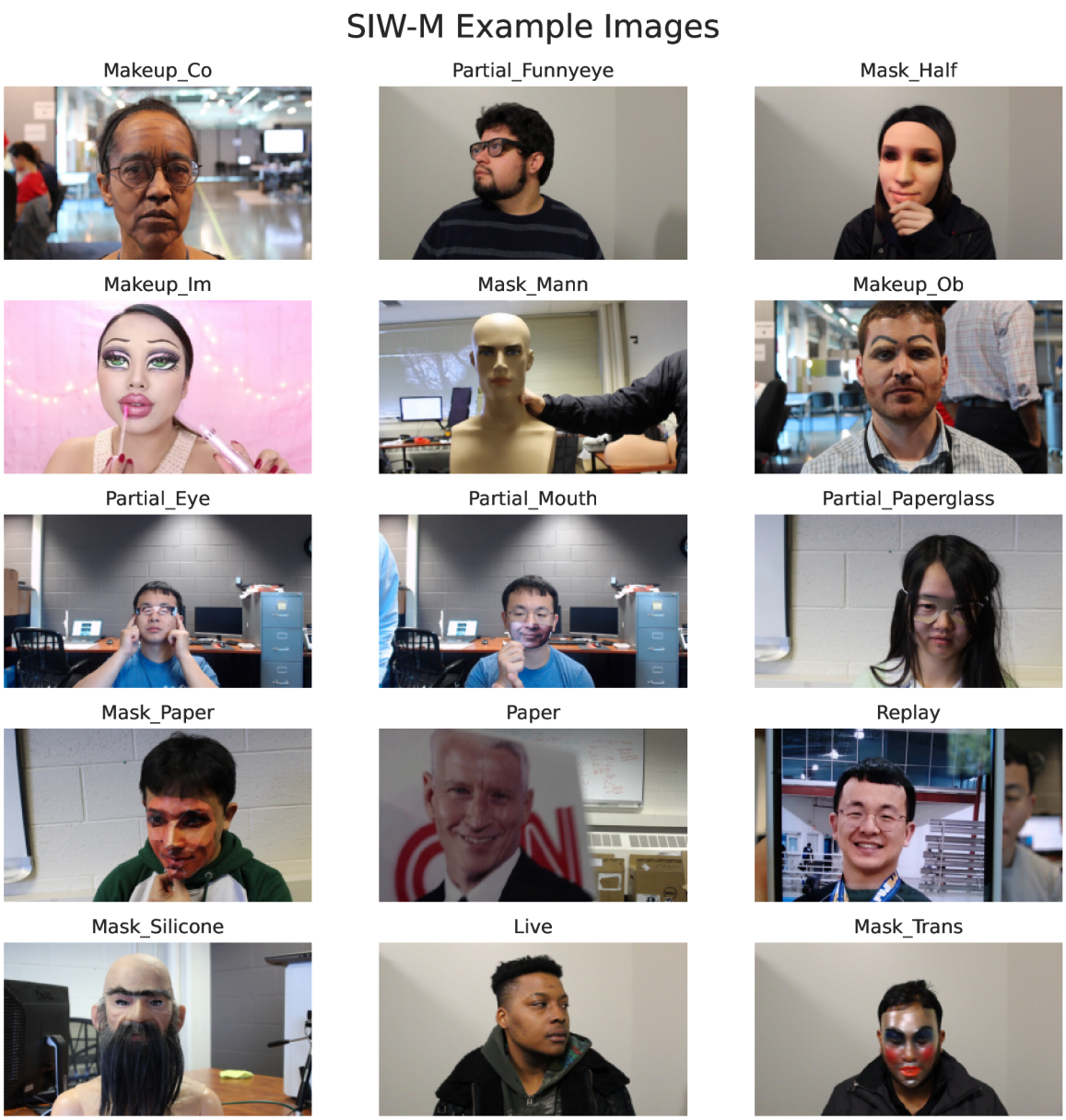


Figure 2.2: Example images showing every class in the SiW-M dataset.

2.3 Face Presentation Attack Detection Systems

To detect face presentation attacks, neural networks are commonly trained directly as multiclass classifiers. Another approach can be to use metric learning for anomaly detection and class distribution modelling. From learning these, embedding similarity measures can be used to distinguish genuine and attack samples [4], similarly to the approaches used in face verification. Due to the focus on explainability, further PAD methods and related face verification methods are also mentioned later in Chapter 3.

Chapter 3

Explainability of Convolutional Neural Networks

In the general sense, machine learning (ML) explainability is defined as the extraction of relevant knowledge from a machine learning model concerning relationships either contained in data or learned by the model [31]. At the same time, different literature uses the term more narrowly as a means of generating explanations for the predictions of a model [33]. This work uses the narrower meaning.

A close term to explainability is interpretability, which aims at understanding the inner workings of a model and its reasoning process [33]. Informally, explainability answers the question „Why?“, whereas interpretability answers the question „How?“. Some literature also uses the terms interchangeably. At the core of explainability, there is the term explanation. This is the answer to the why-question, and it provides an interpretable and faithful description of the model's behaviour – that is, it should be understandable to humans and it should be accurate to the model.

There are multiple ways to categorise explainable ML methods [33]. The first presented categorisation divides them into pre-model, in-model, and post-model methods, based on the development phase of their application. Pre-model methods target the input data itself, looking for data biases and assessing its distributions. In-model techniques are applied during model training and mostly correspond to interpretability methods. Post-model (post hoc) methods are used after training a model, and they mostly correspond to explanation generation methods. Another axis of division distinguishes model-specific and model-agnostic approaches.

To chart out the diversity of the explainable ML field, there exist the following categories of post hoc explainability methods [3]: visualisation, model simplification, local explanations, input attribution (feature relevance), example-based explanations and text-based explanations. The scope of this work is limited though, so only a subset of the available explainability methods is presented here.

In explainable ML, the following categories of evaluation are performed. Predictive performance evaluates the correctness of the model's approximation of the relationships in the data. Descriptive performance is the degree to which an explainability method objectively captures the relationships learnt by a model [31].

Visualisation

There are multiple visualisations that can aid in comprehending neural networks, even if they are not specifically designed as tools for explaining machine learning models. To simplify the visualisation of high-dimensional data such as feature embeddings from a model, traditional dimensionality reduction algorithms like Principal Component Analysis (PCA) or t-distributed Stochastic Neighbour Embeddings (t-SNE) can be used.

PCA [14] is a technique for projecting data to a lower-dimensional space such that the reconstruction error is minimised. Each dimension of a PCA projection is a linear projection of the original data, and the projections correspond to the eigenvectors of the original data. PCA is a deterministic, parameter-free, linear method, but is not robust to outliers in the data.

t-SNE [27] is a popular approach designed specifically for visualisation of high-dimensional data. It is a non-linear, iterative technique, which is non-deterministic, and the results depend heavily on the input hyperparameters. Whereas PCA aims to maintain the global structure of the data, t-SNE maintains pairwise distances between sampled data points to preserve the local structure of the data. Its flexibility and robustness to outliers come at the cost of hidden complexity, making cluster sizes, distances between clusters, and at times even cluster topologies unreliable [49].

3.1 Interpretability

The interpretability of machine learning models belongs mainly to the area of in-model XAI. While there are plenty of interpretable machine learning models, such as linear regression or decision trees, these are mostly not applicable to the task of PAD. Simpler techniques perform worse than large neural network models, so practitioners often sacrifice interpretability for greater performance [41].

An example of interpretability in a computer vision setting is the ProtoPNet [6] model. It uses a standard CNN classifier structure, but the loss function promotes training features to correspond to samples of the training data and therefore to represent prototypes. The subsets of the feature space are aligned with a single representative input region. The benefit of this is that a prediction can be explained through the prototypes, showing similar image regions from the training data. Retrieval of relevant training data through embedding similarity is also a generally used technique. On the other hand, the interpretability of neural network embeddings can also be questionable. Custom loss functions to promote interpretability are also used in the domain of face verification [52, 24].

Some techniques aiming at interpretability in recent face PAD works [33] also include the use of attention [42], spoof pattern estimation [48], depth map estimation [42], and auxiliary supervision [51].

3.2 Explanation Generation

Generating explanations for model predictions belongs to the post hoc explainability category. A major advantage of this category is that the models explained do not need to be designed specifically for explainability. Explanation generation methods can also be categorised. They can be divided into local and global methods, based on whether they explain the model behaviour only for a single data point or for the whole dataset in general.

The presented explanation can also take on many forms. Apart from visual explanations, there are also methods producing counterfactuals, rule sets, natural language explanations, etc. [32]. This section presents methods that can be applied to the task of face PAD. The most common explanation form in this task, and the only one covered in this work, is input attribution. Through analysing the model's internal behaviour, such as the gradients, activations of layers, or attention weights (model inspection), the most salient (important) regions are highlighted.

LIME

LIME [38] (2016), which stands for local interpretable model-agnostic explanations, is a universal model-agnostic perturbation-based method to explain ML models. LIME produces local surrogate model approximations to interpret the original model around individual data points. The surrogate (student) model is a simplified approximation of the explained original (teacher) model's behaviour for the given single data sample. To explain a data point, its local neighbourhood is sampled, and the teacher runs predictions on these perturbed samples. The surrogate learns these samples to create a local approximation of the teacher model around the data point. The teacher model is evaluated as a black box, and the surrogate model can be of an arbitrary type. Commonly used surrogate models are linear, such as decision trees for tabular data or linear regression (ridge/lasso) for image data. The word „interpretable“ in the method title thus stands for the interpretability of the surrogate model, as LIME is an explanation generation method.

LIME is popular method across many data domains. For image classification, the image is segmented using the quick-shift algorithm into 20-50 regions. The image is perturbed by randomly removing some of the regions, and a new prediction is made for each perturbation. The presence or absence of the regions is used to train the linear regression model. Its learnt weights tell which part was the most influential for the prediction, possibly both in the positive and negative direction.

This post hoc method generally gives faithful explanations, but the sampling itself presents an issue for the method's reliability. The local neighbourhood range around a data sample is defined by a smoothing kernel. Critique of the technique argues that this neighbour sampling scheme is unstable [53] and that the whole process depends on the hyperparameters choice. Additionally, since the sampled data points are perturbed, they may lie outside the model's training data distribution [28], due to dependencies in the input data space.

3.2.1 Primary Attribution Methods

These methods explain the behaviour of a neural network in terms of pixels' relevance to the network prediction. Initial approaches have attempted to visualise the gradient of the model predictions, backpropagated to the input image. However, per-pixel gradient is a noisy and hardly useful measure of the importance of input pixels [43]. Because of that, various ways of aggregating the gradient and stabilising the computation of the attribution maps have been a large topic in the development of these methods.

Integrated Gradients

Integrated Gradients [45] is a technique for generating gradient-based explanations of model predictions. One of the motivating problems for this work is the saturation of gradients; the

prediction function can be locally flat, and thus attribute zero gradient to an input change that contributes to the target prediction. Integrated Gradients overcome this problem by summing backpropagated gradients of the target class, accumulated over a „path“ of blends between the input image and a chosen baseline, as shown in Figure 3.1. The concept of absence of cause is explored, stating that an explanation must always reason in reference to a baseline. A black baseline image is chosen for the image domain, as it leads to a neutral prediction. Other baselines, such as random noise, are also suggested.

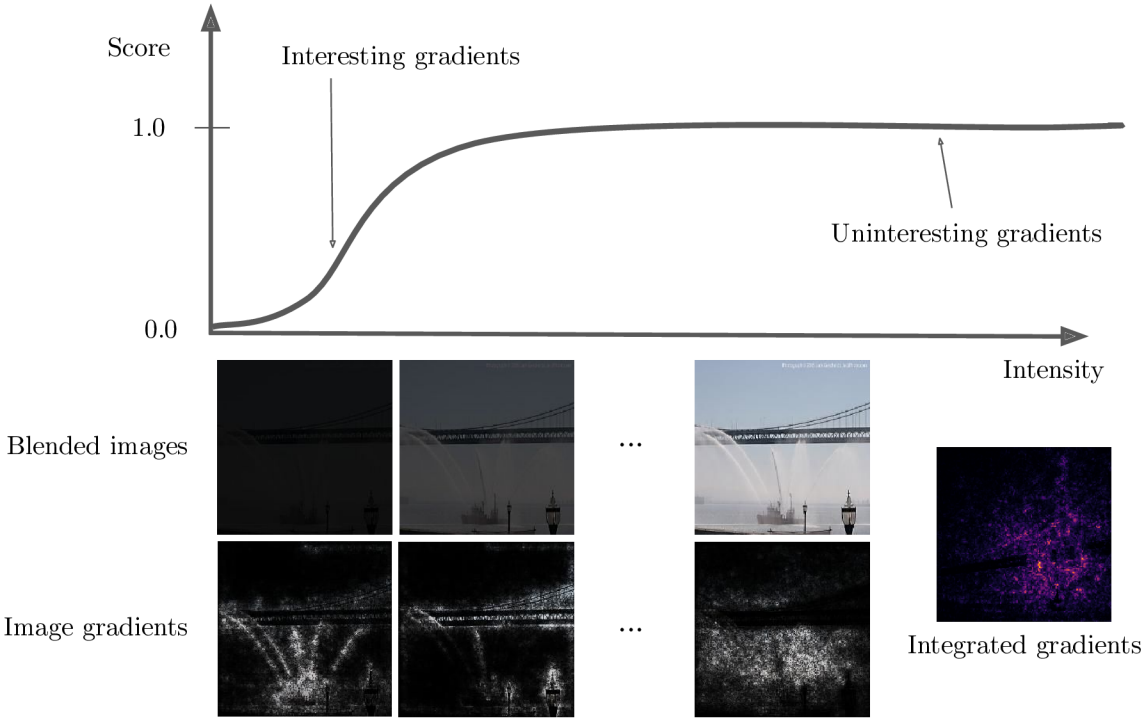


Figure 3.1: Integrated Gradients – top: gradient saturation problem; at high feature presence the model predictions are already capped and do not always give a meaningful feature importance measure. Down: solution to the saturation problem, accumulating gradient across blended versions of baseline and target image, and showing the averaged resulting explanation.[1]

3.2.2 Layer Attribution Methods

Layer attribution methods are used to explain which areas of an input image contribute to the prediction of the model. By visualising the contribution of image areas to the prediction made by the model, one can gain a deeper understanding of the model's prediction. Additionally, such methods can be used to identify factors that contribute to incorrect predictions and refine the model accordingly.

[1] Image adapted from Integrated Gradients author's slides https://github.com/ankurtaly/Integrated-Gradients/blob/master/icml_slides.pdf

CAM

Class Activation Mapping (CAM) [55] is the first explanation method of the CAM family. It is a gradient-free method, based only on forward-pass information. CAM is tied to a specific CNN architecture pattern; it requires the network to end in a sequence of average-pooling, linear layer, and then softmax. In the CNN layers preceding the average pooling, feature activations maintain spatial dimensions like $H \times W \times K$. Spatial dimensions are commonly downsampled iteratively throughout the network layers. Finally, the average pooling squashes these to $1 \times 1 \times K$. CAM presents the explanation as channel-weighted activations A^k of the last convolution layer, summed across channels, as per Equation 3.1, where w_k^c is the weight for activation map k and output class c.

$$L^c = \sum_k w_k^c A^k \tag{3.1}$$

CAM can be computed for an arbitrary target class simply by selecting dense layer weights connected to the chosen class output. The authors describe CAM as a class-discriminative weakly supervised localisation and evaluate the method on object localisation.

GradCAM

GradCAM [40] is a gradient-based attribution method that generalises CAM. It is applicable for any CNN or even transformer architecture, even if its outputs are still dependent on the model structure. An example GradCAM explanation is shown in Figure 3.2.

Formally, the GradCAM output for a given class c is a $H \times W$ localisation map L^c. During the forward pass, the activations of the target layer A^k and the target class score y^c are calculated. As per Equation 3.2, the gradient of the class score is with respect to the target layer is averaged across spatial dimensions (with $Z = HW$ being the normalising constant) to produce the weight α_k^c (contribution of the feature map k to the score of the class c).

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{3.2}$$

As per Equation 3.3, the localisation map is the channel-wise sum of the activations weighted by the per-channel α_k values, out of which only the positive part is kept.

$$L^c = ReLU(\sum_k \alpha_k^c A^k) \tag{3.3}$$

For a CAM-compatible architecture and the last convolutional layer as the target layer, the GradCAM formulation reduces to CAM, except for the ReLU filtering. At the same time, GradCAM offers more flexibility, as any activation map of the network can be used as the target layer, since the gradient can be always back-propagated to it. However, the author-proposed way uses only the outputs of the last convolution layer (just like CAM), and they report progressively worse localisation results with earlier target layers. Other sources also verify [37] that earlier layer activations lose class discrimination and become too noisy. GradCAM can also be computed for an arbitrary target class. The prediction for the chosen class is set to one, keeping all other ones at zero. This value is set as the „loss“, and gradient is backpropagated from it.

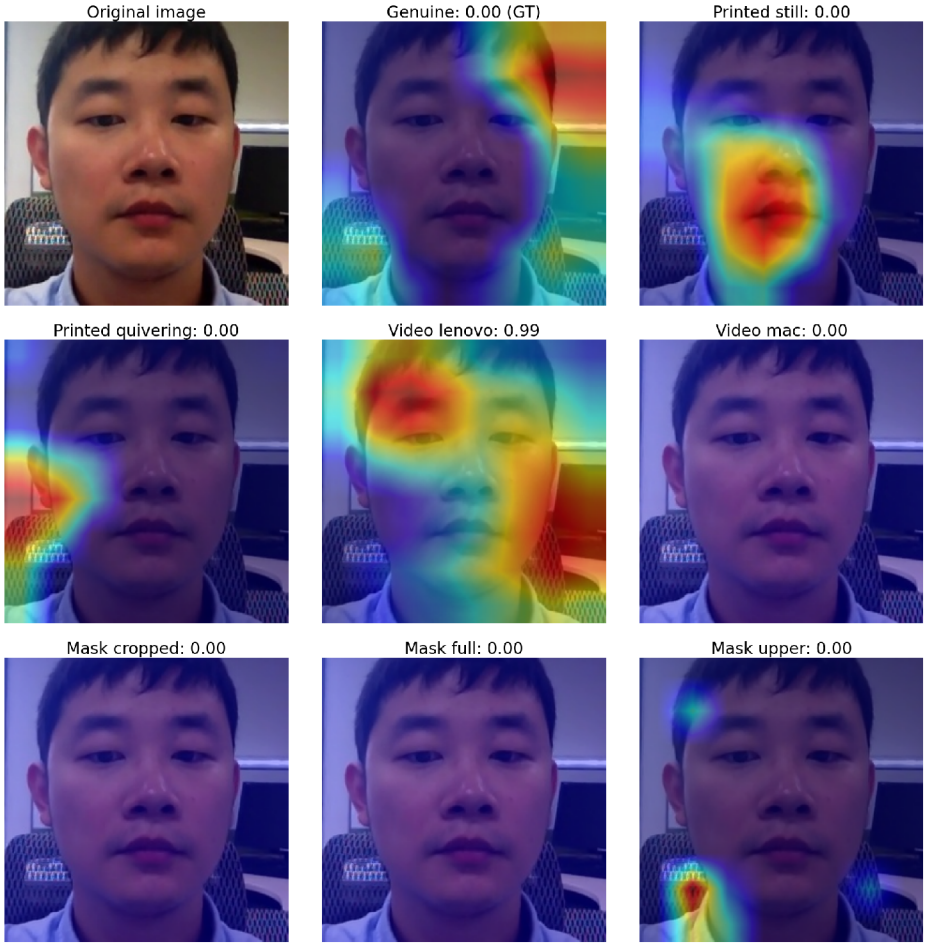


Figure 3.2: Example GradCAM outputs for a 'genuine' class RoseYoutu image. The prediction (an incorrect one) was made by a fine-tuned ResNet18 model. The explanations were generated for each data label and with the model's last convolutional layer set as the target layer. The prediction scores are shown next to the labels.

GradCAM++

Building upon GradCAM, the GradCAM++ method [5] differs only in the weighing scheme for the activations. Squared and cubed gradients are used to calculate the per-channel weights of the activations (as per Equation 3.4), which are then used the same way as in GradCAM (Equation 3.3).

$$\alpha_k^c = \sum_i \sum_j \frac{(\frac{\partial y^c}{\partial A_{ij}^k})^2}{2(\frac{\partial y^c}{\partial A_{ij}^k})^2 + (\frac{\partial y^c}{\partial A_{ij}^k})^3 \sum_i \sum_j A_{ij}^k} \tag{3.4}$$

HiResCAM

HiResCAM [9] keeps the same global approach as GradCAM, but takes a more granular approach to the gradients used. The weights $\alpha_{c,ij}^k$ (same as in Equation 3.2) are calculated

separately for each position i, j in the activation map. The rest of the formulation is the same as GradCAM. Despite the name, the output resolution is the same as in GradCAM.

More GradCAM-like Methods

Further layer attribution methods are presented here briefly. EigenCAM [29] computes the eigendecomposition of the activations and projects the activations on their first eigenvector.

In XGrad-CAM [10], the activations A_k are normalised by their sum, before being weighted by the gradient.

FullGrad [44] method takes the approach of calculating the gradients of biases throughout the network, upscaling all of them to the input image size, and summing them. This is the only layer attribution method mentioned here that provides an output at the input image resolution.

Further layer attribution methods include LayerCAM [19], Ablation-CAM [7], and Score-CAM [47], but their details are not important within the scope of this work.

All of the presented layer attribution methods follow similar principles and gradually attempt to solve claimed shortcomings of their predecessors, improving the methods' faithfulness, localisation performance, stability or class discrimination.

Figure 3.3 shows a comparison of selected CAM-based methods, applied to the same RoseYoutu image, showing the visualisations for the correct predicted class 'Video lenovo'. The model used for the prediction and the visualised layers is the same as in the previous Figure 3.2.

3.3 Evaluating Explanations

Since many explainability methods are being used in various scenarios, it is necessary to quantify the quality of their explanations. Although there is often no ground truth as to why a neural network should make a specific decision, various aspects of descriptive performance can be evaluated.

A recent explainability framework Quantus [16] defines the explanations metrics as follows:

- Faithfulness quantifies the extent to which explanations follow the predictive behaviour of the model, and asserts that more important features have a greater impact on the model's decisions.
- Robustness measures the extent to which explanations are stable in the face of small perturbations in the input, when the model output remains approximately the same.
- Localisation tests whether the explanation is centred around a region of interest, such as a bounding box.
- Complexity captures the extent to which explanations are concise, i.e. that few features are used to explain a model prediction.
- Randomisation tests the extent to which explanations deteriorate as the model or the data labels get randomised.
- Axiomatic measures the extent to which explanations satisfy certain axiomatic properties.

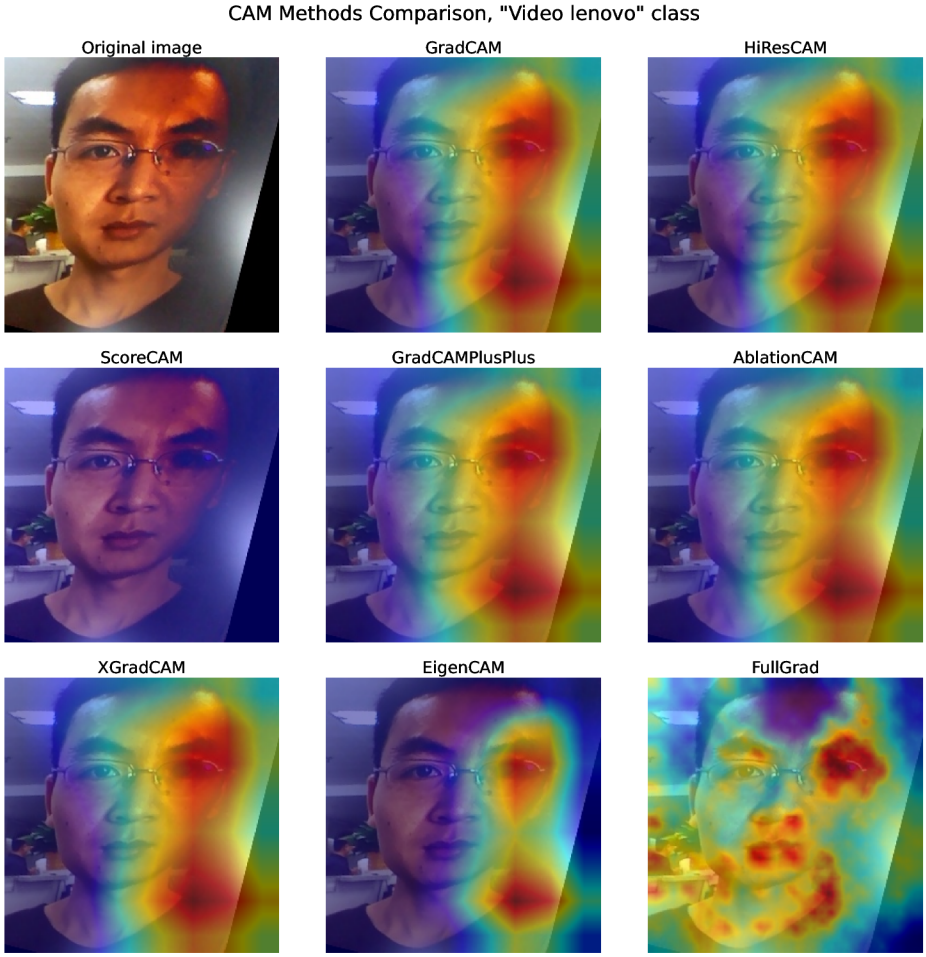


Figure 3.3: Example layer attribution methods outputs for a 'Video lenovo' class RoseYoutu image.

The most-often evaluated aspect is faithfulness[2]. To give an example of viewing a method through these criteria, LIME image explanations are designed for faithfulness, as they are obtained through the model's predictive behaviour in the face of perturbations. LIME explanation complexity is chosen by the image segmentation granularity, and by the surrogate model complexity. While high faithfulness and low complexity are both important aspects of an explanation, such goals are often contradictory. Robustness of LIME depends on the sampling procedure.

The quantitative evaluation of input attribution explanations is commonly done through the perspectives of localisation and ablation.

For the input attribution methods presented earlier, there exist metrics that assess how faithful the visualisations are. Depending on how the model behaves when removing parts of the image selected by the input attribution methods, the quality of the explanation can be estimated. Some of these methods are ROAR [17], and ROAD [39]. Other methods also try to establish baselines for the evaluation of saliency methods [1].

[2]Viewed as the content-evaluation metrics correctness and output completeness in a recent review paper [32].

3.3.1 RISE

The authors of RISE [36] present an input attribution method based on the occlusion of the input image, and standardise two metrics for evaluating the faithfulness of attribution explanations. These metrics are called the deletion metric and the insertion metric. To compute these metrics, the original image is perturbed and the model's prediction score changes are observed.

The deletion metric is based on the intuition that if an important region of the image is removed from the input, the target prediction score should decrease rapidly. For a given attribution explanation, input image pixels are ranked based on their importance. Then, the input pixels are iteratively removed in decreasing order of importance, and model predictions are made for each perturbation intensity. A black baseline canvas is left in place of the removed pixels. As an aggregate measure of the prediction score dropoff, the area under the deletion curve (AUC) is calculated.

The insertion metric takes the same approach and performs a complementary evaluation. Pixel importance is ranked inversely and the most important pixels are iteratively added to the canvas. The insertion metric canvas is a blurred version of the input image. Predictions are made again for each perturbation level and the AUC is calculated.

Figure 3.4: RISE evaluation example – left to right: explanation (generated by RISE method) overlayed on an input image, perturbed image for deletion metric, perturbed (partially blurred) image for insertion metric, deletion metric curve, insertion metric curve[3]. Both images are at a perturbation level of 32.5% of deleted/inserted pixels. Perturbation metric curves show prediction confidence score as a function of perturbation intensity.

Although the deletion and insertion metrics are complementary in the masks of image regions kept or removed, their curves do not show the same information. For the deletion metric, the lower is better, whereas the opposite is true for the insertion metric. Additionally, the 0% points at each curves are not necessarily the same because of the different baseline image choice for the two metrics. An example of both metrics is shown in Figure 3.4.

3.3.2 ROAR

Remove and Retrain (ROAR) [17] follows up on the deletion metric of RISE, but casts doubt on the validity of the perturbation process. The authors claim that a decrease in the model accuracy[4] caused by the perturbations comes from a shift in the distribution of the data and not necessarily from the information removal. To prove this, they add retraining after the

[3]Image source: RISE official demo webpage https://cs-people.bu.edu/vpetsiuk/rise/demo.html.
[4]RISE perturbation metrics show the prediction score dropoff, which also leads to an accuracy drop.

perturbation step and evaluate the accuracy of a model trained on the perturbed samples. The results show that accurate models can be trained even on strongly perturbed data, suggesting a large redundancy in the inputs. With a random input attribution, even data with only 10% pixels retained can be used to train a model with good accuracy (63.5% on ImageNet, down from 76.7%). The authors compare explainability methods using ROAR, finding that many primary attribution methods (e.g., Integrated Gradients) do not beat a random attribution baseline.

3.3.3 Sanity Checks

In Sanity Checks for Saliency Maps [1], explanation methods are compared to baseline techniques such as Sobel edge detection, to investigate whether they reliably beat baseline techniques. Model weights randomisation and data labels randomisation is also performed to explore how the explanations change. Rank correlation between true and baseline methods is performed, together with subjective visual evaluation. Results show that some saliency methods perform just as bad as baselines insensitive to the model and data. For example, Integrated gradients highlights similar input structure even with random data labels. In contrast, GradCAM passes the tests.

3.3.4 Discussion

Model explanations are helpful to understand the predictive behaviour, but must be used with care and thoroughly evaluated, before they can be trusted. In failure modes, the blame is also unclear, as problems can stem from the explanation method, the model, or the data. Additionally, just like adversarial attacks can alter CNN's behaviour, the explanations have also been shown to be easily manipulated [8]. Subjective visual evaluation of explanations therefore runs into many problems, and explanations alone make the user susceptible to confirmation bias [28]. When only the outputs of the model and the explanation are compared, there is also no guarantee that the explanation follows the same reasoning as the model [32]. Also, many explainability methods end up disagreeing with each other when compared [22], making it difficult to choose the appropriate set of methods to use.

Chapter 4

Method

This chapter first discusses important solution aspects influenced by related work, then presents approaches to evaluating the proposed work and finally, presents the solution itself.

4.1 Comparison to Related Work

This work takes inspiration from works at the intersection of face PAD, explainability methods, and evaluation of explainability methods. A closely related work is a recent Exploratory study of interpretability for face PAD [41]. In their work, a custom, minimalist CNN model is employed for the face PAD task, and GradCAM explanations are used. RoseYoutu dataset is used, but different use of labels (explained later in Section 5.1.3) makes direct comparison impossible. Predictive performance is also evaluated across datasets. When comparing generated explanations, it is worth noting that the authors' model used has the last convolutional layer output at a 28×28 resolution.

In another closely related piece of work (Myope models [34]), GradCAM++ is applied to analyse face PAD predictions to determine the importance of background image regions. The predictive performance is evaluated across training modes and across datasets, and ablation studies are performed. However, the explanations are blindly assumed to be of high quality, and are used merely as a quick visualisation tool. The predictive performance is also reported for a model making use of the background image regions, which are likely confounding factors. In comparison, the current work aims to go deeper into evaluating the explanations.

Another line of work [26] in the task of face verification aims to generate explanations (including GradCAM, GradCAM++, and LIME), and also evaluate them through deletion and insertion metrics. For applying the used methods, Siamese training regime and similar approaches would have to be adapted to the face PAD task.

There are two rare cases of works in the task of face PAD that both generate model explanations, and evaluate them. A recent Master's thesis [35] trains a general CNN model, generates primary attribution explanations, and applies evaluation metrics provided by the Captum library [21]. The explanations' evaluation is limited, as no in-depth across-sample analyses are performed, and no comparisons to baseline solutions are performed. The suitability of the chosen explanation evaluation metrics can also be questioned, just as the metrics' values, which are hard to interpret. With a low predictive performance, it is unclear whether the model or the explanations are to blame for the inconclusive results.

A very recent work [18] investigates the gender biases of face PAD explanations, and compares the deletion and insertion metric curves across male and female subject images. However, suitability of the chosen attribution methods (GradCAM, GradCAM++) is not verified, and the overall results are hardly conclusive.

4.2 Predictive Performance

High predictive performance is a necessary condition for meaningful model behaviour analyses. Below, standard evaluation metrics are presented.

4.2.1 Performance Metrics

Predictions of the predictive performance can be, depending on the evaluation framework, treated as multi-class or binary. The following list summarises the metrics commonly used in biometry for sample-level prediction evaluation. The bona-fide category represents the negative samples, and the positive samples comprise all the attack categories. With these categories, the prediction results are categorised as true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

- Accuracy = $\frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$
- Precision = $\frac{\text{TP}}{\text{TP+FP}}$
- Recall = $\frac{\text{TP}}{\text{TP+FN}}$
- F_1 Score = $\frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision+Recall}}$
- False Rejection Rate (FRR) = $\frac{\text{FN}}{\text{TP+FN}}$
- False Acceptance Rate (FAR) = $\frac{\text{FP}}{\text{TN+FP}}$

The FAR and FRR metrics are also often evaluated at multiple thresholds of the decision boundary value. The error rate at which they are equal is the Equal Error Rate (EER). In the context of presentation attack detection, FAR is also called Bona-fide Presentation Classification Error Rate (BPCER). Similarly, FRR is called Attack Presentation Classification Error Rate (APCER). The average of APCER and BPCER is the Average Classification Error Rate (ACER). When relating FAR and FRR, it is also common to set a target value for one of them (e.g. FAR = 10^{-6}) and measure the other one (e.g. FRR@FAR=10^{-6}). This fixing of a threshold expresses various importance given to classification errors.

4.2.2 Evaluation Framework

The training and evaluation of the PAD system are performed in the three modes listed below. The following modes have been used in the PAD task by Sequeira et al. [41].

One Attack

The model is trained as a binary classifier, predicting the genuine and attack classes. Training and test data contain only a single attack class and bona-fide samples. This presents the model's ability to detect a known attack under the optimistic condition of knowing the attack type beforehand. In a practical setting, an ensemble can be built from these models.

Unseen Attack

In the unseen attack (leave-one-out) mode, the model is trained as a binary classifier. The test class is excluded from the training set. This setup challenges the generalisation performance of the model. The predictive performance results here are the most realistic indication of the robustness of the model.

All Attacks

All attacks are present in the training, validation, and test dataset splits. The model can be trained as either a binary or multiclass classifier. This setup best follows the approach for real deployment of the model.

The purpose of using various training modes is not only to assess their predictive performance. The differences in the learnt behaviour are also interesting for the explainability aspect – e.g., how do the explanations differ between a known and an unknown attack? How do the explanations change when a model predicts the specific attack class instead of just a binary label?

4.3 Descriptive Performance

Layer attribution methods have been chosen for the explainability aspect. In line with related work, these have been employed with success for the face PAD task, their performance does not pose a significant limit for the evaluations, and they are not tied to a specific model architecture. Primary attribution methods (Integrated Gradients) do not scale well for a large number of image samples, their critique presents many shortcomings making them less trustworthy, and their higher granularity makes them more complicated to evaluate.

While layer attribution methods provide meaningful explanations in tasks performing object detection, some face PAD categories (e.g. replay attacks) may be also better characterised by global cues, such as colour shift, moiré patterns, or other artefacts. Since the applicability of the chosen methods is not clear, much emphasis is put on evaluating this aspect.

Most of the layer attribution methods presented earlier are chosen as candidates to use, namely GradCAM, GradCAM++, EigenCAM, XGrad-CAM, HiResCAM, Ablation-CAM, Score-CAM, LayerCAM, and FullGrad. Apart from the standard published methods mentioned before, a test-time augmented version of GradCAM (GradCAM_aug) is used. This method applies GradCAM on 6 augmented versions of the input image (multiplied by 0.9, 1.0, 1.1, and either flipped horizontally or kept the same). The large volume of selected techniques hopes to find a suitable one to present meaningful visualisations.

For evaluating the generated explanations, perturbation metrics have been chosen, in line with relevant literature. Localisation metrics make use of object position annotations, which are not available, and also not always suitable, in the task of face PAD. Deletion and insertion metric also localise evidence for the model's prediction, but the explained regions are evaluated through the model itself, without the need for a ground-truth label. Example perturbed images are shown in Figure 4.2.

Both deletion and insertion metrics ignore the actual values of the attribution map and only work with their ranking. This is common for many similar evaluation methods (e.g., Pixel-Flipping [2] and XRAI [20]) and has later been criticised for the lack of model

Figure 4.1: Perturbation metrics baseline images for an example image.

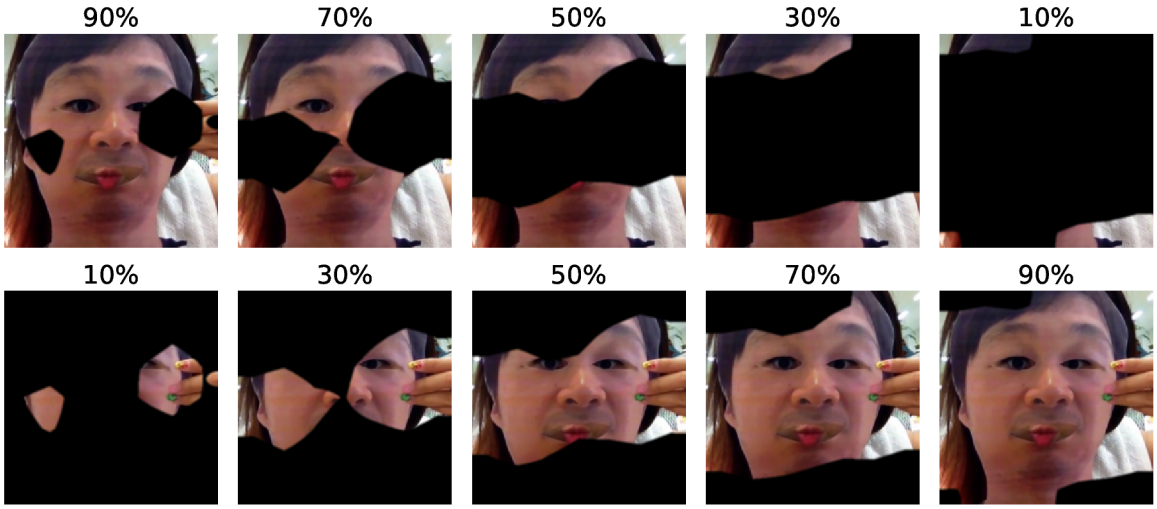


Figure 4.2: Example image for at various perturbation levels used for the deletion and insertion metric. The percentages indicate area of original image kept. Full and empty images are omitted for brevity.

calibration [13]. At the same time, the magnitude of the input attribution is more relevant for the primary attribution methods, which lead to a long-tail distribution over a large value range.

For the perturbation methods, a fitting baseline image must be chosen. Considered baseline images are shown in Figure 4.1. The ideal baseline image is uninformative, leading to neutral model predictions, and also does not create out-of-distribution samples. Both aspects are later studied.

4.4 Approach

The objective of this work is to explore, apply and experiment with explainability methods in the PAD task. The initial research questions are:

- How to apply explainability to the task of face PAD?
- How to use explainability methods to improve the predictive performance of the PAD system?
- How to understand the failure cases of the PAD system?

First, the task of non-explainable PAD is approached. The RoseYoutu dataset has been chosen for development and experiments. A general CNN classifier is trained for the task of

face PAD. Next, explainability methods, such as CAM methods, are applied to the model to understand the behaviour of both the model and the methods. The model's predictive performance is assessed, and the explainability methods are evaluated. Based on the results, the PAD solution is further developed, along with the explainability methods.

Once the results of both are satisfactory, a comprehensive evaluation is performed. Model performance is assessed across various modes of the evaluation framework, and the performance is profiled based on subgroups present in the datasets to find potential shortcomings of both the model and the data.

For explanations evaluation, it is necessary to perform sanity checks, to compare the methods with baselines. Simple baseline solutions are to be set as reference points.

A suitable sanity check for the explanation methods should be a model-independent or data-independent technique. The first such method proposed is the Sobel edge detector. Related work finds edge detection to be perform just as well in explanations evaluation [1]. Another sanity check used in literature is randomising model weights and comparing the explanations generated with the untrained model. Given the faithfulness evaluation through perturbation metrics, this method is not used.

Since the face images are aligned to be centered, it can also be the case that the model would always focus in the centre of the image. This is not necessarily wrong, but an explanation that would always circle the image centre is data-independent, and thus not useful. So as another sanity check, a centered circle explanation is proposed.

The last sanity check is a random explanation. In comparison to primary attribution methods, here the chance of highlighting a credible input region is much higher, given the lower resolution of the explanations.

Chapter 5

Implementation

As a first step, a baseline PAD system was developed. For the RoseYoutu dataset, loading, cleaning, and preprocessing were performed. Initially, a ResNet [15] model was used as a multiclass classifier for the task. A standard ResNet-18 variant of the model was used, pre-trained on the ImageNet1k dataset. The classifier head was replaced to adapt to the number of classes. Later development primarily makes use of the EfficientNetV2_s [46] architecture.

The implementation uses the Python programming language and the PyTorch deep learning library. The solution comprises developer-targeted scripts for loading, cleaning, and preprocessing the datasets, training a model, and evaluating a model. For monitoring the training process, the Weights&Biases tool is used. The ResNet and EfficientNet models are adapted from the official PyTorch repository. For CAM techniques, the „grad-cam“ library [12] implementation is currently used.

5.1 Dataset

The desired dataset must contain only images with faces, aligned to specified positions, and cropped to the target size. Cropping is done for convenient training use. The alignment ensures consistency for evaluation and explanation generation. Initially, RoseYoutu was chosen for development because of its good availability.

The RoseYoutu dataset is available in its full version and in a pre-processed version. The pre-processed version consists of roughly 68 000 image samples of size 256^2, spanning 10 person identities. This version was used for the initial development of the solution. Although the faces are aligned and cropped mostly consistently, manual inspection has found alignment errors. In addition, the image resolution is lower than the native EfficientNet resolution of 384^2. The full version of the dataset (as described in Section 2.2.1) is only available as videos, a frame extraction process was developed.

5.1.1 Extraction

Face alignment is performed using an off-the-shelf MTCNN [54] model. This model was chosen for its popularity and ease of use with a dedicated pip package[1]. MTCNN predicts five face landmarks: left and right eye centres, nose, left mouth corner, and right mouth

[1]Source: https://pypi.org/project/facenet-pytorch/

corner. At the same time, MTCNN also predicts a bounding box for the face, which is not aligned with the landmarks, but roughly encloses the face area.

To extract images for the dataset, k frames are regularly sampled from the video. Next, MTCNN predicts the face position in the frame. An affine transformation is found for the set of predicted face landmarks, to transform them into their reference positions (see Figure **??**). The transformation is computed using three points: left eye, right eye, and mouth centre. Finally, the image is cropped to the target size of 384^2.

Since MTCNN predictions produce false positives, more checks are performed to verify the samples. Computing the variance of the cropped image Laplacian serves as a blurriness detector, and all images with a variance less than 20 are dropped. This threshold has been empirically found to filter out false MTCNN face detections. Sometimes also the affine transformation fails, so a second round of MTCNN prediction is performed on the cropped image to verify the presence and position of the face. Since the original videos' metadata sometimes falsely report the video orientation, the first video frame is used to find potential rotation necessary to fix this. MTCNN is used here again to predict on 90°-spaced rotations of the image, out of which the highest-confidence prediction is kept as the correct upward orientation.

Two dataset variants have been chosen for the development and experiments. The first one („full“) uses 50 frames sampled from each video, leading to total of 187 429 image samples. This variant has been used for training of the models and for their predictive performance evaluation. The second variant („single“) takes $k = 2$ samples, leading to 4599 image samples. It is used for experimentation with explanations. Both dataset variants use the same input videos, and the same attack + identity separation. Their only difference is their scale. Whereas model training works better for a larger amount of samples, and provides more consistent evaluation results, generating predictions' explanations and their analyses were too difficult at such large dataset scale. For this reason, the single variant serves as a smaller representative sample.

5.1.2 Splitting

The generated dataset is split for training/validation/testing, following the splitting chosen in a recent work by Neto et al. [34]. In all evaluation modes (all attacks, unseen attack, one attack), the dataset is split based on the captured person's identity, so as not to create a training-test overlap. Of the 20 identities, the first 10 are used for development (9 for training, 1 for validation) and the last 10 for testing. In the unseen-attack and one-attack modes, the dataset is also split based on the attack classes. Note that in the unseen-attack scenario, it is not enough to split the dataset based on the attack without considering the person identities. Although this would ensure no overlap for the attack samples, the genuine samples of one person could end up in either the training or test set. The model would then be evaluated by part on known data (genuine class, same identity), and by part on unknown data (different class or identity). In the unseen-attack scenario, the same attack is being used for the validation and testing data. Retrospectively, this is a mistake, as choosing the best model for this given attack class is already optimising for a goal that is supposed to be unknown.

5.1.3 Labels

Originally, the RoseYoutu dataset uses eight class labels, as shown earlier in Figure 2.1. In my work, these labels are merged into 4 (genuine, printed, video, mask) to enable potential

Figure 5.1: RoseYoutu samples – original images in the top row, augmented images in the bottom row. Standard ImageNet preprocessing is performed before the augmentations, making the value range go below zero. Zero-filled black rectangles thus appear grey. This is not an issue, as it is still uninformative, and the augmentations are consistent with what is later used in the explanation-related perturbations.

across-dataset evaluation with the SIW-M dataset. In the end, this evaluation has not been performed. The label merge makes the dataset easier for the all-attacks mode, but more challenging for the other modes.

5.1.4 Variants

To enable easy switching between dataset variants, a simple referencing hierarchy is used. Top-to-bottom, the list of available dataset instances is saved in a pickled Pandas DataFrame. Each row is one dataset instance (e.g., RoseYoutu, unseen-attack training mode, test attack „printed“, „full“ dataset length), and it links to the file lists for the given dataset. For each dataset instance, there are training, validation, and testing file lists. A dataset instance file list then contains the image filenames and their annotations. None of the file references contain absolute paths to make the dataset instance portable. All[2] the file lists and dataset lists are saved as Pandas DataFrame pickles for easy I/O and powerful Pandas data manipulation.

5.1.5 Preprocessing

Since the models used are pre-trained on the ImageNet1k dataset, the preprocessing is done in the same way, normalising each image to the ImageNet mean and standard deviation. On top of that, augmentations are performed. TrivialAugment [30] augmentation has been chosen for its parameter-free simplicity. More importantly, random erasing of the images is performed, removing a rectangle of 2-33% of the image area in 50% of the input samples. The erased area is replaced by zeros. While a standard approach in image classification, here it is important for the data distribution of perturbed samples when evaluating the explanations. See Figure 5.1 for examples. Apart from the image artifacts produced by the augmentations, the images may already contain black areas produced by aligning and cropping transformations in the cases where the original frame did not contain enough background area.

[2]Except for metadata about each dataset instance, saved in a json file. These files are not used in the final solution.

5.2 Model Setup and Training

Models are trained for a maximum of 15 epochs through the standard cross-entropy loss, classifying the input images to one of the five categories (Genuine, Printed, Video, Mask, Other), even though the dataset labels are merged to four classes. This unification of predicted classes and data labels was done to make the model compatible with the SIW-M dataset. In the end, this dataset was not used, so the category never appears in any ground truth labels[3].

The Adam optimiser is employed. Predictions and gradient computations during training are performed in mixed float precision[4] and with a 4-batch gradient accumulation. A low learning rate of 10^{-5} is used and is divided by 2 when the validation loss does not improve over 3 training epochs. An early stopping mechanism is also used, halting the training if no progress in the validation loss is made during 5 epochs. The model weights that achieve the highest validation accuracy are saved for future use.

The same setup is used for all training modes, with the exception of the classification head. Initial development has used a multiclass (5-way) classification head for the all-attacks mode, and a binary classification head for the remaining scenarios.

During further development, additional model classification heads were added. The motivation was to provide the model with a more granular training signal, to improve the performance in the unseen-attack and one-attack modes. Thus, 2-way, 5-way, and 8-way classification heads are used, for training on binary, unified, and original RoseYoutu labels. Each classification head is trained using the cross-entropy loss, and the only difference in the models is how the heads are used for test-time prediction. All-attacks mode uses the 5-way head for inference, and the remaining ones use the binary heads. Each classification head is only a linear layer, which processes the model's embeddings. As a reminder, these embeddings are created by average-pooling the last convolutional layer activations. For these model changes, Torchvision implementations of the used models are adapted. Custom forward methods are used and swapped in dynamically, so that a model can always be used in the standard PyTorch way of calling its instance (`model(inputs)`). This is practical for using explanation generation methods, that make use of this interface.

Two development environments were used, a local laptop for small-scale experiments, and a remote computational grid Metacentrum[5] were used. The handling of datasets makes it possible to produce same results in both places, and the Weights&Biases ML experiment management platform[6] collects all of the training logs, run parameters, and evaluation metrics.

5.2.1 Reproducibility

An effort has been made to make any findings reproducible. For this reason, a fixed seed is used for training, being applied to the Python, PyTorch, and NumPy random number generators, to each DataLoader worker process, and to dataset shuffling. Where possible, no randomness is introduced in the first place, e.g., for splitting of the dataset. Since all hyperparameters and training progress logs are gathered in the Weights&Biases, all results

[3] When performing experiments, checks were also performed to verify that the unused label never appears in the model predictions either.

[4] 16-bit precision is used, except for final softmax.

[5] https://metavo.metacentrum.cz/

[6] https://wandb.ai/

are reproducible. All evaluation is also reproducible as no random methods are used and all setup details are loaded from the saved training run details.

5.3 Explanations Generation and Evaluation

All CAM-family explanations used have the resolution of the last convolutional layer output of the model used. For ResNet18, it is 7×7, and for EfficientNetV2s, it is 12×12. However, the chosen target layer is not the last convolution layer but the batch normalisation layer that follows it. This is the same for both models. The differences between the target layers were minimal, but the latter leads to more peaky explanations.

Explanation methods that provide higher-resolution output have been rejected because they are computationally expensive and noisy. FullGrad [44] method uses the sum of all bias gradients, all upscaled to the original input size. This is feasible for a smaller model like ResNet18, but already takes many seconds for EfficientNetV2s (42 layers). Strided operations in the network layers also introduce grid-like patterns in the explanations, which is detrimental to the explanation quality.

The grad-cam library code, which generates the explanations, makes the use of PyTorch forward hooks to capture the activations values. But, by default, the forward pass layer operations are performed in-place, destroying the old output when the next layer is computed. For consistent results, the in-place flag is disabled in the model layers.

The perturbation metrics are implemented based on the input image and the generated explanations. Since the explanations have a low resolution, they are upsampled to the input images resolution with bilinear interpolation. Since the produced input attribution maps still do not contain a smooth distribution of values, a strongly blurred version of the CAM is added to the original upscaled CAM. A large blur sigma of 129 is used to make sure that the values globally have a smooth and flat distribution. This allows for a smooth ranking of the sorted explanation. The ranking is then used to compute decile thresholds. From these, 11 perturbation masks are generated (ranging from 100% to 0% with a step size of 10%).

Using the selected baseline (black, mean, blurred, or darkened blur), the image is masked to keep a given percentage of the image and replace the rest by the baseline. Prediction is run again on the perturbed images.

During the development of the perturbation metrics, it became evident that the face PAD data domain contains a lot of redundancy in the images. It is not expected that the predictions should suffer as drastically from the perturbations as they do in RISE and similar works using datasets like ImageNet or MSCOCO14.

The implementation of the explanations evaluation was done with the layer attribution methods in mind, so another explanation generation method was left unused. LIME explanations generation was implemented, but it did not get used in the end.

Chapter 6

Experiments

All experiments are performed on the „single“ variant of the RoseYoutu dataset, on the test set. Unless specified otherwise, the training/evaluation mode is all-attacks.

6.1 Models Trained

Models are trained as described in Section 5.2. The setup is the same for all training modes.

6.1.1 All Attacks

The main models in the experiments have been trained in the all-attacks training mode.

ResNet18

Although the ResNet models achieve a generally lower prediction performance, this model has also reached a test accuracy of 99.03%. The APCER and BPCER are 0.6% and 0.5% respectively, with the EER of 0.58%, with a threshold tuned on the validation set. The model's training progress can be seen in Figure 6.2, under the name Top-ResNet. The accuracy is reported as a multiclass metric, whereas the remaining ones as binary, which explains their difference.

EfficientNetV2s

The overall best model reaches a test accuracy of 99.18% after 7 epochs of training. Its training progress can be seen in Figure 6.2, under the name Top-EffNet. Furthermore, the model attains the APCER of 1.050% and BPCER of 0.089%. An EER of 0.84% is reached when the threshold is set based on the optimal validation set EER threshold.

The complete runs metrics and logs can be viewed in the Weights&Biases project[1].

6.1.2 Unseen Attack

Results for models trained in the one-attack mode are shown in Table 6.1.

[1] Available at: wandb.ai/petrmiculek/facepad

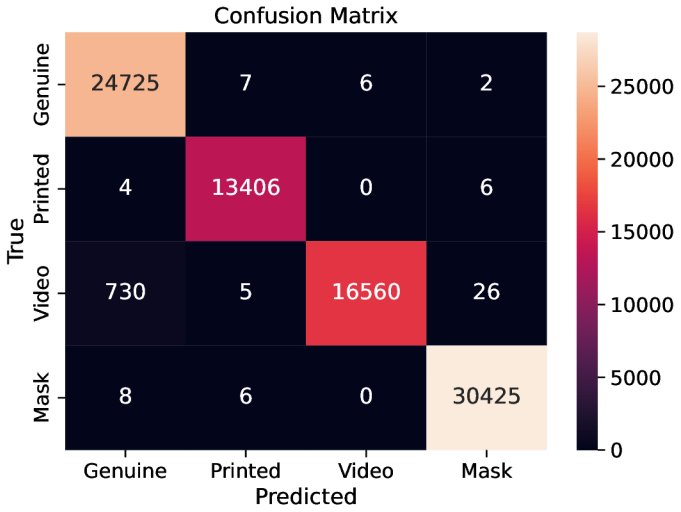


Figure 6.1: Confusion matrix for the best-performing all-attacks model on the RoseYoutu test set („full“ variant).

Table 6.1: Predictive performance evaluation for the best-performing unseen-attack mode models for each attack left out. Error ratios in %. EER threshold has been selected based on the RoseYoutu validation set.

Attack	EER	APCER	BPCER
Printed	14.702	36.110	4.710
Video	38.484	68.550	1.867
Mask	9.844	47.520	1.813

6.1.3 One Attack

Results for models trained in the one-attack mode are shown in Table 6.2.

6.2 Ablation Studies

Certain aspects of model training are evaluated here. The role of multi-task training was also supposed to be evaluated here, but preliminary data was inconclusive, and a more in-depth comparison was not performed in the end. However, the intial conclusions suggested that the more fine-grained training signal does not help the predictive performance.

6.2.1 Pretraining

During development, pretrained models were used, so that fine-tuning can build upon the robust feature extraction of the backbone model. This comparison aims to verify its benefit. A comparison is made between a pretrained and randomly initialised EfficientNetV2s models with an identical setup. Both models were trained for up to 15 epochs, both using the three multi-task losses. The models are shown in Figure 6.2 as Top-EffNet and No-Pretrain. Both the training and validation accuracies differ significantly throughout the

Table 6.2: Predictive performance evaluation for the best-performing one-attack mode models for each attack left out. Error ratios in %. EER threshold has been set for each model based on its validation set performance.

Attack	EER	APCER	BPCER
Printed	0.014	0.246	0.032
Video	0.741	1.409	0.073
Mask	0.020	0.036	0.004

training. The best epoch by the validation accuracy is 7 and 13, respectively. The test set accuracies are 99.18% for the Top-EffNet model, and 98.09% for the No-Pretrain model. Despite the different data domains of the pretraining ImageNet dataset and RoseYoutu, it is shown that pretraining is beneficial to the model's predictive performance. Fine-tuning can thus be run for fewer epochs, preventing possible overfitting.

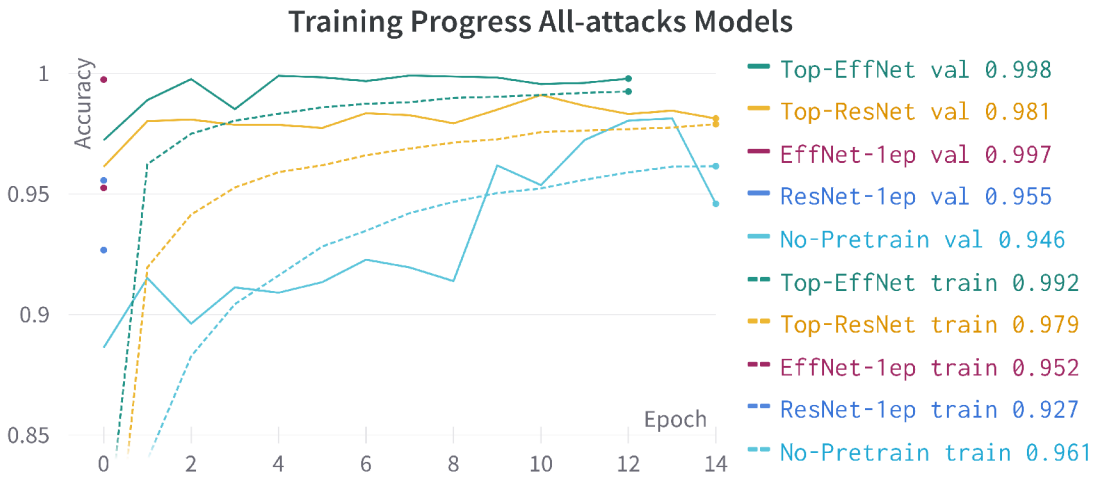


Figure 6.2: Training progress of compared models in all-attacks mode. Training and validation accuracies are shown for each run, with their best value shown in the legend. Note that the training accuracies are computed during training on augmented samples, deflating the value.

6.2.2 Single-Epoch Training

Once an effective training setup was found, a question was raised whether many epochs of training are even necessary. A comparison is thus made between models trained for a single epoch, and those trained for longer. The single-epoch models differ from the standard setup (defined in Section 5.2) only in the learning rate, which is set to $1e-4$ instead of $1e-5$. The best EfficientNetV2s (Top-EffNet and EffNet-1ep) and ResNet18 (Top-ResNet and ResNet-1ep) models are shown in Figure 6.2.

The best epochs by validation accuracy are 7 and 10, respectively.

Table 6.3: Comparison of baselines' deletion metric AUCs for GradCAM explanations. Mean and standard deviation are reported for each baseline.

Baseline	AUC μ	AUC σ
black	0.660	0.305
mean	0.661	0.312
blur	0.587	0.360
blur_div4	0.616	0.340
blur_div8	0.671	0.319

6.3 Descriptive Performance

The faithfulness of the explanation generation methods is evaluated using the deletion and insertion metric.

6.3.1 Baseline Choice

The chosen evaluation metrics depend on the baseline image used. An ideal baseline is uninformative, leads to a stable evaluation, and is consistent across various settings, such as the predicted classes. GradCAM is chosen as a reference explanation method, even though the results could be method-specific.

The following image baselines were chosen for comparison: black image, per-channel mean input image mean (per-image), blurred image, and two darker blurred image variants (divided by 4 and 8). The blurred images are created by a flat low-pass convolution filter with the kernel size of 29. The blur level has been found empirically and confirmed to drop the prediction score below chance level.

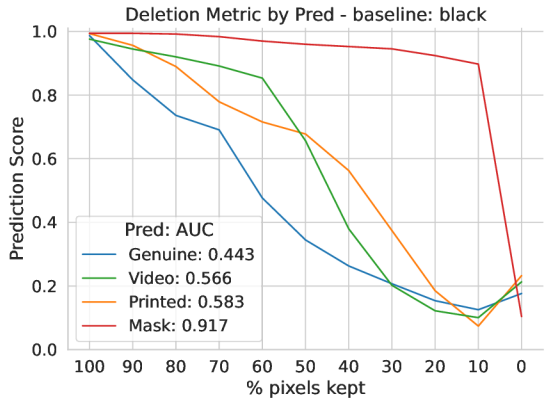
The darker blurred variants have been added when trying to improve upon the blurred baseline results. The results are shown in Figures 6.3 and 6.4. The results show that the baseline choice is not important for the insertion metric. In the deletion metric, the per-class results are never consistent. The black baseline is the only one to reach an approximately neutral prediction with the whole of the image removed. The baselines' deletion metric AUC means and standard deviations are shown in the Table 6.3. These results also turn out inconclusive, as low mean AUC is reached by inconsistently performing baselines (high standard deviation).

In conclusion, the black baseline is chosen for further use, as it gives the most consistent results.

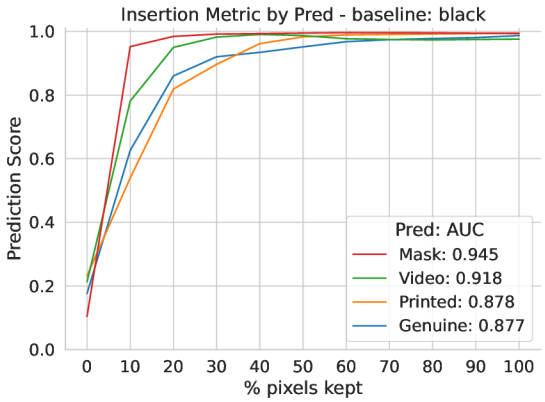
6.3.2 Method Choice

All explanation generation methods have been evaluated by the deletion and insertion metrics, using the selected black baseline image. Figures 6.5 and 6.6 show a comparison of the deletion and insertion metric.

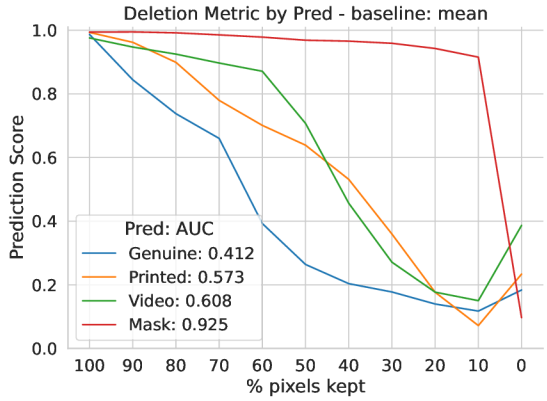
The deletion metric results show that XGradCAM and EigenCAM fail to surpass even model-independent baselines. Among the remaining methods, results do not differ drastically. A surprising result is the RandomCAM, which achieves a higher in the early removal steps than any explanation method. The deletion curves confirm that it is difficult to re-



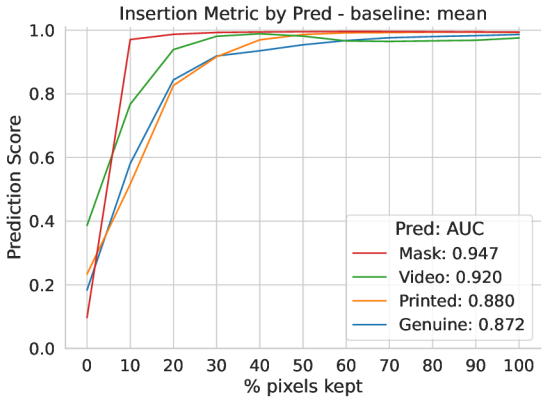
(a) Baseline black, Deletion metric



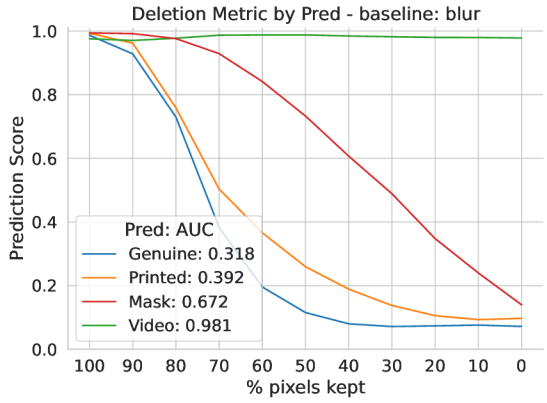
(b) Baseline black, Insertion metric



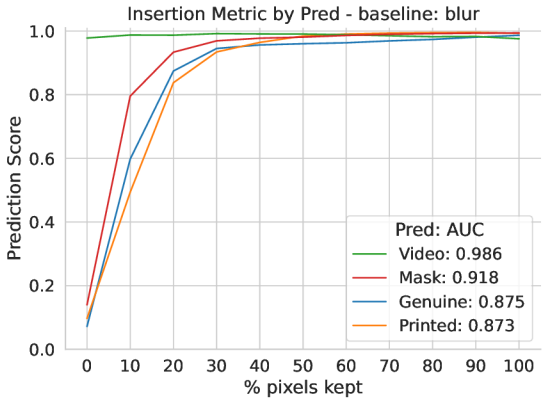
(c) Baseline mean, Deletion metric



(d) Baseline mean, Insertion metric

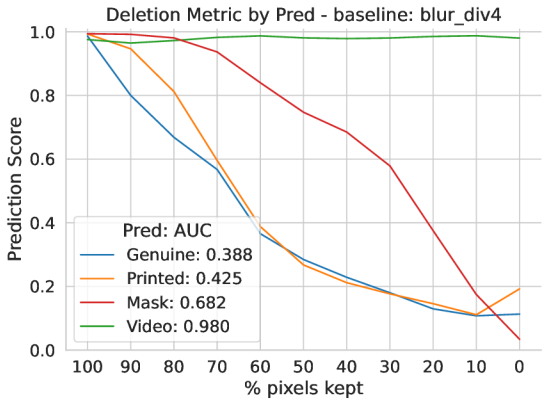


(e) Baseline blur, Deletion metric

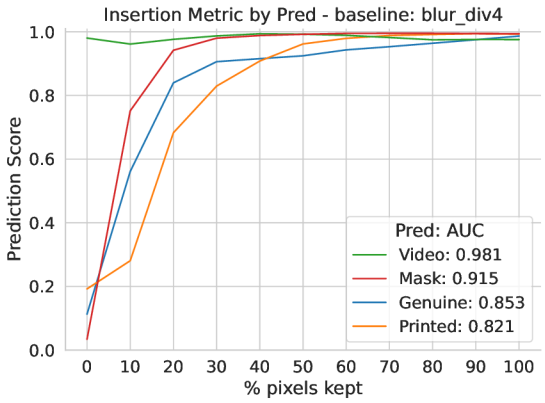


(f) Baseline blur, Insertion metric

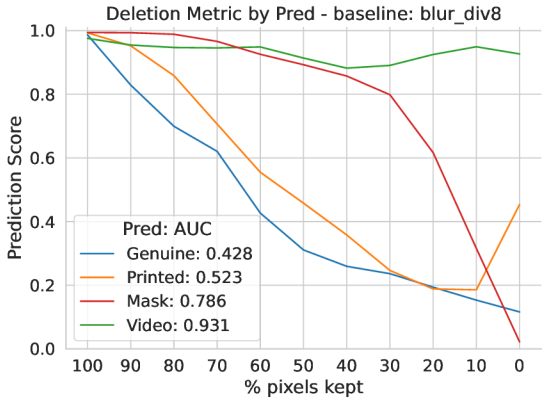
Figure 6.3: Evaluation of the perturbation baselines (1,2,3 of 5), deletion and insertion metric comparison across predicted classes, reporting AUC for each, Grad-CAM explanations used.



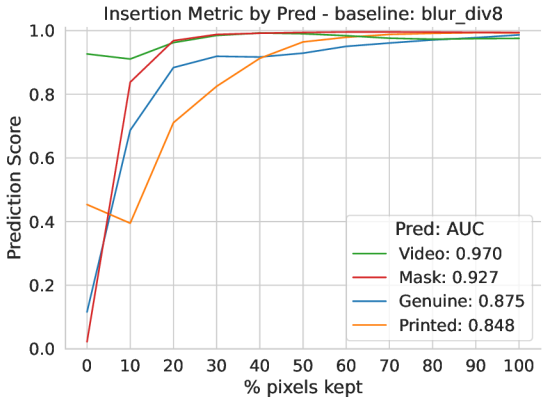
(a) Baseline blur_div4, Deletion metric



(b) Baseline blur_div4, Insertion metric



(c) Baseline blur_div8, Deletion metric



(d) Baseline blur_div8, Insertion metric

Figure 6.4: Evaluation of the perturbation baselines (4,5 of 5), deletion and insertion metric comparison across predicted classes, reporting AUC for each, GradCAM explanations used.

move information from the input data. While the test-time augmented GradCAM does not convincingly beat all other methods, it is chosen as the winner for further evaluations. It is important to note that in either of the figures, the GradCAM_aug model is hardly visible.

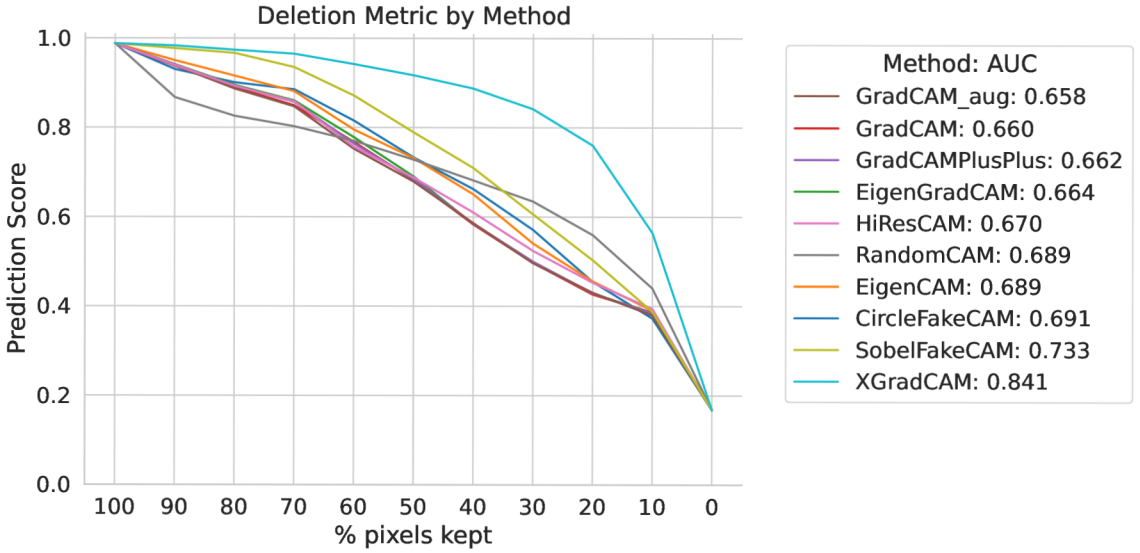


Figure 6.5: Deletion metric – comparison of selected explanation methods using the black baseline image. Area under deletion curve (lower is better) is shown in the legend.

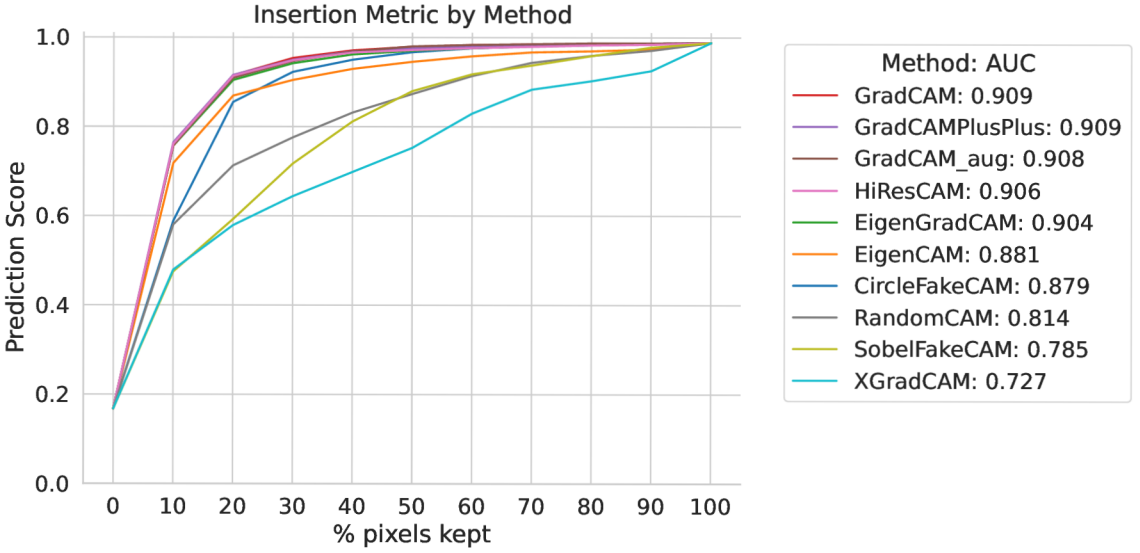


Figure 6.6: Insertion metric – comparison of selected explanation methods using the black baseline image. Area under deletion curve (higher is better) is shown in the legend.

6.3.3 Method Evaluation

To get an intuition for the resulting explanations shapes, average CAMs are generated. A confusion matrix of CAMs is shown in Figure 6.7 to showcase the common cues of a given predicted class, or specific errors made by the model. Each cell shows CAM explanations for the predicted class, for predictions where model actually predicted this class. No explanations are shown for the remaining classes that were not predicted. The results show very similar shapes for all correctly predicted subgroups. This is most likely due to the high model performance, resulting in averaging out any outliers. It also highlights the fact that the model mostly uses the face itself for making the prediction, as opposed to detecting external cues.

Since the incorrect prediction CAMs give more interesting results, the CAMs for the full dataset version have also been generated, while only keeping those of incorrect predictions. The results are shown in Figure 6.8.

6.3.4 Applying the Explanations

In this experiment, explanations that cause the highest prediction score dropoff were extracted. For the Top-EffNet model in the all-attacks mode, examples can be seen in Figure 6.9.

Deletion Score and Prediction Correctness

Some exploratory analyses of deletion metric scores were performed that compared AUC values between data subgroups. An interesting finding was that the AUC is significantly lower for incorrect model predictions. On the „single“ RoseYoutu test set, the deletion AUC is on average 0.664 for correctly classified samples and 0.178 for incorrectly classified. Similarly, the insertion metric is 0.912 for correctly classified samples and 0.600 for incorrectly classified. This suggests that a model's predictive performance could be improved through the stability of the prediction on perturbed samples, given a generated explanation.

More similar analyses were explored in the process_cam.py script, but did not make it into the text. However, the other explorations did not show such significant and surprising results.

6.3.5 Out-of-Distribution Check

Since the perturbations of the input images modify the input samples, the validity of the perturbed samples can be questioned [13]. This experiment aims to answer whether it is the case in the setting used.

Since the models are trained as softmax classifiers, they are not inherently able to assess this problem. Although the model output is a confidence score, softmax classifiers suffer from unreasonably confident predictions for out-of-distribution inputs [11, p. 14], and no confidence calibration measure is taken during training.

One way of analysing the data distribution through the optic of the model is to analyse the model embeddings. The EfficientNetV2s model used produces a 1280-dimensional embeddings, making similarity/distance metrics hard to intepret. Embeddings are thus projected by PCA and t-SNE, and analysed from their visualisation.

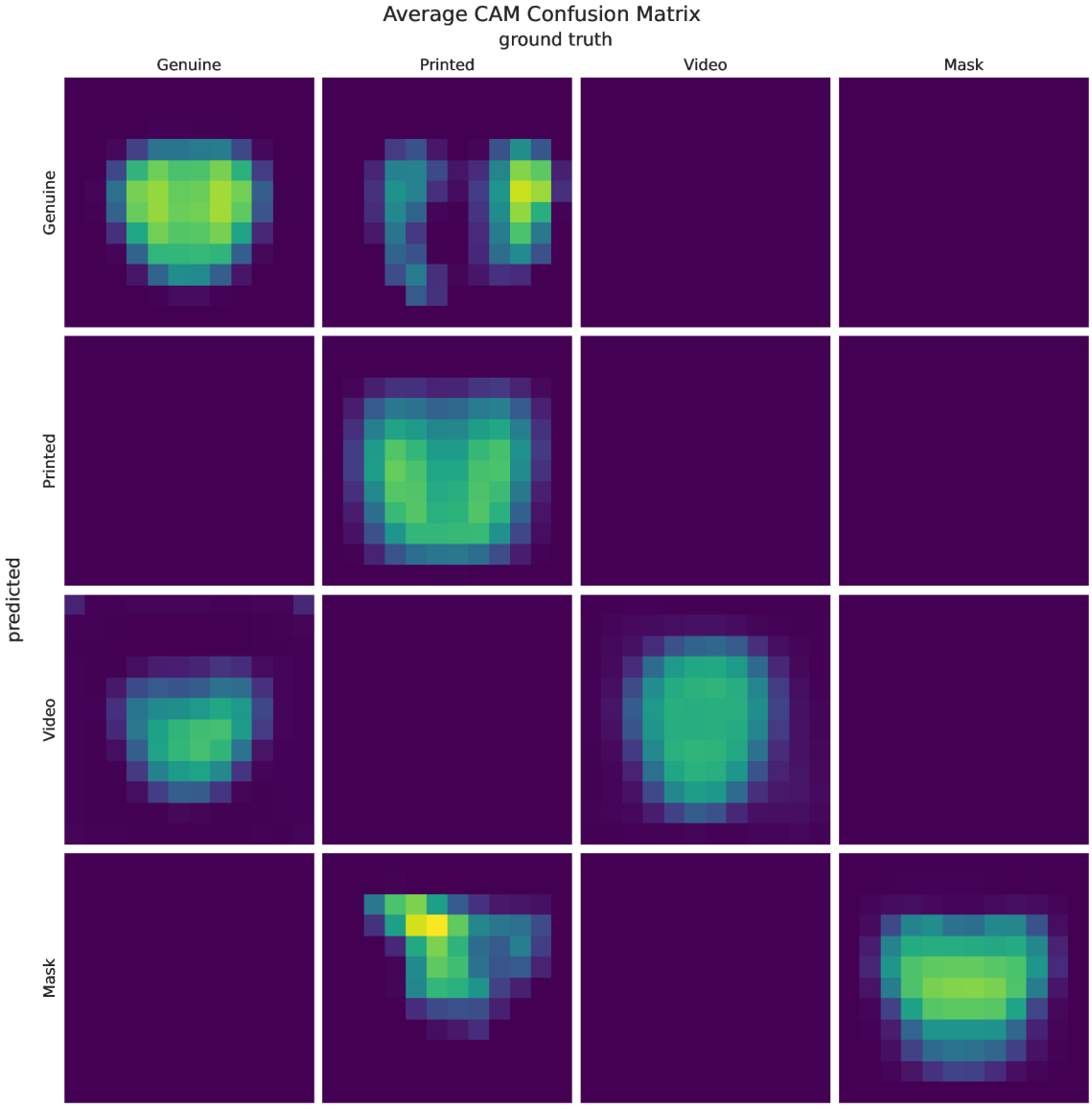


Figure 6.7: GradCAM augmented – Confusion Matrix of average CAMs. Values are not normalised. Missing cells indicate no data for the given combination of a predicted and ground truth class.

The input embeddings are produced by the Top-EffNet model trained in the all-attacks mode, and generated for 640 random samples from the test dataset split. The PCA and t-SNE projections of the perturbed image embeddings are shown in Figure 6.10.

The results show a comparatively smaller shift for the insertion metric, suggesting less of a distribution shift than in the deletion metric. An overlap in the original and slightly perturbed samples gives the impression that they come from the same distribution. Intra-class samples proximity also confirms this notion.

The visualisation also further supports the feature redundancy claims and justifies the use of the random erasing augmentations. In our own interpretation, the visualisation

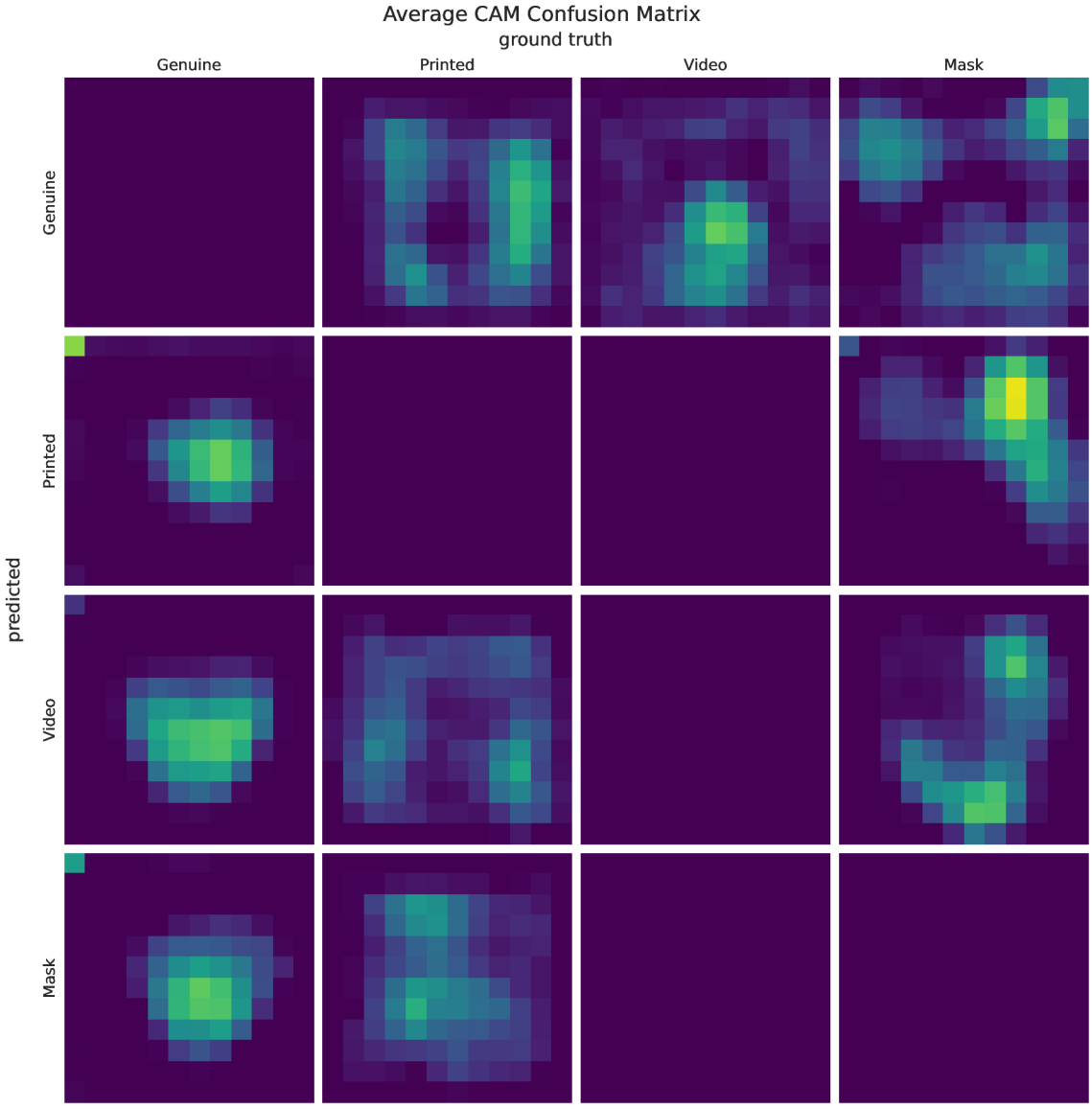
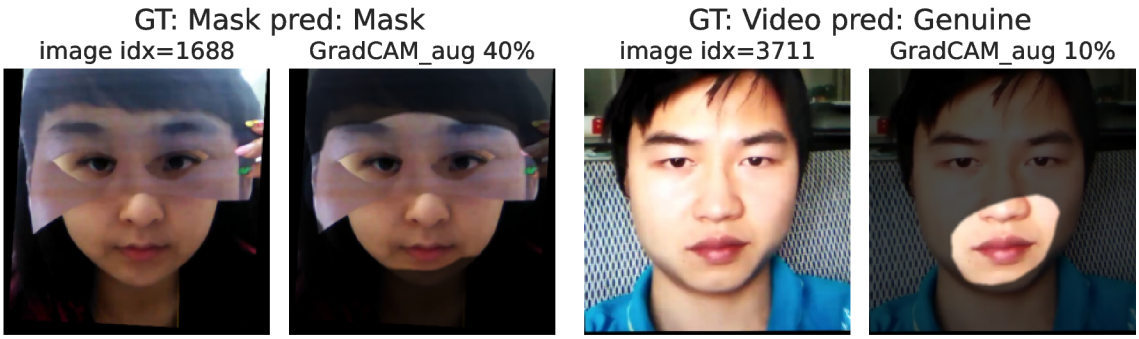


Figure 6.8: GradCAM augmented – Confusion Matrix of average CAMs. Full dataset used and only incorrect predictions were kept. Values are not normalised. Missing cells indicate no data for the given combination of a predicted and ground truth class.

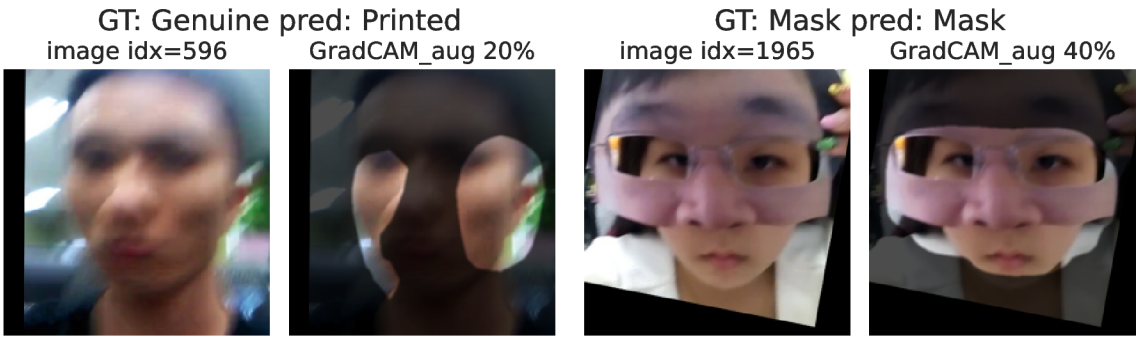
aligns with the fact that the models can recognise the images based on the little presence of features in the input, given the right feature selection (as seen in the insertion metric curves). Since explanations are often located in the image centre, the insertion metric images are also more likely to present a coherent image (illustrated, for example, by the 50% perturbation level for the deletion and insertion metric in Figure 4.2)

It is worth noting that the inner workings of the methods must be taken into account to avoid drawing false conclusions. The continuous character of the PCA projections is expected, as is the clustering of the t-SNE projections. For t-SNE, a perplexity of 30 was



(a) Dropoff explanation 1

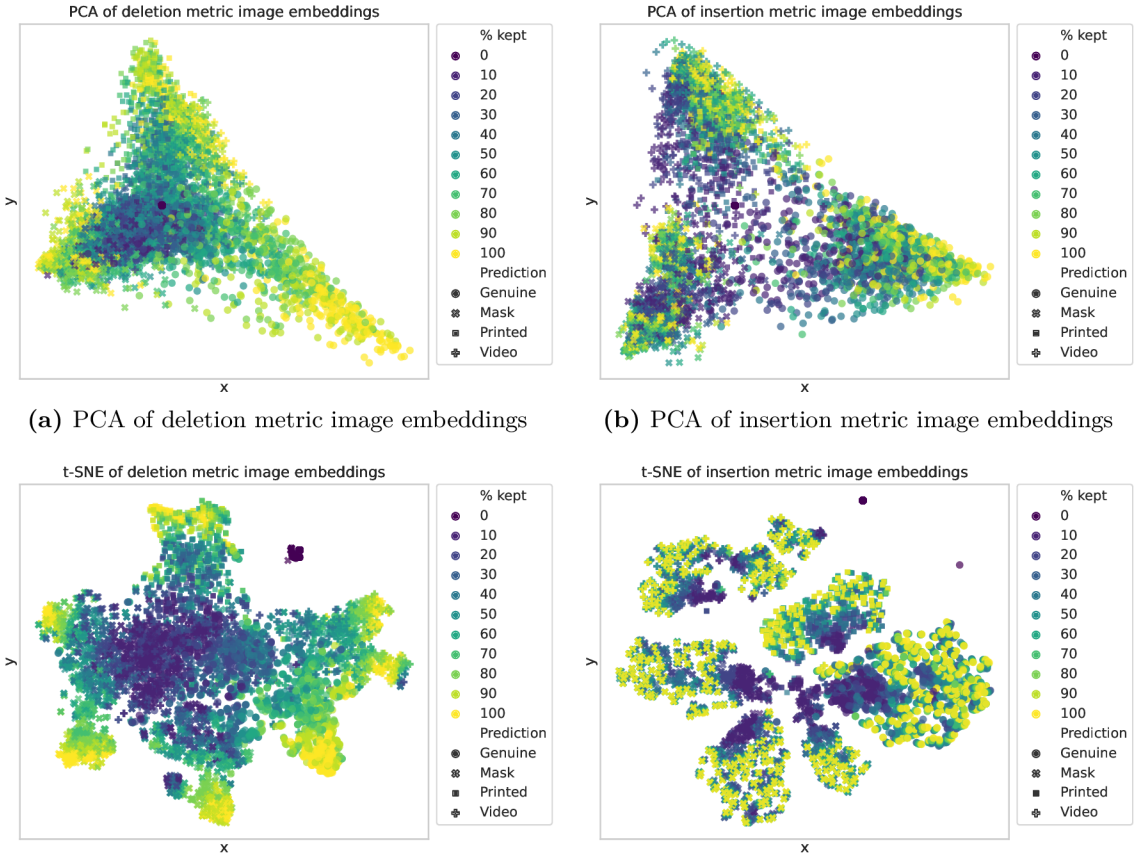
(b) Dropoff explanation 2



(c) Dropoff explanation 3

(d) Dropoff explanation 4

Figure 6.9: Explanations that cause a high prediction score dropoff even with low percentage of input image replaced.



(a) PCA of deletion metric image embeddings

(b) PCA of insertion metric image embeddings

(c) t-SNE of deletion metric image embeddings

(d) t-SNE of insertion metric image embeddings

Figure 6.10: PCA and t-SNE projections of perturbed image embeddings for 640 random test images and perturbations calculated based on GradCAM_aug explanations. The projections show a distribution shift for the highly-perturbed samples. However, they confirm that low-intensity perturbations are still not too far from the original distribution.

used, which is the default choice in the Scikit-Learn package, and it was assumed to be a safe choice without further verification. At the same time, a related work [13] concludes through a visually similar projection generated by the UMAP technique that the perturbed samples are out-of-distribution.

6.3.6 Discussion

The experiments have shown that the predictive performance of the developed face PAD models is very high in the context used. Lower performance is expected for unseen attacks, as the usage of the data set makes it harder in comparison to how it is used in similar works [34]. Experiments with unseen and one-attack models were planned and partially carried out, but a proper comparison was not achieved in time. For the results achieved, an ablation study assesses the decisions taken to train the models, showing that even simpler solutions could have worked as effectively.

Regarding the explainability methods, many methods were evaluated and care was taken to evaluate them fairly. Although the results did not clearly prove the chosen Grad-

CAM_aug method and the black baseline image to be ideal, they were comparatively better than the other methods. More experiments were performed with the method, and its suitability was further studied through the embeddings projections analysis. Explanations were also applied to show model decisions. The interpretation of these was not quite clear in the end, and in general, this work opens up more questions than it manages to answer. In the open and sometimes self-contradicting field of XAI, these results are not a failure, but a light shone on possibilities for future studies. Finally, explainability methods can also be used to evaluate the stability of the prediction in the face of perturbations as an additional assessment of the prediction correctness. This is an interesting point to be studied in the future.

Chapter 7

Conclusion

The goal of this Master's thesis was to explore, apply, and experiment with explainability methods in the domain of face presentation attack detection. This thesis presents the topics of face presentation attack detection and explainable machine learning. The methods in both fields were reviewed and their suitability was assessed. Relevant datasets were obtained and put to use in practise. A PAD model was trained, and layer attribution methods (such as GradCAM) were applied to it. Many methods similar to GradCAM were compared, and the augmented GradCAM was determined to be the best one. In the evaluation through devised perturbation metrics, GradCAM reaches an AUC of 0.658 for the deletion metric, and an AUC of 0.908 for the insertion metric. Then, explainability methods were in turn used to assess the models.

Through the optic of the perturbation metrics, explanations and model predictions on perturbed samples are studied. The suitability of the perturbation metrics is studied in detail, evaluating the choice of their associated baseline images, performing sanity checks, and studying the samples distributions. The behaviour of the prediction performance w.r.t. various data groups is explored. An ablation study analyses the impact of model training aspects.

More experiments were planned to analyse behaviour across models' training mode, but these did not make it into the evaluation, and provide an easy future direction to explore further. The use of additional explanation techniques (such as LIME, whose explanations are already implemented, but did not make it into the experiments), and use of more datasets (e.g., SIW-M, which was also prepared for experiments, but then never used in the end). A more comprehensive evaluation could always be performed, adding more evaluation methods, testing the behaviour for more models, etc. Explainable face presentation attack detection is still an open problem, and this thesis hopes to have explored paths that will benefit future endeavours.

Bibliography

[1] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I. J., Hardt, M. et al. Sanity Checks for Saliency Maps. In: *Neural Information Processing Systems*. 2018. Available at: https://api.semanticscholar.org/CorpusID:52938797.

[2] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R. et al. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*. Public Library of Science. july 2015, vol. 10, no. 7, p. 1–46. DOI: 10.1371/journal.pone.0130140. Available at: https://doi.org/10.1371/journal.pone.0130140.

[3] Barredo Arrieta, A., Díaz Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S. et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*. 2020, vol. 58, p. 82–115. DOI: https://doi.org/10.1016/j.inffus.2019.12.012. ISSN 1566-2535. Available at: https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[4] Baweja, Y., Oza, P., Perera, P. and Patel, V. M. *Anomaly Detection-Based Unknown Face Presentation Attack Detection*. arXiv, 2020. DOI: 10.48550/ARXIV.2007.05856. Available at: https://arxiv.org/abs/2007.05856.

[5] Chattopadhay, A., Sarkar, A., Howlader, P. and Balasubramanian, V. N. Grad-CAM++: Generalized Gradient-Based Visual Explanations for Deep Convolutional Networks. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar 2018. DOI: 10.1109/wacv.2018.00097. Available at: https://doi.org/10.1109%2Fwacv.2018.00097.

[6] Chen, C., Li, O., Tao, C., Barnett, A. J., Su, J. et al. This Looks Like That: Deep Learning for Interpretable Image Recognition. arXiv. 2018. DOI: 10.48550/ARXIV.1806.10574. Available at: https://arxiv.org/abs/1806.10574.

[7] Desai, S. and Ramaswamy, H. G. Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization. In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, p. 972–980. DOI: 10.1109/WACV45572.2020.9093360.

[8] Dombrowski, A.-K., Alber, M., Anders, C. J., Ackermann, M., Müller, K.-R. et al. Explanations can be manipulated and geometry is to blame. In: *Neural Information Processing Systems*. 2019. Available at: https://api.semanticscholar.org/CorpusID:195069127.

[9] Draelos, R. L. and Carin, L. *Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks*. arXiv, 2020. DOI: 10.48550/ARXIV.2011.08891. Available at: https://arxiv.org/abs/2011.08891.

[10] Fu, R., Hu, Q., Dong, X., Guo, Y., Gao, Y. et al. *Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs*. arXiv, 2020. DOI: 10.48550/ARXIV.2008.02312. Available at: https://arxiv.org/abs/2008.02312.

[11] Gal, Y. *Uncertainty in Deep Learning*. 2016. Dissertation. University of Cambridge. Available at: https://mlg.eng.cam.ac.uk/yarin/thesis/thesis.pdf.

[12] Gildenblat, J. and contributors. *PyTorch library for CAM methods*. GitHub, 2021. Available at: https://github.com/jacobgil/pytorch-grad-cam.

[13] Gomez, T., Fréour, T. and Mouchère, H. Metrics for Saliency Map Evaluation of Deep Learning Explanation Methods. In: El Yacoubi, M., Granger, E., Yuen, P. C., Pal, U. and Vincent, N., ed. *Pattern Recognition and Artificial Intelligence*. Cham: Springer International Publishing, 2022, p. 84–95. ISBN 978-3-031-09037-0.

[14] Goodfellow, I., Bengio, Y. and Courville, A. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[15] He, K., Zhang, X., Ren, S. and Sun, J. Deep Residual Learning for Image Recognition. *CoRR*. 2015, abs/1512.03385. Available at: http://arxiv.org/abs/1512.03385.

[16] Hedström, A., Weber, L., Bareeva, D., Krakowczyk, D., Motzkus, F. et al. *Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations and Beyond*. 2023.

[17] Hooker, S., Erhan, D., Kindermans, P.-J. and Kim, B. *A Benchmark for Interpretability Methods in Deep Neural Networks*. arXiv, 2018. DOI: 10.48550/ARXIV.1806.10758. Available at: https://arxiv.org/abs/1806.10758.

[18] Huber, M., Fang, M., Boutros, F. and Damer, N. Are Explainability Tools Gender Biased? A Case Study on Face Presentation Attack Detection. *ArXiv*. 2023, abs/2304.13419.

[19] Jiang, P.-T., Zhang, C.-B., Hou, Q., Cheng, M.-M. and Wei, Y. LayerCAM: Exploring Hierarchical Class Activation Maps for Localization. *IEEE Transactions on Image Processing*. 2021, vol. 30, p. 5875–5888. DOI: 10.1109/TIP.2021.3089943.

[20] Kapishnikov, A., Bolukbasi, T., Vi'egas, F. and Terry, M. XRAI: Better Attributions Through Regions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, p. 4947–4956. Available at: https://api.semanticscholar.org/CorpusID:201125339.

[21] Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B. et al. Captum: A unified and generic model interpretability library for PyTorch. *CoRR*. 2020, abs/2009.07896. Available at: https://arxiv.org/abs/2009.07896.

[22] Krishna, S., Han, T., Gu, A., Pombra, J., Jabbari, S. et al. *The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective*. arXiv, 2022. DOI: 10.48550/ARXIV.2202.01602. Available at: https://arxiv.org/abs/2202.01602.

[23] Li, H., Li, W., Cao, H., Wang, S., Huang, F. et al. Unsupervised Domain Adaptation for Face Anti-Spoofing. *IEEE Transactions on Information Forensics and Security*. 2018, vol. 13, no. 7, p. 1794–1809. DOI: 10.1109/TIFS.2018.2801312.

[24] Lin, Y.-S., Liu, Z.-Y., Chen, Y.-A., Wang, Y.-S., Chang, Y.-L. et al. XCos: An Explainable Cosine Metric for Face Verification Task. *ACM Trans. Multimedia Comput. Commun. Appl.* New York, NY, USA: Association for Computing Machinery. nov 2021, vol. 17, 3s. DOI: 10.1145/3469288. ISSN 1551-6857. Available at: https://doi.org/10.1145/3469288.

[25] Liu*, Y., Jourabloo*, A. and Liu, X. Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision. In: *In Proceeding of IEEE Computer Vision and Pattern Recognition*. Salt Lake City, UT: [b.n.], June 2018.

[26] Lu, Y., Xu, Z. and Ebrahimi, T. *Towards Visual Saliency Explanations of Face Verification*. 2023.

[27] Maaten, L. van der and Hinton, G. E. Visualizing Data using t-SNE. *Journal of Machine Learning Research*. 2008, vol. 9, p. 2579–2605. Available at: https://api.semanticscholar.org/CorpusID:5855042.

[28] Molnar, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2nd ed. 2022. Available at: https://christophm.github.io/interpretable-ml-book.

[29] Muhammad, M. B. and Yeasin, M. Eigen-CAM: Class Activation Map using Principal Components. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Jul 2020. DOI: 10.1109/ijcnn48605.2020.9206626. Available at: https://doi.org/10.1109%2Fijcnn48605.2020.9206626.

[30] Müller, S. G. and Hutter, F. TrivialAugment: Tuning-free Yet State-of-the-Art Data Augmentation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, p. 754–762.

[31] Murdoch, W. J., Singh, C., Kumbier, K., Abbasi Asl, R. and Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*. Proceedings of the National Academy of Sciences. oct 2019, vol. 116, no. 44, p. 22071–22080. DOI: 10.1073/pnas.1900654116. Available at: https://doi.org/10.1073%2Fpnas.1900654116.

[32] Nauta, M., Trienes, J., Pathak, S., Nguyen, E., Peters, M. et al. From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI. *ACM Computing Surveys*. 2022. Available at: https://api.semanticscholar.org/CorpusID:246063780.

[33] Neto, P. C., Gonçalves, T., Pinto, J. R., Silva, W., Sequeira, A. F. et al. *Explainable Biometrics in the Age of Deep Learning*. arXiv, 2022. DOI: 10.48550/ARXIV.2208.09500. Available at: https://arxiv.org/abs/2208.09500.

[34] Neto, P. C., Sequeira, A. F. and Cardoso, J. S. Myope Models - Are face presentation attack detection models short-sighted? *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*. 2021, p. 390–399.

[35] Nóbrega, M. *Explainable and Interpretable Face Presentation Attack Detection Methods*. Porto, Portugal, 2021. Master's Thesis. Available at: https://hdl.handle.net/10216/139294.

[36] Petsiuk, V., Das, A. and Saenko, K. RISE: Randomized Input Sampling for Explanation of Black-box Models. *ArXiv*. 2018, abs/1806.07421. Available at: https://api.semanticscholar.org/CorpusID:49324724.

[37] Rebuffi, S.-A., Fong, R., Ji, X. and Vedaldi, A. There and Back Again: Revisiting Backpropagation Saliency Methods. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, p. 8836–8845. Available at: https://api.semanticscholar.org/CorpusID:214802359.

[38] Ribeiro, M. T., Singh, S. and Guestrin, C. „Why Should I Trust You?“: Explaining the Predictions of Any Classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 2016, p. 1135–1144.

[39] Rong, Y., Leemann, T., Borisov, V., Kasneci, G. and Kasneci, E. *A Consistent and Efficient Evaluation Strategy for Attribution Methods*. arXiv, 2022. DOI: 10.48550/ARXIV.2202.00449. Available at: https://arxiv.org/abs/2202.00449.

[40] Selvaraju, R. R., Das, A., Vedantam, R., Cogswell, M., Parikh, D. et al. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. *CoRR*. 2016, abs/1610.02391. Available at: http://arxiv.org/abs/1610.02391.

[41] Sequeira, A. F., Gonçalves, T., Silva, W., Pinto, J. R. and Cardoso, J. S. An exploratory study of interpretability for face presentation attack detection. *IET Biometrics*. 2021, vol. 10, no. 4, p. 441–455. DOI: https://doi.org/10.1049/bme2.12045. Available at: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/bme2.12045.

[42] Shao, R., Lan, X. and Yuen, P. C. Regularized Fine-grained Meta Face Anti-spoofing. In: *AAAI Conference on Artificial Intelligence*. 2019. Available at: https://api.semanticscholar.org/CorpusID:208267535.

[43] Shrikumar, A., Greenside, P. and Kundaje, A. Learning Important Features Through Propagating Activation Differences. In: *International Conference on Machine Learning*. 2017. Available at: https://api.semanticscholar.org/CorpusID:3385018.

[44] Srinivas, S. and Fleuret, F. *Full-Gradient Representation for Neural Network Visualization*. arXiv, 2019. DOI: 10.48550/ARXIV.1905.00780. Available at: https://arxiv.org/abs/1905.00780.

[45] Sundararajan, M., Taly, A. and Yan, Q. Axiomatic Attribution for Deep Networks. In: *International Conference on Machine Learning*. 2017. Available at: https://api.semanticscholar.org/CorpusID:16747630.

[46] Tan, M. and Le, Q. V. EfficientNetV2: Smaller Models and Faster Training. arXiv. 2021. DOI: 10.48550/ARXIV.2104.00298. Available at: https://arxiv.org/abs/2104.00298.

[47] Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z. et al. Score-CAM: Score-weighted visual explanations for convolutional neural networks. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 2020, p. 24–25.

[48] Wang, Y.-C., Wang, C.-Y. and Lai, S.-H. Disentangled Representation with Dual-stage Feature Learning for Face Anti-spoofing. *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2021, p. 1234–1243. Available at: https://api.semanticscholar.org/CorpusID:239016916.

[49] Wattenberg, M., Viégas, F. and Johnson, I. How to Use t-SNE Effectively. *Distill*. 2016. DOI: 10.23915/distill.00002. Available at: http://distill.pub/2016/misread-tsne.

[50] Xiao, G., Yaojie, L., Anil, J. and Liu, X. Multi-domain Learning for Updating Face Anti-spoofing Models. In: *In Proceeding of European Conference on Computer Vision (ECCV 2022)*. 2022.

[51] Yang, X., Luo, W., Bao, L., Gao, Y., Gong, D. et al. Face Anti-Spoofing: Model Matters, so Does Data. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, p. 3502–3511. DOI: 10.1109/CVPR.2019.00362.

[52] Yin, B., Tran, L., Li, H., Shen, X. and Liu, X. Towards Interpretable Face Recognition. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. October 2019. DOI: 10.48550/ARXIV.1805.00611. Available at: https://arxiv.org/abs/1805.00611.

[53] Zafar, M. R. and Khan, N. M. DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems. *ArXiv*. 2019, abs/1906.10263. Available at: https://api.semanticscholar.org/CorpusID:195584114.

[54] Zhang, K., Zhang, Z., Li, Z. and Qiao, Y. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *CoRR*. 2016, abs/1604.02878. Available at: http://arxiv.org/abs/1604.02878.

[55] Zhou, B., Khosla, A., Lapedriza, À., Oliva, A. and Torralba, A. Learning Deep Features for Discriminative Localization. *CoRR*. 2015, abs/1512.04150. Available at: http://arxiv.org/abs/1512.04150.