Centre for Modeling and Simulation
Savitribai Phule Pune University

Master of Technology (M.Tech.)
Programme in Modeling and Simulation

Project Report

Interpretations of Deep learning models for prediction of Intracellular skin diseases

Kunal Rathore
CMS1723

Academic Year 2017-19

Centre for Modeling and Simulation
Savitribai Phule Pune University

Certificate

This is certify that this report, titled

Interpretations of Deep learning models for prediction of Intracellular skin diseases,

authored by

Kunal Rathore (CMS1723),

describes the project work carried out by the author under our supervision during the period from January 2019 to June 2019. This work represents the project component of the Master of Technology (M.Tech.) Programme in Modeling and Simulation at the Center for Modeling and Simulation, Savitribai Phule Pune University.

Kshitij Deshmukh, Lead Project Engineer,
Persistent Systems, Pune, India

Bhushan Garware, Technical Expert
Persistent Systems, Pune, India

Jayaraman Valadi, Hon. Adjunct Professor
Centre for Modeling and Simulation
Savitribai Phule Pune University
Pune 411007 India

Mihir Arjunwadkar, Director
Centre for Modeling and Simulation
Savitribai Phule Pune University
Pune 411007 India

Centre for Modeling and Simulation
Savitribai Phule Pune University

Author's Declaration

This document, titled

Interpretations of Deep learning models for prediction of Intracellular skin diseases,

authored by me, is an authentic report of the project work carried out by me as part of the Master of Technology (M.Tech.) Programme in Modeling and Simulation at the Center for Modeling and Simulation, Savitribai Phule Pune University. In writing this report, I have taken reasonable and adequate care to ensure that material borrowed from sources such as books, research papers, internet, etc., is acknowledged as per accepted academic norms and practices in this regard. I have read and understood the University's policy on plagiarism (http://unipune.ac.in/administration_files/pdf/Plagiarism_Policy_University_14-5-12.pdf).

Kunal Rathore
CMS1723

Abstract

Machine Learning models have shown a remarkable predictive response as well as adoption in this era. But can you trust this model? Will it work after deployment? What else can it tell you about the world? Despite widespread use, machine learning models remain mostly black boxes. Besides this understand the reasons behind predictions is, quite important in building trust, particularly while choosing from a set of trained models. Such understanding also provides intrinsic properties of the model, which can be used to convert an untrustworthy model or prediction into a trustworthy one. However, the highest performance score for large datasets can be achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models. This complex model stretch distances between accuracy and interpretability. To address this objective we tried to implemented some well known methods that wre independent of the model viz. LIME, Shap and GradCAM on Intra-cellular skin disease image dataset and compared for better consistency with human intuitions.

Acknowledgements

I am indebted to **Prof. Jayaraman Valadi, Bhushan Garware** and **Kshitij Deshmukh** for their guidance on a daily basis during this project. I am thankful to Bhushan Garware for allowing me to work on this project topic. At the same time Kshtitj's guidance made me, think with different perspective to address this problem, his qualitative rather then quantitative approach taught me a lot. I am thank him to his cooperation and positive working nature.

I am really thankful to Prof. Jayaraman for their pushing effort to make us viable to work at well known and prosperous Company, also he understood our capabilities and motivated to publish paper at IEEE ICECCT 2019. I also thank Prasad Ovhal for his real hard work and activeness for making those experiments come fruitful.

Prof. Mihir Arjunwadkar efforts are really appreciable and I thank him for maintaining such a frank and amateur learning environment at CMS.

I would like to convey my utmost gratitude to Persistent System, Pune for providing me such a great opportunity to work at a global & leading institute providing me with a wonderful working environment.

I think this is a right place to thank my friend Aakash Patil to make me aware that such a great learning Centre exists in Pune & motivating me to join CMS.

Last but not the least I would like to thank my friends, classmates, Parents and colleagues for encouraging me trough this journey at CMS and Persistent Systems.

Contents

Chapter 1

Introduction

Learning tasks are employed into lot of areas this days, modeling new aspects for learning also comprises of Machine learning algorithms which are increasingly employed into various high risk applications, such as medical diagnosis or self driving. However a wide use of this algorithms do not create trust in user. In the current Artificial Intelligence Researchers are not confident enough with their generated models, as they are not good enough to explain with human intuitions, specifically with the Neural Net models. These Neural net models act as **Black box models** (i.e. models that are not capable enough to explain the cause of predicts made by the model).[2]

These Black box models generate a need of explanation in human frame of reference, and the ability to which a model is explainable with the human consistency is called its interpretability. In such context if a Machine learning model is easier for someone to understand the cause of prediction made by model, then it is highly interpretable.[1]

Machine Learning models are nothing but best methods that finds patterns in the training data and connects the input data features to the response variables called target, this generated model is called a trained Machine learning(ML) model. ML algorithms build a mathematical model based on sample data, known as ”training data”, used to make predictions or decisions without being explicitly programmed to do the task, in more mathematical term this learning models generate a expectation function $f(x)$ with input feature values, that can be used for further predictions.

1.1 Are all Models Intpretable?

Validating the model only on the basis of performance parameters viz. Accuracy, precision, Recall could not be a trustworthy approach. If a prediction is made by a model, can we easily accept it ? General human tendency asks ”but why?”. Application of any model before certain steps for validation may create fatal conditions like accident in the case of self driving vehicles. So to build trust on the model is an urgent necessity for researchers, machine learning developers at the same time. Till today researcher face the same problem as no mathematical definition is perfectly defined for interpretability, this is due to the diversities in human perspectives. Each human has own ways of imaginations for a single problem, and the same in explanation approaches. The basic models machine learning such as Randomforest, linear regression are derived with basic analogy of learning to human whereas the Neural Network models are complex algorithms derived from Neural aspects of human brain. This makes Neural network models,very difficult to understand and explain, thus they act as black box models. Neural network models provide high accuracy even in high variance data but are not trustworthy till it could be explained to someone. At the same time researchers are exploring new techniques to maintain a trade-off between model accuracy and model interpretability. In this case study, our focus is primarily on Pre trained Neural network model interpretations and explore best methods that visualizes the cause of classification upto certain extent.

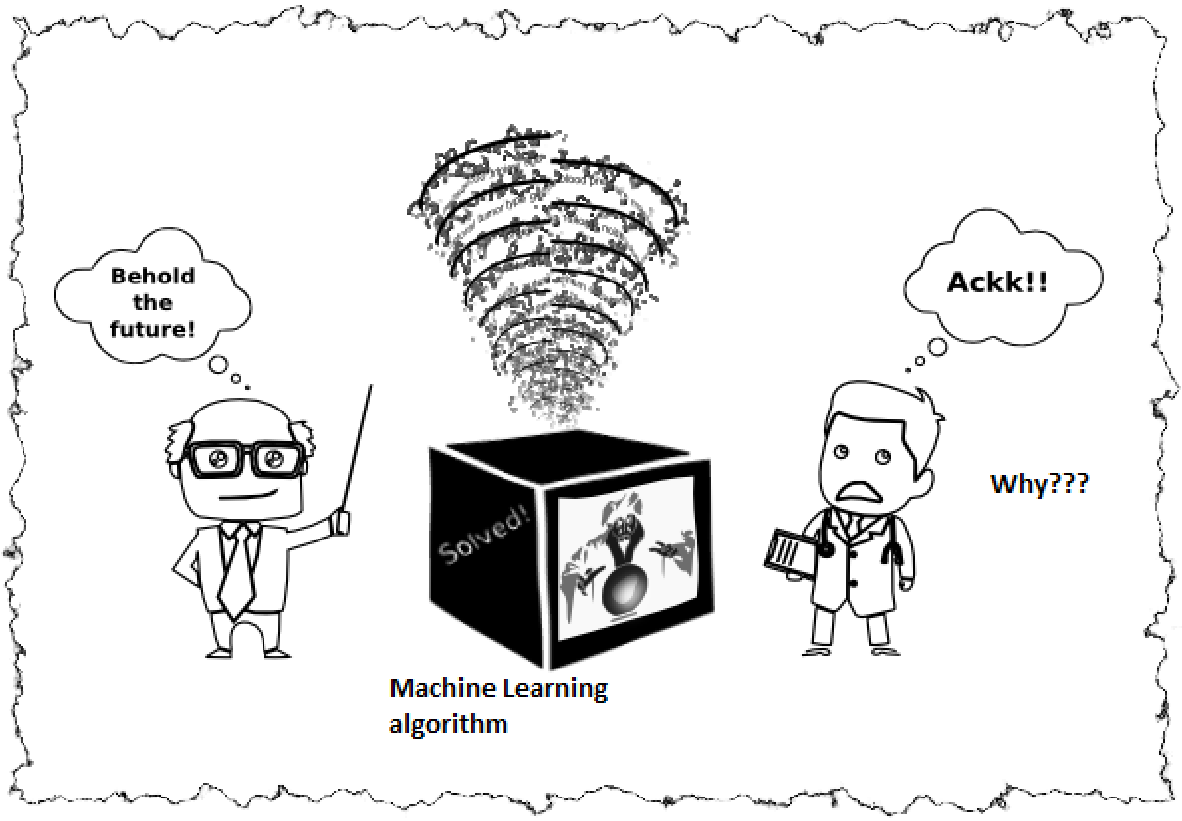

Figure 1.1: Black-box models creates suspicion

1.1.1 Nomenclature of Interpretable Methods

Methods used for explanation of ML models can be classified as: **Intrinsic or post hoc?** This criteria recognizes whether explanation can be accomplished by confining the multifaceted nature of the AI model (intrinsic) or by applying techniques that could investigate the model after training. (post hoc). **Intrinsic interpretability** can be defines as to ML models that are considered easily due to their simple linear structure, such as decision trees or regression models. Whereas **Post-hoc interpretability** refers to methods that can be applied after training the model. For example, calculating impotance values of attributes is a post-hoc explanation technique.

Various interpretation methods can be be categorized on basis of their results generation.[11].

- **Attribute Statistics:**Most of the basic method include statistical summary for each attribute. Some methods produces a single value per attribute, such as attribute importance, or a more complex result, such as the subset attribute strengths, which consist of a permutations for each attribute subset.

- **Statistical visualization:** Lots of data visualization techniques serve good for this methods. Visualization help to increase the explanation tendency.

- **Model intrinsic:**This section consists of Intrinsic ML models . Examples are the coefficients of linear models or a structured tree models. Other method that yields internals is the visualization of filters learned in convolutional networks(CNN).

- **Intrinsic explainable model:** One way of explaining black box model is to replicate it with a linear model.

- **Local or global?** This depend on whether the method is able to explain individual prediction or entire model?

1.1.2 Horizon for Interpretations

A model is trained to genarate the expectations. Each succession can be assessed as far as explanations.

Method Transparency

Method transparency is about how method takes in a model from the information and what sort of connections it can learn. In contexts of CNN algorithm on images and can explain that this method learns edge identifiers , filters and transformations. This is a comprehension of how the method works, yet not for the particular model that is found out at last, and not for how individual predictions are made. Method transparency just requires information of the model and not of the data or learned model.That is why Deep learning approaches are considered less transparent.

Global Model Explanations

To explain the global model predictions, you need the trained model, learning of the information from data. This element of explanation is connected to perceiving how the model chooses, in light of a widely inclusive viewpoint on its features and all of the insightful sections, for instance, parts, for example, loads, various parameters, and structures. Which highlights esteems are critically important and what kind of connection between them occur? Overall model interpretation understands the movement of your target result rely on the features or pixel values esteems in classification of images. Thus global model explanation is difficult to achieve for all intents and purposes.

Local explanations on a sample of data

Predictions for various cases can be explained with global model methods or with interpretation of individual cases.The global methods can be used on group of instances, regarding them as though the gathering were the finished dataset subsets, and utilizing the global techniques with this subset. The single instance explanation methods has to be applied on each instance and then aggregated for the entire sample.

1.1.3 Properties of interpretations

An interpretation methods show mapping between the features values [8] of a case to its model predictions in a humanly justifiable manner. Different kinds of explanation comprise of a lot of information cases. For instance, we could foresee logistic regression applying and explaining predictions utilizing the local technique or Gradient localization technique.
Some basic properties of Interpretation Methods

- **Representability** is the capability of method to represent the expectation function predicted by the model.
- **Dependence** shows the dependency of method on ML model and its input parameters.
- **Method Complexity** describes the computational complexity of the method that generates the explanation. This property should be well considered while application.

Chapter 2

Algorithms and Implementations

2.1 Model-Agnostic Methods

The methods that create explanations separately from training the ML model has some advantages over other methods. Model-skeptic understanding strategies have extraordinary preferred position over model-explicit ones as far as their adaptability. For this situation ML developer get a free hand to pick and prepare with any ML model when this strategies can be applied for explanations.[10].

Anything that expands on the scope of ML models, for example, graph or UI based, likewise becomes independent of the hidden AI model.

For the most part, one, yet numerous kinds of AI models are assessed to illuminate an undertaking, and when looking at models as far as interpretability, it is simpler to work with model-independent explanations, in light of the fact that a similar strategy can be utilized for a model.

A choice to model-skeptic translation strategies is to utilize just interpretable models, which regularly has the huge detriment that prescient exhibition is lost contrasted with other AI models and you constrain yourself to one kind of model. Another option is to utilize model-explicit elucidation strategies. The impediment of this is it additionally ties you to one model sort and it will be hard to change to something different.

Some of the desirable aspects of a model-agnostic interpretation methods are:

- **Model flexibility:**The method sholud be applicable to any pre-trained ML model, such as Support vector machine or Convolutional networks.
- **Explanation flexibility:** Its better to explain with local models such as linear models, random forest rather then going into complex tasks.
- **Visualization flexibility:**The explanation method must be able to use a different visualization techniques as the model being explained. For a content classifier that utilizations dynamic word implanting vectors, it may be desirable over utilize the nearness of individual words for explanation.

Further elaborating some Agnostic methods that seemed to be capable of provide best Interpretations in our case: In this scenario we have to interpret the pre-trained neural net models on skin diseases.

2.1.1 Gradient weighted Class Activation Mapping (Grad-CAM)

This technique is based on Gradient localization technique.It produces visual clarifications for choices from a huge class of CNN- based models, making them progressively straightforward. Gradient weighted Class Activation Mapping (Grad-CAM),uses last convolutional layer to deliver a coarse restriction guide featuring the significant pixels in the image for foreseeing the idea.[6] tasks with multi-modal inputs reinforcement learning, without architectural changes or re-training. Here we combined Grad-CAM with existing visualizations to create a high-resolution class - discriminative visualization and apply it to image classification models. With regards to picture characterization models, our perceptions

(a) produces experiences into

failure methods of these models,

(b) are increasingly trust to the basic model, and (c)

help achieve model theory by recognizing dataset slope and model parameter judgments.

This procedure depends on Zhou et al's. proposed a system called

Class Activation Mapping (CAM) for distinguishing discriminative locales utilized by a confined class of picture characterization CNN.

Graduate CAM technique makes 'great' visual clarification from

the model defending an anticipated class with target as

(a) class discriminative (for example confine the objective classification in the picture)

(b) high-goals (for example catch fine-grained detail).

Convolutional includes normally

hold spatial data which is lost in completely associated layers, so we can expect the last convolutional layers to

have the best trade off between abnormal state semantics and

definite spatial data.

Graduate CAM utilizes the slope data streaming into the last convolutional layer of the CNN

to comprehend the significance of every neuron for a choice

of intrigue. The class discriminative localization map Grad-CAM $L^c_{GradCAM} \in \mathrm{R}\ ^{u\times v}$ of width u and height v for any class c, y^c w.r.t. the feature maps A^k of a convolutional layer, i.e. $\partial y^c/\partial A^k$. These gradients flowing back are global-average-pooled to obtain the neuron importance weights α^c_k.

$$\alpha^c_k = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \qquad \underbrace{\frac{\partial y^c}{\partial A^k_{ij}}}_{\text{gradients via backprop}} \tag{2.1}$$

here weight α^c_k represents a partial linearization of the deep network downstream from A, and captures the importance of feature map k for target class c.

On performing a weighted combination of forward activation maps, and follow it by a ReLU to obtain,

$$L^c_{GradCAM} = ReLU \underbrace{\left(\sum_k \alpha^c_k A^k\right)}_{\text{linear combination}} \tag{2.2}$$

This results in a coarse heat-map of the same size as the convolutional feature maps (14×14) in the case of last convolutional layers of VGG and AlexNet. ReLU is applied to the linear combination of maps to get the features that have a positive influence on the class of interest, i.e. pixels whose intensity should be increased in order to increase y^c. Negative pixels belong to other categories in the image.

Grad-CAM as a generalization to CAM

CAM creates a restriction map for a picture arrangement CNN with a particular sort of engineering where global normal pooled convolutional highlight maps are nourished straightforwardly into softmax. Specifically, let the penultimate layer produce K feature maps, $A^k \in R^{u \times v}$.

The feature value are then spatially pooled using Global Average Pooling (GAP) and linearly transformed to produce a score S^c for each class c.[7] To produce the localization map for modified image classification architectures, such as above, the order of summations can be interchanged to obtain L^c_{CAM}.

$$S^c = \frac{1}{Z} \sum_i \sum_j \underbrace{\sum_k \omega_c^k A_{ij}^k}_{L^c_{CAM}} \tag{2.3}$$

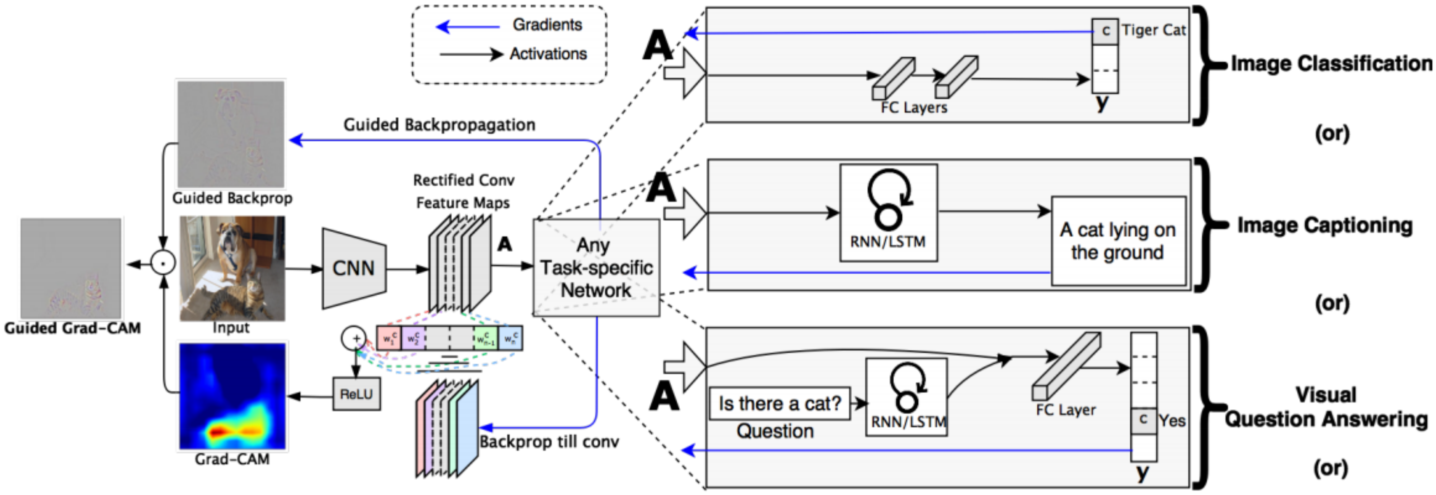

Figure 2.1: Grad-CAM overview:

The above figure depicts a class of interest (e.g., 'AIKEC type disease' or some other sort of differentiable class) as information, we forward spread the picture through the CNN part of the model and after that through calculations to get a crude scores for the class. The gradients for different classes are set to zero with the exception of the desired class (AIKEC type disease), which is set to 1. This sign is then backpropagated to the redressed convolutional highlight maps of intrigue[4], which we consolidate to figure the coarse Graduate CAM restriction (blue heatmap) which speaks to where the model needs to hope to settle on the specific choice. At last, we pointwise duplicate the heatmap with guided backpropagation to get Guided Grad-CAM representations which are both high-goals and idea explicit.

2.1.2 Local interpretable model-agnostic explanations (LIME)

The main objective of LIME[5] is to identify a basic model over the other visulization methods that is more explanable and uses a representation that is understandable to humans, regardless of the actual features used by the model. For example, a possible representation for text classification is a count vector representing the frequencies for each word in list of sentences, althoguh the classifier may use more complex features such as word term and document frequency matrix. Similarly in case of image classification, a basic explainable method is the representation of the ” or ” of a spatial importance of similar pixels, whereas a classifier may represent an image as tensor with three channels per pixel. We denote x $\in R^d$be the original representation of an instance being explained, and we use x $\in \{0,1\}^d$to denote a binary vector for its explainable representation.

Accuracy-Interpretability Trade-off: Formally, we define an explanation as a model g $\in G, whereGisaclassofpotentiallyinterpretablemodels, suchaslinearmodels, decisiontrees, i.e.amodel$g $\in G$ can be readily presented to the user with visual or textual artifacts. The domain of g is $\{0,1\}^d$, i.e. g acts over absence/presence of the interpretable components. In order to ensure both interpretability and local fidelity, we must minimize $L(f;g;\pi_x)$ while having $\Omega(g)$ be low enough to be interpretable by humans.

The explanation is produced by:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \quad L(f;g;\pi_x) + \Omega(g) \tag{2.4}$$

Local model Explanations If we define G be the class of linear models such that $g(z') = w_g z'$. We use the locally weighted square loss as L, as defined in Eq. (2), where we let $\pi_x(z) = exp(D(x;z) = \sigma)$

2.1.3 SHapley Additive exPlanations (SHAP)

The best explanation of a straightforward model is simply the model however complex models are not easy to understand, for example, ensemble methods or deep networks. For these mind boggling model we need a basic clarification model which is dened by any interpretable guess of the first model.

Let f be the original prediction model to be explained and g the explanation model. Here, we focus on local methods designed to explain a expectation f(x) based on a single input x, as proposed in LIME. Explanation models often use simplied inputs x' that map to the original inputs through a mapping function $x = h_x(x')$. Local methods try to ensure $g(z') \approx f(h_x(z'))$ whenever $z' \approx x'$. (Note that $h_x(x') = x$ even though x' may contain less information than x because h_x is specic to the current input x.)

Feature attribution methods[3] have an explanation model that is a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i^{'} z' (2.5)$$

where $z' \in \{0,1\}^M$, M is the number of simplied input features, and $\phi_i \in R$
Many current methods matches with several of which are discussed below:

- **LIME** This method explains the individual model expectations based on locally approximating the model around a given prediction. The local linear explanation model that LIME uses follows equation 2.5 exactly and is thus a feature attribution method.

- **DeepLIFT** This method attributes to each input x_i a value $C \triangle x_i \triangle y$ that represents the effect of that input being set to a reference value as opposed to its original value. This means that for DeepLIFT, the mapping $x = h_x(x')$ changes over paired qualities into the first data sources, where 1 shows that an info takes its unique esteem, and 0 demonstrates that it takes the reference esteem. DeepLIFT uses a "summation-to-delta" property that states:

$$\sum_{i=1}^{n} C_{\triangle x_i \triangle o} = \triangle o \tag{2.6}$$

where O = f(x) is the model output, $\triangle o = f(x)f(r)$, $\triangle x_i = x_i r_i$, and r is the reference input. If we let $\phi_i = C \triangle x_i \triangle o$ and $\phi_o = f(r)$, then DeepLIFTs explanat4 model matches Equation 2.5 and is thus feature attribution method.

- **Layer-Wise Relevance Propagation** The layer-wise relevance propagation method interprets the predictions of deep networks. From Shri kumar et al.,this menthod is equivalent to DeepLIFT with the reference activations of all neurons xed to zero. Layer-wise relevance propagations explanation model, matches Equation 2.5.

- **Classic Shapley Value Estimation** Three previous methods use classic equations from cooperative game theory to compute explanations of model predictions: Shapley regression values, Shapley sampling values, and Quantitative Input Inuence. Shapley regression values are feature importance values for linear models in the presence of multi collinearity.

The Shapley Value Linear models are generally based on the coefficient values that suffice to generate a good classier line onto the data. Let see how linear model resembles for one instance case:

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

where x is the instance for which we want to compute the contributions. Each x_j is a feature value, with j = 1,,p. The β_j is the weight corresponding to feature j. The contribution ϕ_j of the j-th feature on the prediction $\hat{f}(x)$ is:

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j) = \beta_j x_j - \beta_j E(X_j)$$

where $E(\beta_j X_j)$ is the mean effect estimate for feature j.

$$\begin{aligned}\sum_{j=1}^{p} \phi_j(\hat{f}) =& \sum_{j=1}^{p} (\beta_j x_j - E(\beta_j X_j)) \\ =& (\beta_0 + \sum_{j=1}^{p} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{p} E(\beta_j X_j)) \\ =& \hat{f}(x) - E(\hat{f}(X))\end{aligned}$$

This is the predicted value for the data point x minus the average predicted value. Here shapley value can be defined using a value function val of players in S.

$$\phi_j(val) = \sum_{S \subseteq \{x_1,\ldots,x_p\} \setminus \{x_j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \left(val\left(S \cup \{x_j\}\right) - val(S)\right)$$

(2.7)

where S is a subset of the input attributes, x is the vector of input attributes values of the instance to be explained and p the number of total attributes. $val_x(S)$ is the prediction for attributes values in set S that are marginalized over attributes that are not included in set S:

$$val_x(S) = \int \hat{f}(x_1, \ldots, x_p) d\mathbb{P}_{x \notin S} - E_X(\hat{f}(X)) \tag{2.8}$$

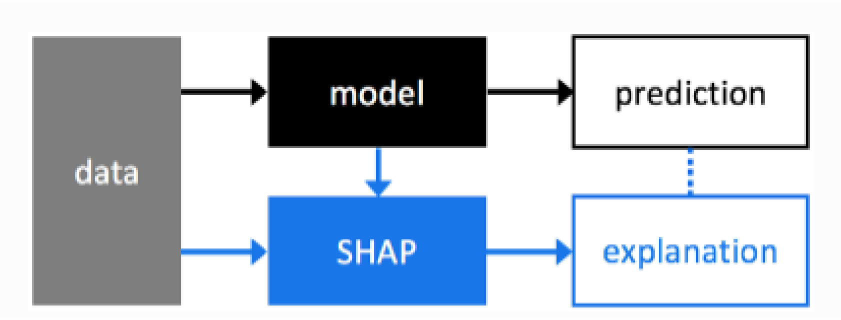

Figure 2.2: Shap workflow

2.1.4 Pseudo Codes

LIME

Algorithm 1 Algorithm 1 Sparse Linear Explanations using LIME

```
Require: Classier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel $\pi_x$, Length of explanation K.
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, ..., N\}$ do
      $z'_i \leftarrow samplearound(x')$
      $Z \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
  end for
  $\omega \leftarrow K - Lasso($ Z,)$withz$'$_i$ as features, f(z) as target

return $\omega$
```

Algorithm 2 Algorithm 2 Submodular pick (SP) algorithm

```
Require: Instances X, Budget B
  for  all $x_i \in X$ do
      $W_i \leftarrow explain(x_i, x'i)$                    ▷ Using Algorithm 1
  end for
  for $j \in \{1, ..., d'\}$ do
      $I_j \leftarrow \sqrt{\sum i = 1^n \|W_i j\|}$        ▷ Compute feature importances
  end for
  $V \leftarrow \{\}$
  while $\|V\| < B$ do                                       ▷ Greedy optimization
      $V \leftarrow V \cup argmax_i c(V \cup \{i\}, W, I)$
  end while
  return V
```

2.2 Implementation

2.2.1 Tensorflow-Keras Implementation

The algorithms as mentioned in Section 3.5 require computations of large tensors. And Google's one of the finest work for deep learning tool is Tensorflow,[9] core open source library to help us develop and train ML models. Also allows us to run projects on Google Colab or cloud platforms.

Further, algorithmic implementations were developed Pre-build models neural net architectures available from Tensorflow viz. Resnet, VGG16, DenseNet, Squeezenet etc. which performed well on the Input image dataset.

Input Data

Initially, training data-set consisted of 7 classes of skin disease with total 87 images. This classes contained skin disease images with random pixel size. And Keras module contains commands over TensorFlow backend for Neural net procedures. Data Pre-processing involves data generation using Image data generator functions.

Parameters required for processing the data sets are

- image size (224, 224, 3), were defined as(Height, width, Channels)
- Data directory location.

2.2.2 PyTorch Implementation

For the reasons of speed, efficiency, flexibility and to run code on Google colab some algorithms are implemented using PyTorch. Due to Cuda base and its distributed training it is quite faster to implement Neural Net Models in PyTorch. Integration with Numpy arrays makes PyTorch tensors convertable well. PyTorch is largely supported on different cloud platforms, providing frictionless development and easy scaling through prebuilt images, large scale training on GPUs, ability to run models in a production scale environment, and more.

Output

All three algorithms gives different visualizations on same instance image. Grad-CAM method generates a colored contour plot over the image, showing the gradients flow over the image. SHap method generates images with red colored points onto image which represent important pixels. LIME method helps to mask and unmask the pixel locations with black color which shows an effective visualization.

Chapter 3

Results and Discussion

3.1 Results from implemented methods

3.1.1 Formulation

To get the explanations of trained models we applied three methods, which produces different visualization patterns for each instance image. This models are trained on skin disease images with size 224 × 224 and then the explainable method are implemented using prebuild python packages.

3.1.2 Discussion

The results generated from Grad-CAM methods are visualized in the grid structures with input image on left side and Grad-CAM results on right applied on different pretrained ML models viz. Alexnet, VGG16, Resnet, Densenet, Squeezenet etc.Those results are compared with updated Grad-CAM++ method. The following Figure 3.4 show results with the different images from different classes.

With implementation of these global Model Agnostic methods we have generated an explainable visualization of trained models on the skin image dataset. Also on evaluation with human intuitions they are good enough to explain the cause behind the classifiers working. These visualization will also help machine learning developers for selection of pretrained models. The representation will help to gain trust over the machine learning models.

Grad-CAM Visualization:

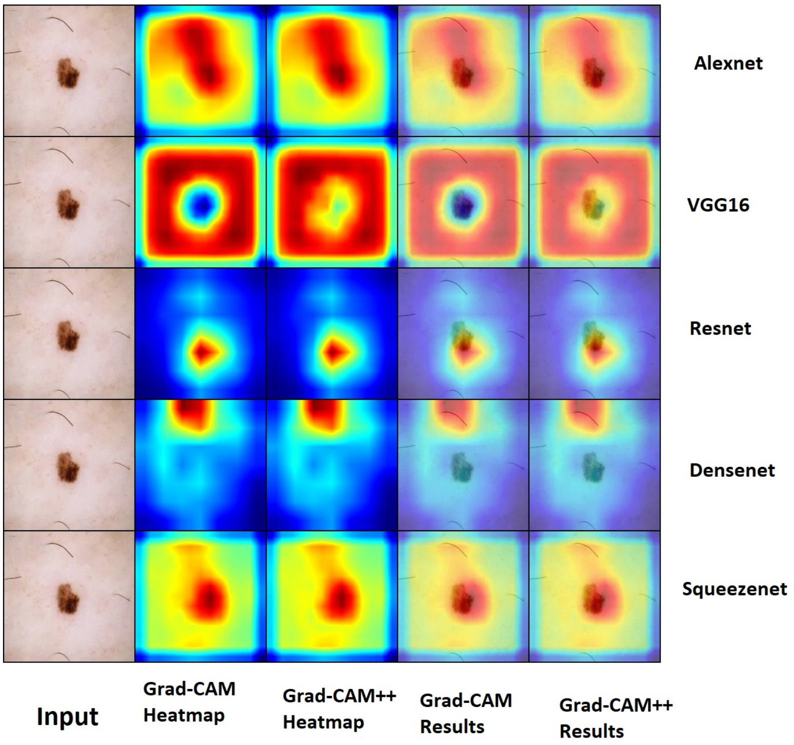

Figure 3.1: GRAD-CAM results for ISIC0027343 image

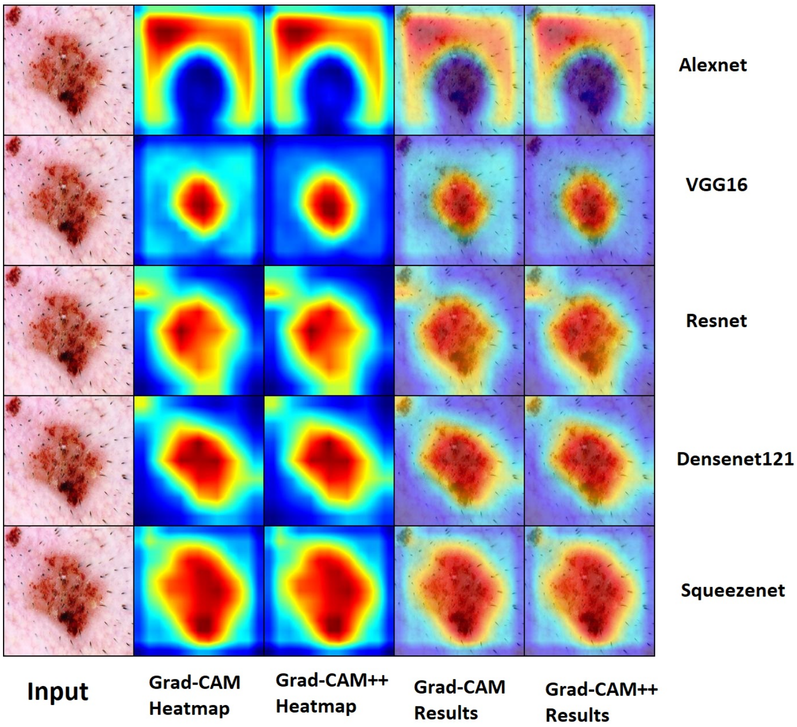

Figure 3.2: GRAD-CAM results for ISIC0025948 image

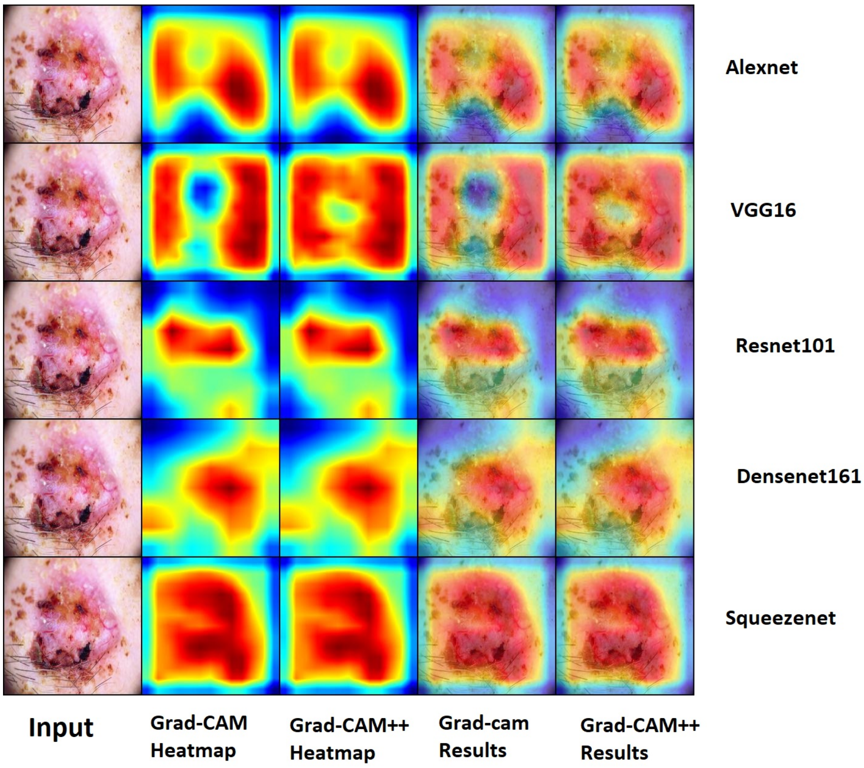

Figure 3.3: GRAD-CAM results for ISIC0024457 image

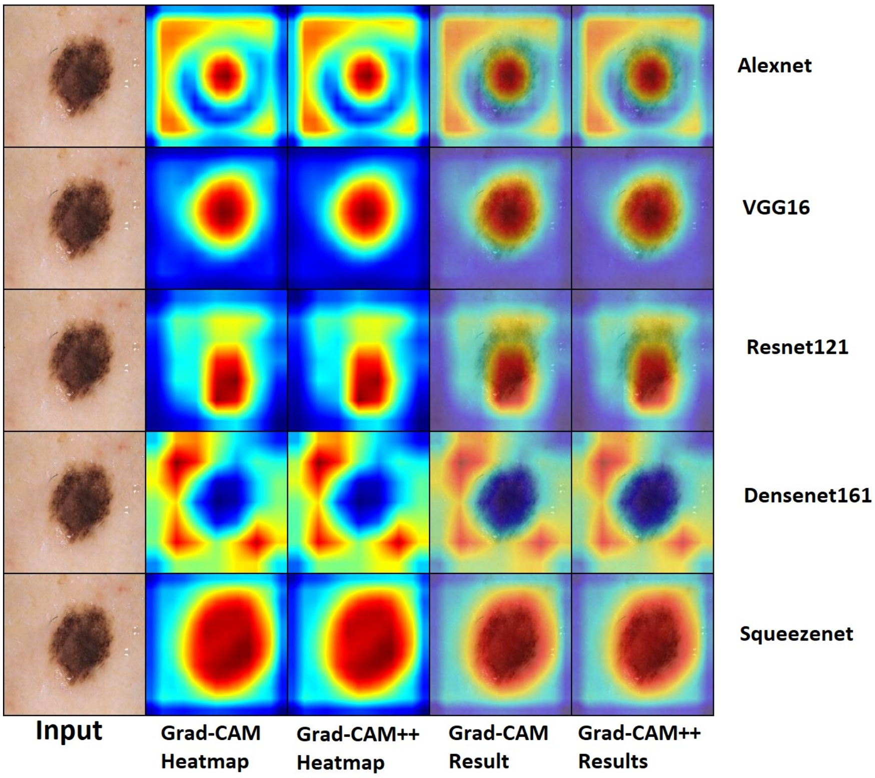

Figure 3.4: GRAD-CAM results for ISIC0024319 image

LIME visualization:

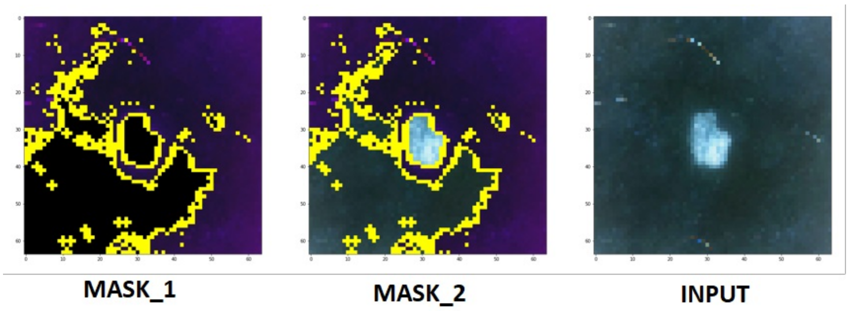

Figure 3.5: GRAD-CAM results for ISIC0024319 image

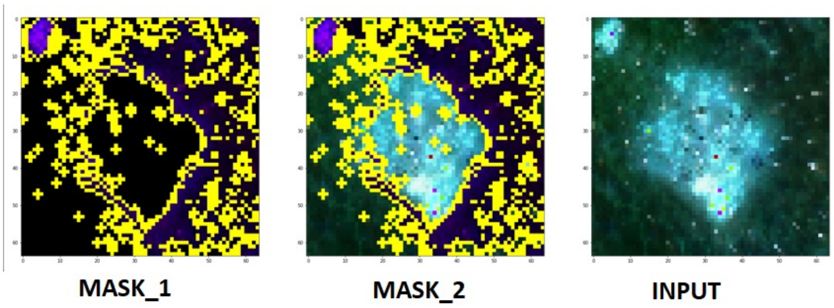

Figure 3.6: GRAD-CAM results for ISIC0025948 image

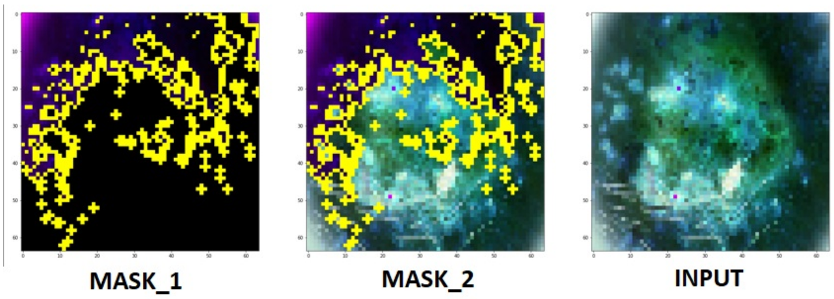

Figure 3.7: GRAD-CAM results for ISIC0024457 image

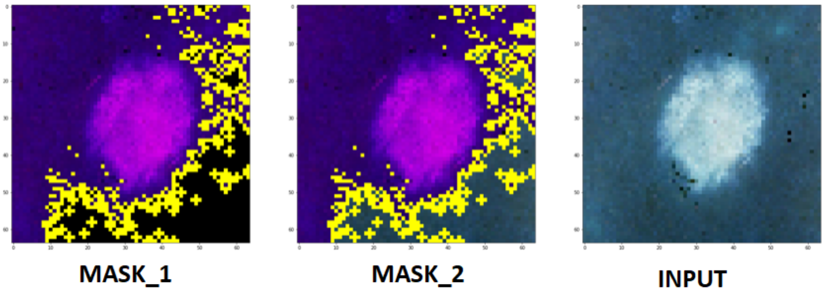

Figure 3.8: GRAD-CAM results for ISIC0024319 image

SHAP Visualization

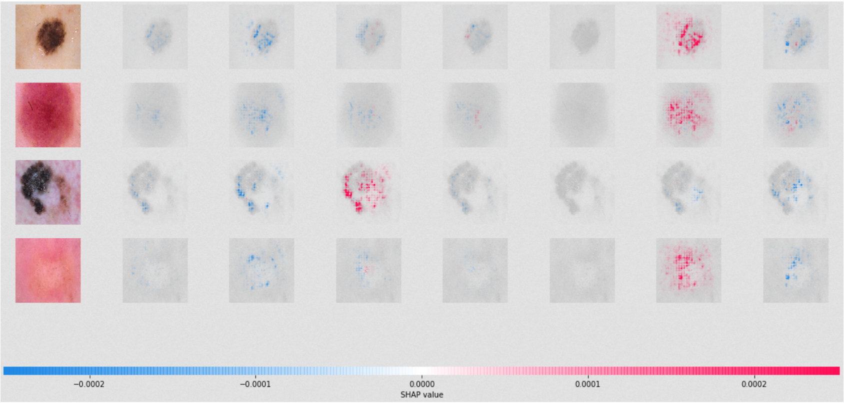

Figure 3.9: SHAP results for images from different classes

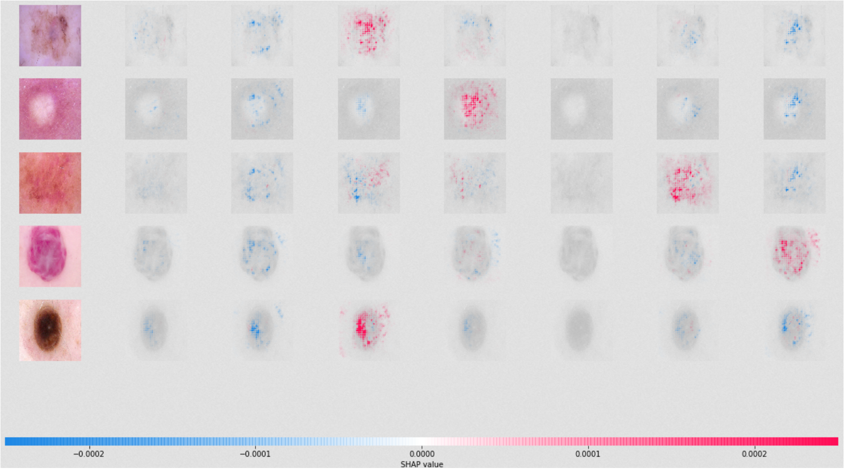

Figure 3.10: SHAP results for images from different classes

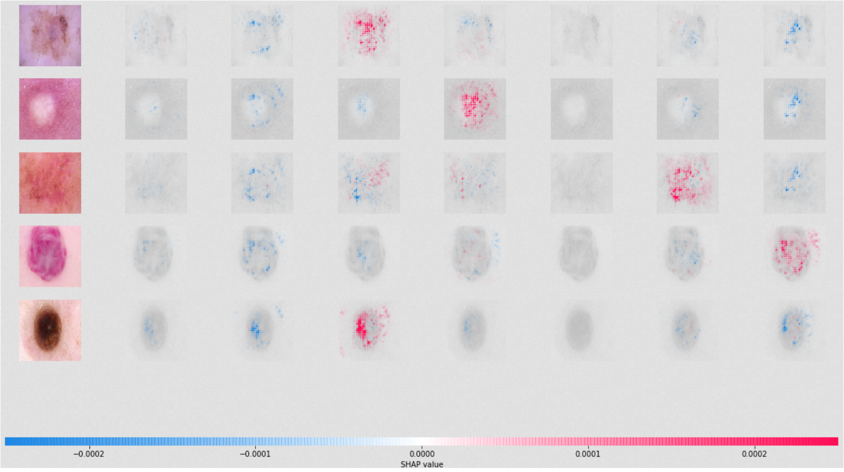

Figure 3.11: SHAP results for images from different classes

Chapter 4

Summary and Conclusion

We have implemented and evaluated the current frameworks to generate interpretations of Pre-trained Neural Network Classifiers. The generated results from three different methods have generated explainable visualizations. And can be further improved with the different developing visualization techniques.

Results generated from Grad-CAM technique shows the importance of selection model from subset of trained models. While comparison between models Grad-Cam method shows the exact regions localized by classifiers for predictions which easily gain trust over good model. In Lime technique the mask and unmask regions of the image instances show a positive response interpretation of the trained model. Lime method produces a good human perception for explaining black box models.

The applied shape method retains the exact pixel locations within image instances and class wise visualization technique shows cause behind the prediction for that instance. After all this experimentation we are agreed with the clause that there is always a need to interpret the trained model even if it provides highest performance values. One should make sure to understand the cause and produce interpretations of the same model.For example, deal with making a trusted medicinal services operator may be surrounded as concentrated on the requirement for explanation because of local contributions at the neighborhood scale, assessed at the dimension of an application. In other way,working on the linear models may also help to encircled as concentrated on the requirement for explanation because of unknown inputs of information, however this time assessed at global scale.

All this three methods have different visualization techniques, though seems to work well enough in this case. Although with this experimentation we faced solid problem in evaluation of interpreted results from these method, some how the evaluation strategies is defined over to human understanding, by visual explanation we were able to build a trust onto the accepted trained models and able to explain the reason behind acceptance of those model. Also all the three methods can be easily implemented in python with available python modules. There is good space and need for enumerating machine learning techniques and we believe there is a dire need to explore capabilities and cause behind each neural model in the view of developing or in making it trustworthy.

Bibliography

[1] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. 2017.

[2] Z. Lipton. The mythos of model interpretability. *Communications of the ACM*, 61, 10 2016.

[3] S. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *NIPS*, 2017.

[4] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *NIPS*, 2017.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin. ”why should i trust you?”: Explaining the predictions of any classifier. In *HLT-NAACL Demos*, 2016.

[6] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[7] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.

[8] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[9] M. Staniak and P. Biecek. Explanations of Model Predictions with live and breakDown Packages. *The R Journal*, 10(2):395–409, 2018.

[10] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[11] K. Zolna, K. Geras, and K. Cho. Classifier-agnostic saliency map extraction. *CoRR*, abs/1805.08249, 2018.