Light Direction Reconstruction Analysis and Improvement using XAI and CG

Markus Miller[1]
markus.miller@hm.edu
http://orcid.org/0000-0001-9571-5997

Stefan Ronczka[1, 3]
stefan.ronczka@hm.edu
http://orcid.org/0000-0001-6738-9127

Alfred Nischwitz[1]
nischwitz@cs.hm.edu
http://orcid.org/0000-0003-3826-5584

Rüdiger Westermann[2]
westermann@tum.de
http://orcid.org/0000-0002-3394-0731

[1] *Dept. of Computer Science and Mathematics, University of Applied Sciences Munich, Lothstr. 64, D-80335, Munich, Bavaria*

[2] *Chair of Computer Graphics and Visualization, Technical University Munich, Boltzmannstr. 3/II, D-85748, Garching, Bavaria*

[3] *w&co MediaServices GmbH & Co KG, Charles-de-Gaulle-Str. 8, D-81737, Munich, Bavaria*

ABSTRACT

With rapid advances in the field of deep learning, explainable artificial intelligence (XAI) methods were introduced to gain insight into internal procedures of deep neural networks. Information gathered by XAI methods can help to identify shortcomings in network architectures and image datasets. Recent studies, however, advise to handle XAI interpretations with care, as they can be unreliable. Due to this unreliability, this study uses meta information that is produced when applying XAI to enhance the architecture – and thus the prediction performance – of a recently published regression model. This model aimed to contribute to solving the photometric registration problem in the field of augmented reality by regressing the dominant light direction in a scene. Bypassing misleading XAI interpretations, the influence of synthetic training data, generated with different rendering techniques, is furthermore evaluated empirically. In conclusion, this study demonstrates how the prediction performance of the recently published model can be increased by improving the network architecture and training dataset.

Keywords

Light, direction, estimation, reconstruction, explainable AI, photometric, registration, deep learning.

1 INTRODUCTION

After the tremendous progress of deep learning (DL) research in the past decade, gathering information on how deep neural networks (DNNs) make decisions from given input is gaining importance, as it may help to identify weaknesses and flaws in datasets or network architectures. This specific knowledge constitutes the foundation of certification processes in security- or safety-critical applications.

Therefore, in the past years, several explainable artificial intelligence (XAI) methods to achieve a human-comprehensible explanation of a DNN's decision process have been introduced, such as class activation mapping (CAM), gradient-weighted CAM (GradCAM), local interpretable model-agnostic explanations (LIME) and layer-wise relevance propagation (LRP). Both CAM (Zhou et al., 2016) and GradCAM (Selvaraju et al., 2020) result in heat maps showing what is considered important in an input image by a network for a specific inferencing task. However, CAM requires changes to a network's architecture, which renders this adjusted network incomparable to its previous architecture. LIME (Ribeiro et al., 2016) uses sampling masks to manipulate elements in the input images so that the influence of a specific element on the output function of a network can be estimated. As a sampling-based approach, LIME requires high-resolution masks to yield meaningful results when investigating filigree feature structures in the input images. LRP (Bach et al., 2015) propagates the relevance from the output all the way back to the input layer and generates a relevance map, highlighting the pixels in the input images that contributed most to a network's output decision. Most approaches deploy XAI methods to investigate classification problems. In an approach to count leaves of plants (Dobrescu et al., 2019), LRP is used to investigate how the number of counted leaves is derived from a given photograph.

Applying these XAI methods, we present an analysis of the reconstruction process of our DNN Net$_{s_x,s_y}$, which was proposed in an earlier publication (Miller et al., 2021) to predict the dominant light direction of a scene in stereographic coordinates, and derive architectural adjustments from it. We further investi-

https://www.doi.org/10.24132/CSRN.3201.24

gate the influence of computer graphics (CG) rendering techniques used in synthetic training data, such as different shading models (Blinn, 1977; Cook and Torrance, 1981; Lambert, 1760; Oren and Nayar, 1994) and shadow algorithms (Boksansky et al., 2019; Fernando, 2005; Williams, 1978), on the reconstruction performance and derive recommendations to generate synthetic training data.

The main contributions of this work can be summarized as follows:

- Insights on how to improve the reconstruction results of our DNN Net_{s_x,s_y} when regressing the dominant light direction from real scene images using XAI.
- Reduction of the reconstruction error by optimising the net architecture.
- Reduction of the reconstruction error by optimising the training dataset.

2 RELATED WORK

Previous work (Miller et al., 2021) showed that the dominant light direction of a real scene can be estimated more accurately from red-green-blue (RGB) images by using a stereographic coordinate representation of the dominant light direction, resulting in the stereographically predicting neural network Net_{s_x,s_y}. This network, trained on synthetically generated images, achieved an average angular reconstruction error $\overline{E_\measuredangle}$ of 3.7° on synthetic reference test data T_{SYN}, as well as 25.5° on real reference test data T_{REAL}, which could be improved to 7.1° by training Net_{s_x,s_y} on mixed data (99.2 percent synthetic and 0.8 percent real training images). The higher reconstruction error on real test data was assumed to be caused by a notable domain gap between synthetic and real datasets.

One possibility to improve the reconstruction performance achieved by Net_{s_x,s_y} is to analyse its decision process using an XAI model (such as LRP or GradCAM) and derive architecture optimisations from any insight gained.

LRP (Bach et al., 2015), when applied to a DNN, propagates the relevance, which is the contribution of a pixel or hidden neuron to the predicted output value, layer by layer from the output layer back to the input layer. Between two consecutive layers, relevance distribution is associated with the connections between each layer's neurons, following a given distribution rule. The distribution rule determines how relevance, i. e. positive or negative contributions to the regression result, or combinations of both, is being propagated, which may allow the importance of specific features in an input image to be investigated.

To analyse a given layer, as a first step, GradCAM (Selvaraju et al., 2020) computes weighted feature maps by scaling each feature map in the layer with its average gradient. Those weighted feature maps are then combined into a layer saliency map. By upscaling the layer saliency map to the input resolution and overlaying the input image with it, sensitive regions in the input image can be highlighted. Unlike LRP, GradCAM analyses one layer at a time.

Sanity checks (Adebayo et al., 2018) were introduced to gain intuition on how reliable explanations of different XAI methods may be by applying randomisation tests for both model parameters, as well as data labels and comparing the changes in the produced saliency maps. According to Adebayo et al. (2018), visual inspection of explanations alone may result in misleading conclusions. An extending study (Sixt et al., 2020) concludes that the gradient of most back-propagation based XAI approaches, such as LRP with certain relevance distribution rules, converges to a rank-1 matrix, which is why saliency maps of those approaches tend to highlight features of rather shallow network layers, not sufficiently showing decisions in deeper layers. Hence, despite the benefits XAI methods may provide, they also need to be handled with care.

Though not referring to this, in an approach to count the leaves of a plant (Dobrescu et al., 2019), LRP is applied to analyse a VGG-16 DNN, similar to Net_{s_x,s_y}, when regressing the number of leaves in a given image. By investigating the features extracted by the convolutional section and other experiments, such as manually covering leaves, it was concluded that the investigated DNN has indeed learned to regress the number of leaves from actual depictions of leaves. However, the content displayed in the investigated features was unhesitantly accepted, disregarding a potential unreliability.

Another possibility to improve the reconstruction performance is to optimise the dataset used for training, in particular the synthetic dataset, by investigating the influence of CG rendering techniques on the reconstruction result and tailoring a well-performing training dataset.

When optimising the training datasets to predict illumination situations, CG rendering techniques responsible for illumination and shadows are most relevant. Most basic illumination is achieved by applying the Lambertian reflection model (Lambert, 1760), which creates diffusely illuminated surfaces. The Phong-Blinn illumination model (Blinn, 1977) adds specular highlights to Lambertian reflecting surfaces by incorporating a half-way vector between light and view direction into the illumination computation. Extending the Lambertian reflection model, a more realistic diffuse illumination was achieved by assuming surfaces consisted of microfacets (Oren and Nayar, 1994), modelling surface roughness with a probability function. By taking physical models for refraction, roughness and self-shadowing into account, specular reflections could be improved to appear more realistic (Cook and Torrance, 1981). Since local illumination models disregard global phenomena like shadows, shadow mapping (Williams, 1978) introduced the ability to add hard shadows to a CG scene, independent from the used illumination model, by comparing computed depth values to values sampled from a previously generated depth or shadow map. The percentage closer soft shadows (PCSS) approach (Fernando, 2005) introduced soft shadows with variable penumbra by taking the distance between the shaded surface and blocker into account. When hardware acceleration for ray tracing became widely available, algorithms (Boksansky et al., 2019) for both hard and soft ray traced shadows could be incorporated into real-time applications using APIs, such as Vulkan or NVidia OptiX (Parker et al., 2010).

The presented study uses available XAI methods and tailored datasets to further improve the prediction performance achieved by Net$_{s_x,s_y}$.

3 APPLYING EXPLAINABLE AI

The architecture of Net$_{s_x,s_y}$ (Fig. 1) inherits the convolutional section of the VGG-16 (Simonyan and Zisserman, 2014) architecture, initialized with ImageNet (Russakovsky et al., 2015) pre-trained weights. Convolutional block C_5 was unlocked for training to adjust to the regression task, so its weights have changed. C_5 is followed by a custom fully connected (FC) block, consisting of a single FC layer L_h and a linear output layer L_o. L_h contains 4,096 neurons, is activated by a rectified linear unit (ReLU) function and uses a dropout value of 0.25. L_o consists of two output neurons without an activation function to regress stereographic coordinates s_x and s_y, representing the dominant light direction. Net$_{s_x,s_y}$ was trained using Adam as the optimizer, a batch size of 32 and a uniform learning rate of $1\mathrm{e}-3$[1].

In order to improve the prediction results of Net$_{s_x,s_y}$, a deeper understanding of its internal function and decision process may be helpful. However, additional architecture elements, as required by CAM, might notably change the reconstruction performance of Net$_{s_x,s_y}$, as well as the net itself, which is why CAM cannot be reasonably applied. Initial evaluation of LIME indicated that a fine-grained sampling mask would be required to produce meaningful explanations, which requires impractically high computing time even on high-performance computers.

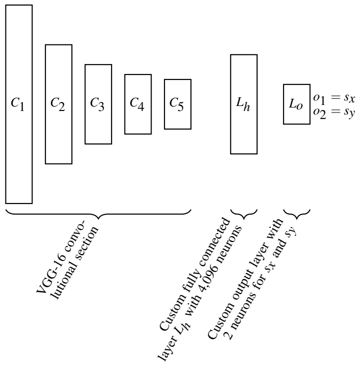

C_1
C_2
C_3
C_4
C_5
L_h
L_o
$o_1 = s_x$
$o_2 = s_y$
VGG-16 convo-lutional section
Custom fully connected layer L_h with 4,096 neurons
Custom output layer with 2 neurons for s_x and s_y

Figure 1: Diagram of the architecture of Net$_{s_x,s_y}$.

LRP and GradCAM neither require high computing power, nor changes to the architecture when investigating a network. Thus, LRP and GradCAM are applied to analyse Net$_{s_x,s_y}$.

Ideally, XAI methods yield easily interpretable saliency maps, identifying distinct image regions in the input, such as surface shading or shadow edges. Those image regions, when changed, may significantly influence the reconstruction performance. Consequently, to investigate positive and negative contributions to the regression result, when applying LRP, the relevance is distributed with the $\alpha\beta$-rule (Bach et al., 2015)

$$R_i = \sum_j \left(\alpha \cdot \frac{(a_i w_{ij})^+}{\sum_i (a_i w_{ij})^+} - \beta \cdot \frac{(a_i w_{ij})^-}{\sum_i (a_i w_{ij})^-} \right) R_j \quad (1)$$

with $\alpha = 1$ and $\beta = 0$ for positive, or $\alpha = 0$ and $\beta = 1$ for negative contributions, respectively. When propagating the relevance back through a network using the $\alpha\beta$-rule, the relevance R_i of a particular neuron i in the target layer is computed by proportionally summing up the relevance R_j of each neuron j in the source layer that it is connected to. The proportion to which each R_j contributes to R_i is given as the quotient of the connection contribution, i. e. the product of i's activation value a_i and the connection weight w_{ij} between neurons i and j, and the sum over all connection contributions between neuron j and any neuron in the target layer. This proportion is then separated into a positive and negative partial sum, indicated by superscript plus sign and minus sign, respectively. However, distributing the relevance with the $\alpha\beta$-rule causes LRP to converge to a rank-1 matrix (Sixt et al., 2020) and thus may not reliably provide insight into the decision process of Net$_{s_x,s_y}$. GradCAM as a gradient-based XAI method may provide valid insights, as it is not affected by this issue.

Interpreting image regions highlighted by the saliency maps of LRP or GradCAM remains difficult, nonetheless, as Net$_{s_x,s_y}$ does not predict discrete classes, but continuous values of s_x and s_y, denoting the dominant

[1] Given as uniform learning rate of 1 in Miller et al. (2021), which means 0.001 when using Adam as the optimizer.

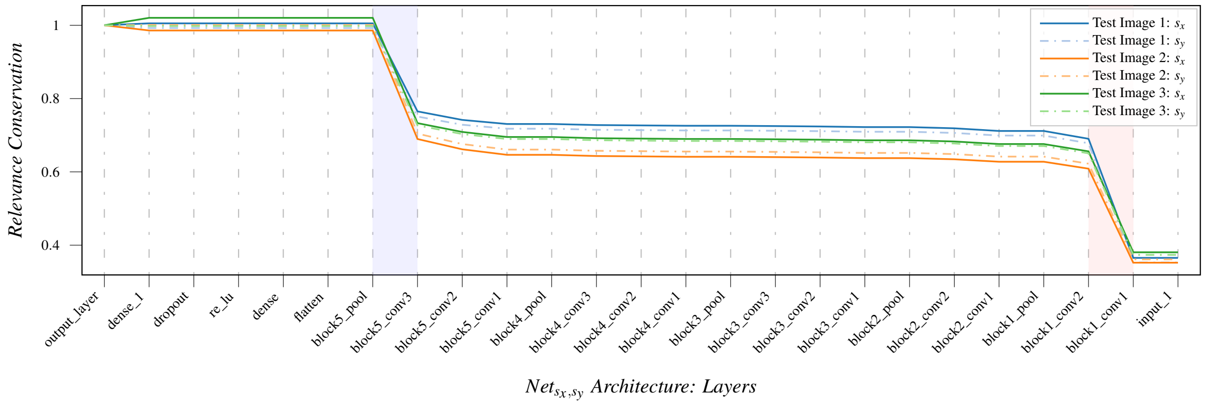


Figure 2: Relevance conservation of mixed trained Net_{s_x,s_y} for s_x (solid) and s_y (dash-dotted) neurons. Steep changes between layers indicate loss of relevance (highlighted), most likely due to the bias (note: the red highlight is inherent to the pre-trained VGG-16).

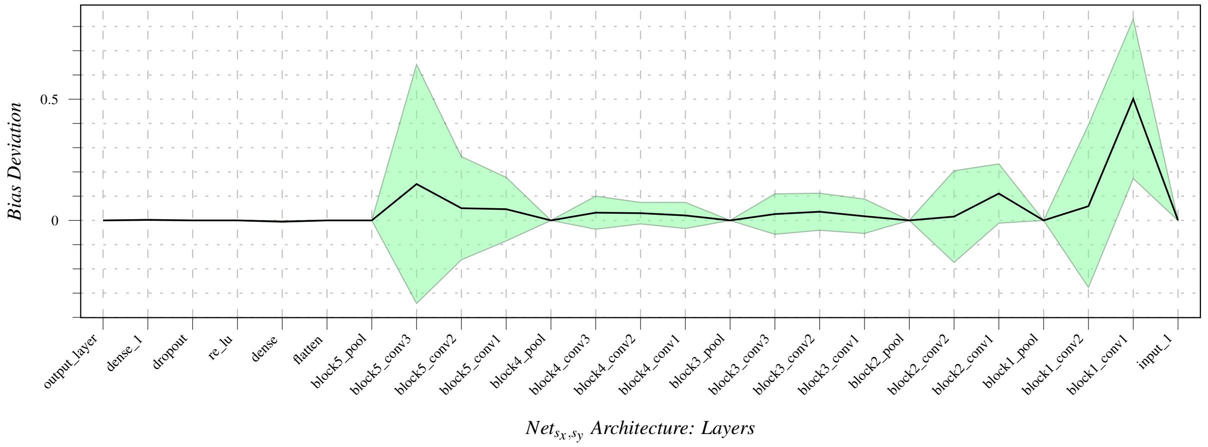


Figure 3: Bias deviation of mixed trained Net_{s_x,s_y}. The high bias deviation in *block5_conv3* may suggest the reason for the occurring relevance loss.

light direction in the scene. Different from classification, no statistical accumulation of saliency maps, which relates to a certain class, is formed due to the continuous values. Therefore, deriving statistical information, which image regions are important for a certain prediction, from saliency maps of single images, which constitute merely a momentary snapshot, cannot be considered reliable. A statistical evaluation, indicating how significant particular image regions are associated to a regression result, might accumulate saliency maps over a series of input images with same predictions and thus identify important image regions. However, this is most likely bound to fail as well due to the scene diversity (meaning different objects, light directions and camera directions) depicted in the images and the spatial variation of the image regions, leading to a map with scattered accumulated highlights not containing helpful information.

So, how can XAI be applied to analyse Net_{s_x,s_y} as a regression model? During the calculation of XAI methods, meta information is generated, such as relevance conservation when applying LRP, which can be used to gain intuition about the inner structure of a network. Relevance conservation means that relevance is assumed to be constant across the layers of a DNN and is a constraint of LRP computation. Inside a layer, the relevance is distributed to the bias and all neurons. However, only the relevance distributed to the latter is propagated to the next layer. Hence, jumps in a relevance conservation plot indicate a bias-heavy decision contribution in the affected layer. Initially, the convolutional section of Net_{s_x,s_y} was assumed to extract relevant features from the input image, and the FC block would connect this feature information into knowledge about the illumination situation. Analysing the relevance conservation of Net_{s_x,s_y} (Fig. 2), however, indicates a significant loss of relevance, particularly between the third convolutional and pooling layer of C_5 (*block5_conv3* and *block5_pool*), suggesting the weighted sum that is passed to the activation function is significantly influenced by bias. This is further supported by analysing the bias deviation in each layer (Fig. 3), showing a high bias deviation in *block5_conv3*. The bias deviation is analysed by plotting the deviation of bias values from the average bias value in each layer. The bias value of single bias layers, such as FC layers, are captured by the average bias value in that layer, denoting the actual bias value. Interpreting the graphs, the convolutional section is not only extracting features, but also appears to pre-select important features in *block5_conv3* through bias values. Each feature map of *block5_conv3* maintains a dedicated bias value, which may suppress or dampen a feature map with negative or small bias values in favour of feature maps with a significant positive bias value when passed to the ReLU activation function. Due to this pre-selection, it appears that the FC block merely combines already existing information into regression values without providing new knowledge in that sense, leading to the hypothesis that the FC block may be replaceable with a linear layer. This hypothesis is investigated by replacing the FC block of Net$_{s_x,s_y}$ with a single linear layer with two output neurons. The changed architecture (Fig. 4) is called linear feature aggregation network (LFAN).

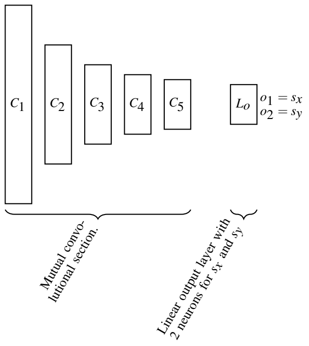

C_1
C_2
C_3
C_4
C_5
Lo
$o_1 = sx$
$o_2 = sy$
Mutual convo-lutional section.
Linear output layer with 2 neurons for sx and sy

Figure 4: Diagram of the LFAN architecture.

When analysing the regression of the amount of leaves in a given image, Dobrescu et al. (2019) interpreted the contents of features extracted by the convolutional section of their VGG-16 and concluded their DNN would count leaves in given images using intended information, though disregarding research (Adebayo et al., 2018; Sixt et al., 2020) recommending to handle results of XAI methods based on back-propagation with care. However, extracted features, computed by forward-propagating an image through the convolutional section of Net$_{s_x,s_y}$, choosing a particular feature map and propagating its relevance back through the net, may contain a different form of meta information, such as certain regions a feature map is sensitive to in the input image (Fig. 6). Seemingly, the convolutional section of Net$_{s_x,s_y}$ extracts abstract chunks as features, containing small sections of the input scene. These abstract chunks show combinations of unrelated image content, such as partial shadow edges or fractions of shaded surfaces. When trying to predict the dominant light direction of a scene, considering a wide or even global area of the input scene may be beneficial, leading to the hypothesis that a deeper convolutional section may improve the regression result. This hypothesis is investigated by using a fully convolutional network (FCN), which adds two additional convolutional blocks, each consisting of one convolutional and one max-pooling layer, to the convolutional section and completes the network with a single linear layer with two output neurons (Fig. 5).

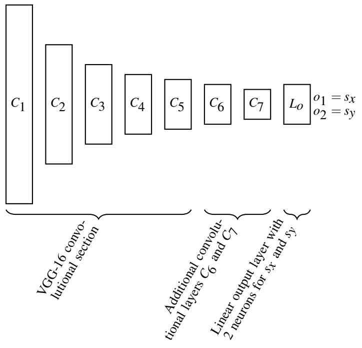

C_1
C_2
C_3
C_4
C_5
C_6
C_7
Lo
$o_1 = sx$
$o_2 = sy$
VGG-16 convo-lutional section
Additional convolu-tional layers C6 and C7
Linear output layer with 2 neurons for sx and sy

Figure 5: Diagram of the FCN architecture.

Analysing the extracted features this way does not rely on a subjective interpretation of feature content, but on neutral metrics, such as the receptive field of a certain feature map.

3.1 Training LFANs and FCNs

Since LFANs and FCNs share architecture elements with Net$_{s_x,s_y}$, synthetically and mixed pre-trained weights are available to be used for transfer learning. Hence, the same training process used for Net$_{s_x,s_y}$ applies, meaning the training is split into hyperparameter-tuning (where needed) and fine-tuning. During hyperparameter-tuning, the convolutional section inherited from VGG-16 is disabled for training. Then, convolutional block C_5 is enabled for training during fine-tuning and the learning rate is reduced to a thousandth. Due to the FC block only being replaced by a single linear layer, LFAN training solely requires fine-tuning and is performed with the same hyperparameters used to train Net$_{s_x,s_y}$ (Section 3, mentioned optimizer, batch size and learning rate). Training the FCNs requires hyperparameter-tuning prior to fine-tuning in order to find suitable parameter values, such as the number of filters in both the first and second added convolutional layer, whether to use a bias in these layers, the optimizer to use, the learning rate and the batch size. To tune the hyperparameters, the Bayesian optimisation module of Keras Tuner[2]

[2] After a comparison of Keras Tuner (https://github.com/keras-team/keras-tuner), Optuna (Akiba et al., 2019) and Talos (http://github.com/autonomio/talos), Keras Tuner was selected due to its tuning performance compared to the required tuning time.

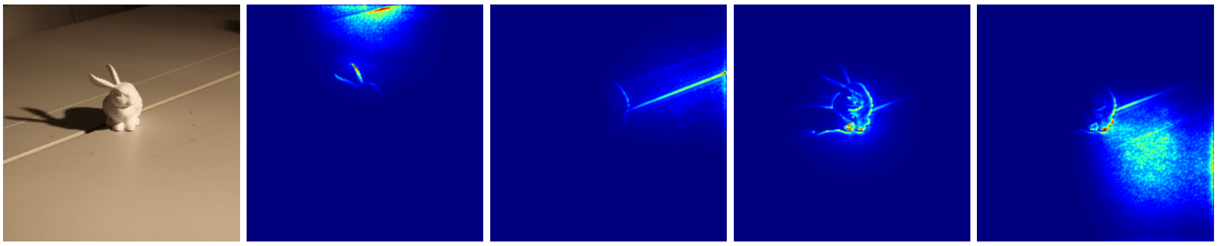


Figure 6: Given an input sample (left most), the convolutional section of mixed trained Net$_{s_x,s_y}$ extracts abstract feature chunk samples (remaining images).

Hyperparameter	FCN$_{\text{SYNTH}}$	FCN$_{\text{MIXED}}$
optimiser	Adam	RMSProp
learning rate	$1.5e-4$	$6.1e-5$
batch size	16	64
filters$_1$	682	1371
filters$_2$	1347	64
use_bias	false	false

Table 1: Best hyperparameter combinations for FCNs with synthetic (FCN$_{\text{SYNTH}}$) and mixed (FCN$_{\text{MIXED}}$) base weights of Net$_{s_x,s_y}$.

is used over 50 trials, training for five epochs each and identifying the best hyperparameter combinations (Table 1). Further, all trainings are conducted with the same datasets, synthetic and mixed, Miller et al. (2021) used to train Net$_{s_x,s_y}$ due to comparability. Additionally, the influence of data augmentation is investigated by conducting each training with enabled and disabled augmentation, i. e. the variation of image brightness, image translation and image zoom. All trained networks are predicting the dominant light direction in stereographic coordinates.

4 DATASETS

To investigate the influence of CG rendering techniques on the prediction performance of Net$_{s_x,s_y}$, DNNs using the same architecture and hyperparameters (Section 3, first paragraph) as Net$_{s_x,s_y}$ are trained on datasets with varying rendering techniques and tested on T$_{\text{SYN}}$ and T$_{\text{REAL}}$ (Section 2, first paragraph).

The datasets are generated with combinations of varying rendering techniques, each combination consisting of technique selections from three categories: diffuse reflections, specular reflections and shadows. Generating the datasets is realized in a dedicated application, using OpenGL and NVidia OptiX (Parker et al., 2010) and implementing rendering techniques for each category. As common CG rendering techniques for diffuse illumination, the Lambertian (Lambert, 1760) and Oren-Nayar (Oren and Nayar, 1994) reflectance models are implemented. Phong-Blinn (Blinn, 1977) and Cook-Torrance (Cook and Torrance, 1981) reflection models are implemented for specular reflections. Shadows of varying quality are evaluated by implementing plain shadow mapping (Williams, 1978) for hard shadows and PCSS (Fernando, 2005) for shadows with variable penumbra. When implementing PCSS, soft shadows are simulated by implementing percentage closer filtering (PCF), introduced by Reeves et al. (1987), applying a randomly rotated Poisson disc for each fragment when sampling the shadow map to avoid artefacts at the edges of the shadow. All shadow mapping techniques are implemented with an adaptive depth bias (Ehm et al., 2015) to avoid further shadow artefacts. Additionally to shadow mapping, ray tracing is used to render shadows. When implementing ray traced shadows, NVidia OptiX (Parker et al., 2010) is used, as de-noising is handled automatically by the framework. Both hard and soft ray traced shadows (Boksansky et al., 2019) are implemented by sending shadow rays from a possibly shaded surface location towards the light source. Since no real-time requirement applies to generate the datasets, the naive approaches are implemented. For hard shadows, caused by a point light source, one single shadow ray is cast; for soft shadows, up to 256 shadow rays are cast to randomly sample a spatially extended light source. Additionally, a naive approach for ambient occlusion (AO) with ray tracing (Nischwitz et al., 2019) is implemented by sampling the close proximity of a fragment with fixed length rays checking for hits with structures and reducing the light intensity proportionally.

Light directions and camera positions used in the generated datasets are similar[3] to the directions and positions used in the reference dataset presented by Miller et al. (2021) to train and test Net$_{s_x,s_y}$, ranging from $[0^\circ, 360^\circ[$ azimuth and $[5^\circ, 90^\circ]$ elevation in 5° steps each for the light direction distribution and in steps of 45° and 30° for azimuth and elevation of the camera positions, respectively.

To avoid an explosion of datasets and trainings required to be evaluated, the influence of the chosen reflection models is investigated systematically by starting out with shadowless combinations of Phong-Blinn and Lambert, Phong-Blinn and Oren-Nayar, and Cook-

[3] Identical, except for the starting value of the elevation angle, which is 5° instead of 1° due to visibility. Next iteration of camera positions is 30° instead of 35°.

Torrance and Lambert, varying the surface roughness with discrete values of 0.25, 0.5, 0.75 and 1.0 for combinations with either Oren-Nayar or Cook-Torrance. Two additional combinations using Cook-Torrance and Oren-Nayar with a roughness of 0.5 and 0.75 are being investigated, after the first evaluation, resulting in a total of 11 datasets with different illumination. Each of the illumination datasets is then rendered with a shadow algorithm: hard shadows with shadow mapping, hard shadows with ray tracing, soft shadows with PCSS, ray traced soft shadows with 256 shadow rays and ray traced soft shadows with ray traced AO (both sampled with 256 rays). This way, 55 datasets with illumination and shadows, as well as 11 datasets with illumination but without shadow, are generated and investigated, resulting in a total of 66 different datasets. The three datasets that perform the best on $\mathrm{T}_{\mathrm{REAL}}$ are eventually mixed with the same small fraction of real data used in the mixed data to train Net_{s_x,s_y} and evaluated on $\mathrm{T}_{\mathrm{SYN}}$ and $\mathrm{T}_{\mathrm{REAL}}$ to determine the dataset with best performance overall.

5 RESULTS

To remain comparable to results achieved by Net_{s_x,s_y}, all introduced network architectures were tested with the same synthetic and real reference test dataset as used in Miller et al. (2021), aforementioned as $\mathrm{T}_{\mathrm{SYN}}$ and $\mathrm{T}_{\mathrm{REAL}}$.

The reference network Net_{s_x,s_y}, trained with mixed data, achieved an average angular error $\overline{E_\measuredangle}$ of 7.1° on $\mathrm{T}_{\mathrm{REAL}}$. When being trained on synthetic data, Net_{s_x,s_y} achieved an error of $\overline{E_\measuredangle} = 3.7°$ on $\mathrm{T}_{\mathrm{SYN}}$ and 25.5° on $\mathrm{T}_{\mathrm{REAL}}$, respectively. No data was given for Net_{s_x,s_y} being tested on synthetic data and trained with mixed data (Table 2).

	$\mathrm{T}_{\mathrm{SYN}}$	$\mathrm{T}_{\mathrm{REAL}}$
SYNTH TRAINED	3.7°	25.5°
MIXED TRAINED	n/a	7.1°

Table 2: Net_{s_x,s_y} reference results.

Each architecture variant, LFAN and FCN, was trained on different base weights, i. e. the inherited convolutional section was initialized before training with different pre-trained weights: the weights of both the synthetically and mixed trained Net_{s_x,s_y}, and weights of an ImageNet pre-trained VGG-16. Using ImageNet pretrained base weights did not show any improvement and thus are not displayed.

Investigating the LFAN architecture performance (Table 3), initializing the convolutional section with base weights of the synthetically trained Net_{s_x,s_y} improved the average angular error $\overline{E_\measuredangle}$ on $\mathrm{T}_{\mathrm{REAL}}$ from 25.5° (previously achieved by the synthetically trained Net_{s_x,s_y}) to 22.8° when using synthetic data to train the LFAN.

	$\mathrm{T}_{\mathrm{SYN}}$	$\mathrm{T}_{\mathrm{REAL}}$	Base	Augm.
SYNTH TRAINED	1.6°	22.8°	SYNTH Net_{s_x,s_y}	NO
SYNTH TRAINED	1.7°	6.8°	MIXED Net_{s_x,s_y}	NO
MIXED TRAINED	2.4°	18.3°	SYNTH Net_{s_x,s_y}	NO
MIXED TRAINED	2.4°	6.2°	MIXED Net_{s_x,s_y}	NO

Table 3: LFAN evaluation results. Entries in the column *Augm.* indicate whether data augmentation was used.

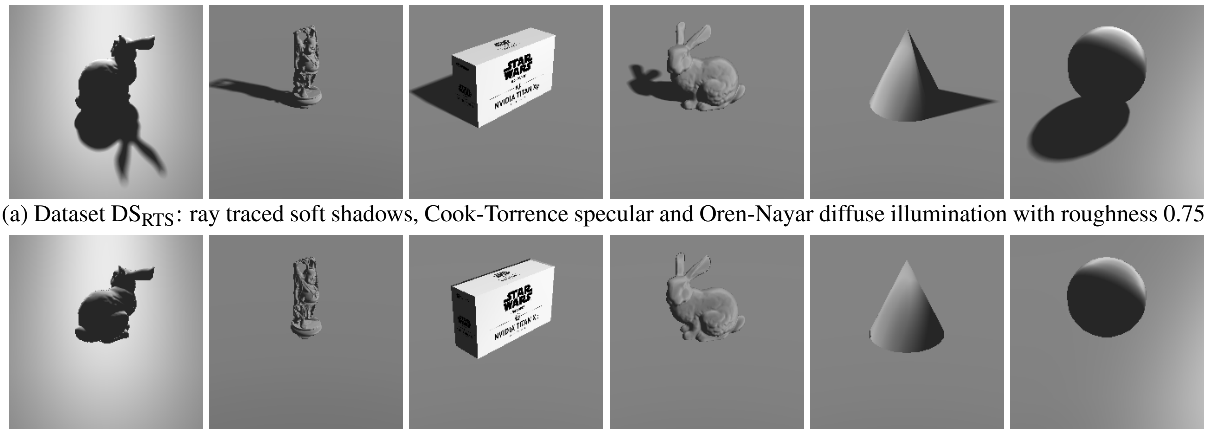
	$\mathrm{T}_{\mathrm{SYN}}$	$\mathrm{T}_{\mathrm{REAL}}$	Base	Augm.
SYNTH TRAINED	1.4°	32.4°	SYNTH Net_{s_x,s_y}	NO
SYNTH TRAINED	1.4°	6.5°	MIXED Net_{s_x,s_y}	NO
MIXED TRAINED	1.3°	7.3°	SYNTH Net_{s_x,s_y}	NO
MIXED TRAINED	1.4°	5.7°	MIXED Net_{s_x,s_y}	NO

Table 4: FCN evaluation results. Again, entries in the column *Augm.* indicate whether data augmentation was used.

This could be further improved to 18.3° by training the LFAN with mixed data. On $\mathrm{T}_{\mathrm{SYN}}$, these synthetically based LFANs achieved an error of 1.6° using synthetic and 2.4° with mixed training data. Using the base weights of mixed trained Net_{s_x,s_y}, the average angular error $\overline{E_\measuredangle} = 7.1°$ on $\mathrm{T}_{\mathrm{REAL}}$ (achieved by mixed trained Net_{s_x,s_y}) could be improved to 6.8° when training the LFAN with synthetic data and to 6.2° with mixed training data. When tested on $\mathrm{T}_{\mathrm{SYN}}$, similar values as before with 1.7° with synthetic and 2.4° with mixed training data are achieved.

Compared to the synthetically trained Net_{s_x,s_y}, the prediction performance of the FCN architecture improves to $\overline{E_\measuredangle} = 1.4°$ from 3.7° on $\mathrm{T}_{\mathrm{SYN}}$ using synthetic training data and synthetic base weights, but decreases from 25.5° to 32.4° on $\mathrm{T}_{\mathrm{REAL}}$ (Table 4). Using synthetic base weights and mixed training data, similar behaviour is shown, as the prediction performance improves to 1.3° on $\mathrm{T}_{\mathrm{SYN}}$ and declines to 7.3° on $\mathrm{T}_{\mathrm{REAL}}$ from 7.1° originally. Contrary to that, when using mixed base weights, the FCN architecture achieves 1.4° on $\mathrm{T}_{\mathrm{SYN}}$ with both synthetic and mixed training data. Also, using mixed base weights improves the prediction performance of this FCN variant with synthetic training data to 6.5° and with mixed training data to 5.7° on $\mathrm{T}_{\mathrm{REAL}}$.

After investigating the various datasets, the dataset $\mathrm{DS}_{\mathrm{RTS}}$ (Fig. 7a) using Cook-Torrance as specular, Oren-Nayar as diffuse reflection model and ray traced soft shadows without AO performed the best. Though not containing any shadow, dataset $\mathrm{DS}_{\mathrm{NoS}}$ (Fig. 7b) using Cook-Torrance and Oren-Nayar without AO performed second best. To distinguish the models, the DNNs trained with $\mathrm{DS}_{\mathrm{RTS}}$ and $\mathrm{DS}_{\mathrm{NoS}}$ are named $\mathrm{Net}_{\mathrm{RTS}}$ and $\mathrm{Net}_{\mathrm{NoS}}$, respectively, though they use the same architecture and hyperparameters as Net_{s_x,s_y}, as well as ImageNet pre-trained weights. When trained solely on synthetic $\mathrm{DS}_{\mathrm{RTS}}$ images, $\mathrm{Net}_{\mathrm{RTS}}$ achieves an average angular error of $\overline{E_\measuredangle} = 20.2°$ on $\mathrm{T}_{\mathrm{REAL}}$ and 35.5° on $\mathrm{T}_{\mathrm{SYN}}$. Mixing $\mathrm{DS}_{\mathrm{RTS}}$ with the small real training set



(a) Dataset DS_{RTS}: ray traced soft shadows, Cook-Torrence specular and Oren-Nayar diffuse illumination with roughness 0.75.

(b) Dataset DS_{NoS}: No shadows, Cook-Torrance specular and Oren-Nayar diffuse illumination with roughness 1.0.

Figure 7: Dataset samples of the two best performing datasets.

used by Miller et al. (2021), the thereby mixed trained Net_{RTS} achieves an error of 4.4° on T_{REAL} and 23.4° on T_{SYN}, indicating a significant domain gap between DS_{RTS} and T_{SYN} (Table 5). Similar behaviour is shown by DS_{NoS}, though accuracy is worse overall compared to DS_{RTS}. Net_{NoS}, when trained solely with synthetic images of DS_{NoS}, achieves an angular error of 20.2° on T_{REAL} and 35.5° on T_{SYN}. Adding the small real training images to DS_{NoS} and training Net_{NoS} with this mixed dataset achieves an error of 5.7° on T_{REAL} and 38.5° on T_{SYN}, again indicating a significant domain gap between DS_{NoS} and T_{SYN}.

	T_{SYN}	T_{REAL}	Augm.
SYNTH Net_{NoS}	35.5°	20.2°	YES
SYNTH Net_{RTS}	32.1°	20.2°	YES
MIXED Net_{NoS}	38.5°	5.7°	YES
MIXED Net_{RTS}	23.4°	4.4°	YES

Table 5: Evaluation results of the dataset investigation. Prefixes in front of the network names indicate the used training dataset. Entries in the column *Augm.* indicate, whether data augmentation was used.

It is noteworthy that both LFAN and FCN architectures perform best exclusively without data augmentation, whereas DNN architectures with FC elements performed best with data augmentation enabled.

In summary, the previous state of the art with an error of 7.1° (mixed trained Net_{s_x,s_y}) on T_{REAL} is improved to 6.2° with a mixed trained LFAN. It could be further improved to 5.7° with a mixed trained FCN. Both DNN architectures were pre-trained with mixed trained Net_{s_x,s_y} base weights. Mixed trained Net_{RTS} with an error of 4.4° achieves the best performance (Table 6).

6 DISCUSSION

In conclusion, this study demonstrates successful use of XAI meta information to systematically improve the prediction performance of the recently published regression model Net_{s_x,s_y}, which predicts the dominant light direction in a given scene, from an average angular error $\overline{E_{\measuredangle}} = 7.1°$ to an error of 6.2° using the presented LFAN architecture. Eventually, the improvement goes down to an error of 5.7° with FCNs on real reference test data T_{REAL} by deriving architectural adjustments from aforementioned meta information.

	T_{REAL}
MIXED Net_{s_x,s_y}	7.1°
MIXED LFAN	6.2°
MIXED FCN	5.7°
MIXED Net_{RTS}	4.4°

Table 6: Summary of results gathered from tables 2 to 5.

An investigation of the influence of CG rendering techniques on the prediction result of Net_{s_x,s_y} reveals that the dataset rendered with techniques that most accurately approximate reality, i. e. Oren-Nayar for diffuse, Cook-Torrence for specular illumination and ray traced soft shadows without the naive AO implementation, achieved the best result with the mixed trained Net_{RTS} on the real reference test set T_{REAL}, achieving an average angular error $\overline{E_{\measuredangle}} = 4.4°$ and outperforming the mixed trained FCN using mixed base weights as best performing architecture adjusted DNN. Though DS_{NoS} does not contain shadows, the reconstruction performance of mixed trained Net_{NoS} is noteworthy, as this DNN may have learned to reduce the domain gap between DS_{NoS} and T_{REAL} from few training examples. Similar behaviour of Net_{s_x,s_y} is presumably shown on T_{REAL}, since the edge between the two tables (Fig. 6, left most image) appears to be extracted by the convolutional section as a distinctive feature (Fig. 6, image in the middle). However, this interpretation may be inaccurate and misleading due to the findings of Adebayo et al. (2018) and Sixt et al. (2020).

While analysing features in different layers is a common and reasonable approach to optimise the prediction results of a network, deriving conclusions from meta information, such as relevance conservation and bias analysis, as shown in this work, appears to be unprecedented, as other approaches, despite analysing the conservation of relevance across the layers of a network, do not derive information in a similar way as described in this work.

However, a major drawback of the LFAN and FCN architectures are their inherent lack of regularisation, such as dropout, and thus their inherent possibility to overfit, which becomes most likely apparent in the FCN variant using synthetic base weights of Net_{s_x,s_y} and synthetic training data, considering the angular average error $\overline{E}_{\measuredangle}$ of 1.4° on synthetic test data compared to an error of 32.4° on real test data. Additionally, the derived LFAN and FCN architectures are likely to be less robust when regressing from images with deviating brightness, as well as sufficiently non-centred or zoomed objects, as this appears to be too difficult when mapping the extracted features linearly to the output neurons. One indication for this is that both architectures perform worse when being trained with data augmentation enabled, which produces training images with according changes. Another indication is that, despite taking global features into account (Fig. 8), FCNs are affected, nonetheless.

A fundamental problem when applying LRP in the investigated situation occurs when investigating regressed values of $s_x, s_y = (0,0)$. Propagating a value of 0, meaning a relevance value $R_j = 0$ (eq. 1), would not yield a meaningful result, despite the fact that regressing values of $s_x, s_y = (0,0)$ are valid stereographic coordinates, denoting a light direction coming right from above in a given scene.

7 FUTURE WORK

Considering the improvements achieved by adding convolutional blocks to the FCN architecture, we intend to investigate whether applying attention-based DNNs may further improve the reconstruction performance.

Another opportunity for subsequent work is the investigation of possibilities to incorporate regularisation into the derived architectures and thus reduce the inherent potential to overfit. Furthermore, it is worth to investigate whether the FCN architecture may regain the ability to perform better when being trained with augmented data while maintaining its prediction performance when again adding a FC layer to map the extracted features to the output layer. Moreover, combining the FCNs architecture and further improvements to it with the datasets in this work may further improve the prediction results, too.

With larger real image data, containing more complex and diverse scenes, we will further investigate the robustness of the presented architectures.

Eventually, the presumed ability of Net_{s_x,s_y} to generalise on unknown data (Section 6, end of second paragraph) may be investigated by applying further XAI methods.

ACKNOWLEDGEMENTS

We would like to thank the Competence Center Image Processing of the University of Applied Sciences in Munich[4] with all their funding companies[5], in particular w&co MediaServices GmbH & Co KG, as well as the Chair of Computer Graphics and Visualization[6] of the Technical University Munich for allowing us to conduct this research.

We would further like to thank IBM for their support and help. IBM kindly granted us access to their OpenPOWER[7] deep learning system.

Eventually, we would like to thank Simon Neureither for his assistance in creating synthetic datasets with varying CG techniques and Conrad Smith for his help with questions regarding linguistic matters.

REFERENCES

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. (2018). Sanity checks for saliency maps. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 9525 – 9536, Red Hook, NY, USA. Curran Associates Inc.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2623 – 2631, New York, NY, USA. Association for Computing Machinery.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1 – 46.

Blinn, J. F. (1977). Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '77, pages 192 – 198, New York, NY, USA. Association for Computing Machinery.

[4] https://www.hm.edu/allgemein/forschung_entwicklung/forschungsfelder/competence_center/bildverarbeitung/index.de.html

[5] https://www.hm.edu/allgemein/forschung_entwicklung/forschungsfelder/competence_center/bildverarbeitung/partner.de.html

[6] https://www.in.tum.de/cg/

[7] https://openpower.ucc.in.tum.de

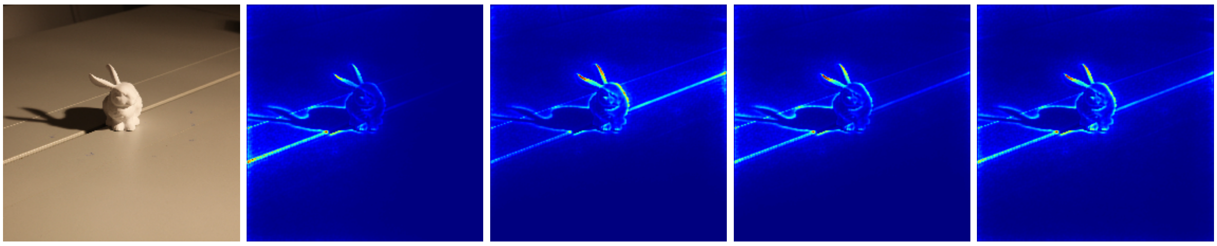


Figure 8: When investigating the extracted features from a given input image (left most), FCNs indeed take the entire image region into account (remaining images, all slightly different).

Boksansky, J., Wimmer, M., and Bittner, J. (2019). *Ray Traced Shadows: Maintaining Real-Time Frame Rates*, pages 159 – 182. Apress, Berkeley, CA.

Cook, R. L. and Torrance, K. E. (1981). A reflectance model for computer graphics. In *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '81, pages 307 – 316, New York, NY, USA. Association for Computing Machinery.

Dobrescu, A., Giuffrida, M. V., and Tsaftaris, S. A. (2019). Understanding deep neural networks for regression in leaf counting. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2600 – 2608.

Ehm, A., Ederer, A., Klein, A., and Nischwitz, A. (2015). Adaptive depth bias for soft shadows. In *23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: full papers proceedings*, pages 219 – 228. Computer Science Research Notes.

Fernando, R. (2005). Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, SIGGRAPH '05, page 35, New York, NY, USA. Association for Computing Machinery.

Lambert, J. H. (1760). *Photometria Sive De Mensura Et Gradibus Luminis, Colorum Et Umbrae*. Klett, Augsburg.

Miller, M., Nischwitz, A., and Westermann, R. (2021). Deep light direction reconstruction from single RGB images. In *29. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: full papers proceedings*, pages 31 – 40. Computer Science Research Notes.

Nischwitz, A., Fischer, M., Haberäcker, P., and Socher, G. (2019). Schatten. In *Computergrafik: Band I des Standardwerks Computergrafik und Bildverarbeitung*, pages 480 – 554, Wiesbaden. Springer Fachmedien Wiesbaden.

Oren, M. and Nayar, S. K. (1994). Generalization of lambert's reflectance model. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 239 – 246, New York, NY, USA. Association for Computing Machinery.

Parker, S. G., Bigler, J., Dietrich, A., Friedrich, H., Hoberock, J., Luebke, D., McAllister, D., McGuire, M., Morley, K., Robison, A., and Stich, M. (2010). Optix: A general purpose ray tracing engine. *ACM Trans. Graph.*, 29(4).

Reeves, W. T., Salesin, D. H., and Cook, R. L. (1987). Rendering antialiased shadows with depth maps. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 283 – 291, New York, NY, USA. Association for Computing Machinery.

Ribeiro, M., Singh, S., and Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 97 – 101, San Diego, California. Association for Computational Linguistics.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211 – 252.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2020). Grad-CAM: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336 – 359.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, volume abs/1409.1556.

Sixt, L., Granz, M., and Landgraf, T. (2020). When explanations lie: Why many modified BP attributions fail. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9046 – 9057. PMLR.

Williams, L. (1978). Casting curved shadows on curved surfaces. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '78, pages 270 – 274, New York, NY, USA. Association for Computing Machinery.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921 – 2929.