Face Mask Detection with Explainable Artificial Intelligence

1st Athena, Liu
Halicioğlu Data Science Institute
University of California, San Diego
La Jolla, California
atl074@ucsd.edu

2nd Che-Wei, Lin
Halicioğlu Data Science Institute
University of California, San Diego
La Jolla, California
chl820@ucsd.edu

3nd Gavin, Tran
Halicioğlu Data Science Institute
University of California, San Diego
La Jolla, California
gatran@ucsd.edu

***Abstract*—This report addresses the concern of careless mask wearing due to the ongoing pandemic, which has caused small businesses and large businesses alike financial problem. As a result, we tackle this issue by building a face mask detector that will recognize whether a person is wearing a mask. More importantly, not only is our face mask detector able to detect whether a person has a mask on, it can also detect whether the person is wearing a mask correctly, or with both chin and nose covered. Our detector was trained on a dataset called MaskedFace-Net, which contains more than 35000 images, and was able to fit the training data while performing even better on the validation and test set. It is able to achieve 88% a 95% accuracy on training set and both validation set and test set, respectively.**

***Index Terms*—Explainable AI, Image Classification, Grad-CAM, Integrated Gradient**

I. Introduction

In recent years, enhanced computational power allows us to handle data in large amount at unprecedentedly high efficiency, which in turns gives us the opportunity to do task that has never been done before. For example, in the field of computer vision, researchers have been using deep learning, which is a method based on neural networks in order to learn from data, to create algorithm that locates objects and retrieves their attributes from an image. This approach has obtained impressive results in numerous computer tasks such as image classification, semantic segmentation, and instance segmentation. However, using deep learning to conduct computer vision related tasks has also garnered some criticisms. One criticism describes neural network as a black box, which basically means we teach it with some expectations on the output by feeding them enormous data. However, we do not have any idea of how it achieves what we want. Therefore, the question "How does a neural network achieve what it's supposed to achieve" is crucial in understanding how does an algorithm detects objects and its features from an image. This will not only convincingly make people understand how it works, but also allow researchers to troubleshoot.

In this project, after building a task-oriented image classification algorithm, we specifically would like to investigate the problem of how can such algorithm be trusted to give the correct results. We would like to make sure that the process through which the neural network is able to produce the output we expect is one that is explainable and logical. To paraphrase it and put it in simpler terms, how can we make sure that an algorithm can successfully recognize the objects in an image because of its unique features and not because of something else is the key question to answer. Approaches to finding such answers are also known as "Explainable AI" methods, which lies at the core of this report. In summary, the workflow of this project can be described as below:

1) Train a model that will help classify images.
2) Generate performance of the model
3) Implement "Explainable AI" methods to ensure the performance of the model is valid and reasonable.

II. Context/Motivation

2020 has been a year of tribulations and sufferings. The COVID-19 pandemic has taken lives of many and is still widespread across the world. More importantly, many people's lives are changed in a drastic manner. For example, to comply with government laws, many businesses are asked to require customers to wear masks upon entering a building, and they have to refuse services for customers who don't cooperate. Therefore, to both customer and business owner, it has never been more important to wear masks in public, especially for business owners, who need the general public to wear masks in order for their businesses to survive. As a result, having a face mask detector that will detect whether a customer has his mask worn upon entry is crucial to survival of their business. Therefore, we've decided to build a model that will recognize whether a person in an image is wearing a mask or not.

Previous work on face mask detection mainly dealt with detecting whether a person has a mask on or not in an image. For example, Adrian RoseBrock, in his website [1], had successfully created a face mask detector. However, the dataset he worked with only consists of 1376 images, which is fairly small for computer vision tasks. In addition, the dataset only has images with people wearing masks or not. It doesn't contain images where a person is wearing a mask but not covering his nose or chin. In other words, it doesn't have images where a person is wearing a mask improperly. We believe that this is a deficiency that is important to address, and our model and dataset will take this into account.

In this report, we aim to develop an image classification model that will recognize the content of an image and label it correctly. In addition, to make our model more powerful, we will provide the model with the ability to detect whether a person in an image is wearing a mask correctly or not. As mentioned above, this feature will be particularly useful due to the fact that a lot of people are wearing their masks improperly because apparently they find trouble breathing. After finish building the model, we will check whether the model is correctly interpreting the results by implementing Explainable AI methods such as Grad-CAM and Integrated Gradient, which are two powerful visualizing algorithm that can validate the workings of our model. Grad-CAM and Integrated Gradient will be explained in greater detail later.

In summary, our contributions to this report include:

- We build a face mask detector that will be put into good use in light of the pandemic.
- We provide the face mask detector with attention to detail skills that are crucial during the pandemic.
- We make sure our model is working the way we want through Explainable AI methods (Grad-CAM, Integrated Gradient)

III. Dataset

Our model will be trained and evaluated on MaskedFace-Net [2], which contains more than 50000 images, each one of them with a person in it. The entire dataset is split into three parts: train, validation, and test. Furthermore, each image is classified as either "correctly masked", "incorrectly masked", and "not masked". Figure 1, 2, and 3 show all three types of image:

Fig. 1. An image with correctly masked person

Fig. 2. An image with incorrectly masked person

Fig. 3. An image with a person that is not masked

This multi-label feature allows us to build a model that is sensitive to improper mask wearing. Table I shows the statistics of the dataset:

	Train	Validation	Test
Correctly Masked	12362	3090	3863
Incorrectly Masked	12338	3084	3856
Not Masked	12800	3199	4000
Total	37500	9373	11719

TABLE I
STATISTICS OF MASKEDFACE-NET

From the table, it can be seen that the subset of the dataset is distributed equally across all three sets.

IV. Methodology

A. Task Formulation

Our classification problem can be summarized as follows: Given an image consisting of pixels $I = \{x_1, x_2, ..., x_n\}$, where $x_i \in \mathbb{R}$ is the i-th pixel containing 3 color channels, and a category $Y \in \{$correctly masked, incorrectly masked, not masked$\}$, we aim to train a model that maps each I to Y such that the accuracy is maximized.

B. ResNet50

Our first step in building a reliable face mask detector is to build a model that will correctly classify the image. For our baseline model, we will use a deep residual learning framework called ResNet50 [3]. The concept of ResNet50 draws from the fact that both the training error and test error actually increase when a plain deep neural network is added more layers. Figure 4, which is from the paper, compares the error from networks with 20 layers and 56 layers.

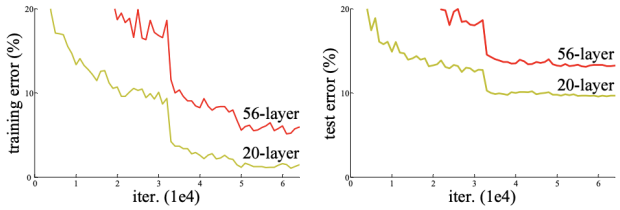

Fig. 4. Training and testing errors on CIFAR-10 from network with 20 and 56 layers respectively. Notice the network with 56 layers has higher training and testing error.

When the model becomes much deeper, meaning most of the features have already been learned, adding an additional layer will only result in the model trying to map the previous set of feature to the exact same set of feature. This mapping is also known as identity mapping, which should be a simple process. However, at this point the model will tend to over complicate the process and thus result in high error.

To combat this issue, the authors proposed a novel approach based on the concept of "residual block". ResNet is made up of residual blocks. A diagram of residual block can be seen in Figure 5, which is from the paper:

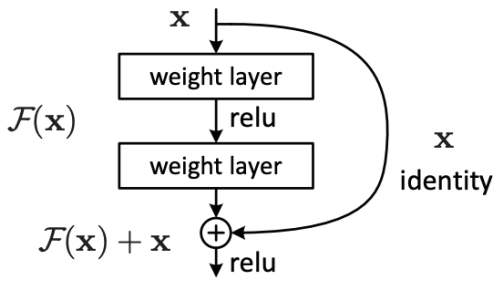

Fig. 5. A diagram of residual block

A residual block is composed of more than one weight layer, and is designed in a way that will combat high error due to more layers being stacked. In a residual block, a method called skip connection (illustrated as an arc in the figure) is used so that, instead of an input x having to go through each layer, it will directly skip some layers to prevent the model over complicating the process. At the same time, we will still apply learning to a new x so the $F(x)$ is produced, which represents small additional feature learned during the mapping. And the result is obtained by adding $F(x)$ and (x). This way not only will the over complication problem be avoided, but the model will learn new feature. Figure 6, which is from the paper, illustrates the comparison of plain neural network and ResNet:

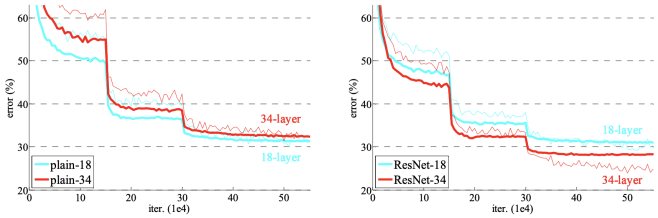

Fig. 6. Training and testing errors from plain network and ResNet. Notice that with ResNet, adding more layers result in lower error.

We decide to choose ResNet50 (50 layers) over ResNet18 (18 layers) or any other with fewer layers because we believe with the feature of residual block and 50 layers, it will discern the difference between correctly wearing mask and incorrectly wearing mask without producing higher error than the one with fewer layers. In addition, since our classification task is a multi-label classification task, we will fine tune our model by adding an additional linear layer that maps the output of original ResNet50 to a 3 x 1 vector instead of binary results.

C. Grad-CAM Algorithm

Once we finish building a face mask detector with ResNet50 as its core component and recording its performance, we will validate its performance with Grad-CAM [4]. Grad-CAM, short for "Gradient-weighted Class Activation Mapping", is a technique that overlays a class activation map over the original image that highlights the important object in the original image. Figure 7 illustrates the process of producing a class activation map:

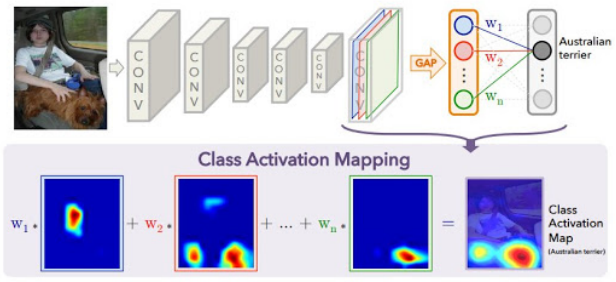

Fig. 7. Process of producing a class activation map in a neural network

In the figure, as an image passes through the convolutional layers, its size gets reduced, but the features learned gets increased. The result is a series of feature maps, which is then passed into a GAP (Global Average Pooling) layer. The GAP layer will transform each feature map into a single value that best represents each feature. Equation (1) shows the mathematical formula of obtaining the output of GAP layer:

$$F^k = \frac{1}{Z}\sum_i\sum_j A_{ij}^k \tag{1}$$

where $\frac{1}{Z}\sum_i\sum_j$ is the process of global average pooling, A_{ij}^k is the feature map k, and F^k is the global average pooled output.

Then, the class activation map is created by calculating the weighted sum of the weighted associated with each feature and the feature value. Equation (2) shows the formula of obtaining the weighted sum

$$Y^c = \sum_k w_k^c \cdot F^k \tag{2}$$

where w_k^c is the weight connecting the k^{th} feature map with the c^{th} class, and Y^c is the weighted sum.

Finally, to make a connection between CAM and Grad-CAM, we will recompute the weight w_k^c by pooling the gradient of weighted sum with respect to the feature value and apply an activation function $ReLU$, as shown in Equation (3) and (4) respectively:

$$w_k^c = \sum_i\sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \tag{3}$$

$$L_{GradCAM}^c = ReLU(\sum_k w_k^c A^k) \tag{4}$$

The $ReLU$ activation function will replace every negative value with 0 to emphasize the positive weight, which indicates that this particular feature is strongly associated with the important object in the image.

Last but not least, it is worth noting that the map is able to locate the object in the original object because during the pooling operation, which aggregates multiple values into a single value, it is able to locate positive values which associate with the object the most. Therefore, the location of such values will be the location of the highlighted object.

We choose Grad-CAM because we believe with proper training of the model on MaskedFace-Net, our model will be able to learn important features such as the mask and a person's nose, mouth, and chin, all of which will be highlighted when Grad-CAM is applied.

D. Integrated Gradient

Next, we will also validate our model with Integrated Gradient [5]. Similar to Grad-CAM, Integrated Gradient is another visualizing method that detects features that are important in classifying an image. Integrated Gradient relies on a method

called "attribution method", which, given an input x and a network function $f(x)$, will assign a score to each feature. Similar to Grad-CAM, a positive value indicates that the feature is strong associated with the output , and a negative value indicates that the feature is weakly associated with the output. 0 indicates that the feature has no effect on the output.

Figure 8 shows the formula to compute the integrated gradient:

$$\phi_i^{IG}(f, x, x') = \overbrace{(x_i - x'_i)}^{\text{Difference from baseline}} \times \underbrace{\int_{\alpha=0}^{1}}_{\text{From baseline to input...}} \overbrace{\frac{\delta f(x' + \alpha(x - x'))}{\delta x_i}}^{\text{...accumulate local gradients}} d\alpha$$

Fig. 8. Formula of Integrated Gradient

where x_i is the input, x_i^i is the baseline input, which is usually a blank or black image meant to represent absence of feature, and f is the network function. To understand the formula, we will divide it into 3 steps:

1) Compute the gradient of function output with respect to feature i
2) Integrate over the gradients to avoid saturation problem (meaning some features might have small gradients even if they are important)
3) Multiply the difference from baseline to get the feature importance score

Integrated Gradient is also an important method to use because we would like to check if the feature importance score around the mask is high, which will validate our model's inner workings.

V. Result

A. Experiment

The model we specifically trained the Masked-Face Net dataset on was a pretrained ResNet50 model with the base parameters of training on CPU, a batch size of 32, a learning rate of 0.001, and two epochs. For training the model on a CPU, the base parameter is set to using the CPU in the case that the user doesn't have a GPU in their system, but we ended up training our model using a GPU due to how much quicker the training process finishes. For the batch size of 32, using a size of 32 is a fairly common value, the importance of the batch size relates to the number of samples that are read in at a time, affecting memory, speed, and accuracy of a model. We mostly decided on using this batch size due to fluctuations on both accuracy and runtimes when either increasing or decreasing the batch size. For our learning rate, we decided to stick with a value of 0.001, the importance of a learning rate is in regards to tuning the weights to optimize the loss function, which simplified, a lower learning rate makes training more reliable but longer runtime whereas vice versa, a large learning rate makes training less reliable but shorter runtimes. We chose this value for the learning rate due to the same reason as our batch size evaluation, in which this value performed better and faster than the other common learning rate values of 0.1 and 0.01. We used two epochs to train our model, which refers to how many times we trained our model on the entire dataset, so in this case we trained the model on the dataset twice, since our model takes a fairly long time to run and the accuracy we gain from running more epochs is negligible.

B. Performance

After conducting the experiment, the performance of ResNet50 on training, validation, and testing set from MaskedFace-Net can be seen in Table II:

	Train	Validation	Test
Correctly Masked	89%	88%	94%
Incorrectly Masked	78%	93%	92%
Not Masked	94%	99%	98%
Total	88%	95%	95%

TABLE II
Performance of Fine Tuned ResNet50 on MaskedFace-Net

Across the sets of data, we are surprised to find that ResNet50 not only fits the training data but also performs well on both the validation data and training data, both of whose accuracy are higher than the training data. This is different from we expected in the beginning, as we thought the model will have an overfitting problem. Instead, it turns out that the model is able to prevent that. Equally surprising is that the model did well across different labels. Although the images with incorrectly masked person have a slightly lower accuracy than others, it is reasonable due to the fact that to be able to recognize a person is not wearing a mask requires lots of features to be detected, not just mask. Also, we are satisfied with the fact that the images with people who are not masked achieved the highest accuracy, which is totally reasonable as well because the absence of mask will be very easy for the model to learn and detect.

Moving on to the visualization of Grad-CAM on our model, we are also content with the result. Figure 9, 10, and 11, show the images with correctly masked, incorrectly masked, and not masked, and for comparison, Figure 12, 13, and 14 show the Grad-CAM implementation of the images:

Fig. 9. An image with correctly masked person

Fig. 10. An image with incorrectly masked person

Fig. 11. An image with a person that is not masked

For the image with a correctly masked person, it is evident that the model has learned the feature "mask" because in Figure 12, the heat map is generated with the emphasis on the mask, which is a desirable result. For the image with a

Fig. 12. Grad-CAM results of Figure 9

Fig. 13. Grad-CAM results of Figure 10

Fig. 14. Grad-CAM results of Figure 11

incorrectly masked person, it is also obvious that the model has kept its focus on the mask. However, the heat map also has successfully focused near the area between the edge of the mask and the person's nose, indicating that the model has learned the feature "philtrum" and "nose". Last but not least, for the image with a person that is not masked, with the heat map covering the mouth and chin of the person, it is possible that the model has learned the feature "mouth" and "chin" and is able to figure out that if both mouth and chin are exposed, then it's impossible for a person to have a mask on, be it correctly or incorrectly. In summary, judging from the results of Grad-CAM from each of the label, the high accuracy is attributed to the model's correct learning of the feature present in the images.

As for Integrated Gradient, Figure 15, 16, and 17 show the results of the implementation:

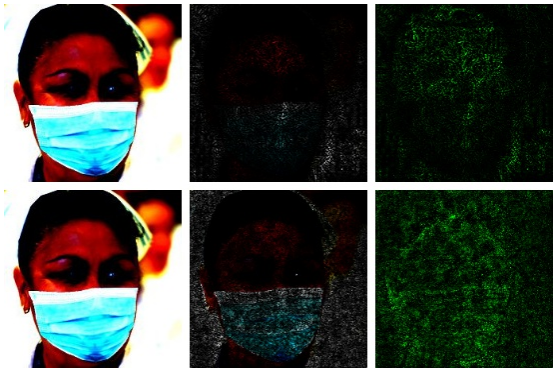

Fig. 15. Implementation of Integrated Gradient on Figure 9

In the 2-by-3 image matrix, from left to right the top row represents the original image, the gradient overlay, and gradient. The bottom row represents the original image, the integrated gradient overlay, and integrated gradient. We are interested in the bottom right image. The pixel in green means that the model believes this pixel is important in predicting the output.

In Figure 15, most of the pixels around the mask are highlighted, meaning the model has learned to predict this image as "correctly masked" because of the presence of the mask. In Figure 16, most of the pixels highlighted are scattered around the image. Our guess is that perhaps the Integrated Gradient has also recognized that the model has learned various other features as well, such as eyes and ears. This is likely due to

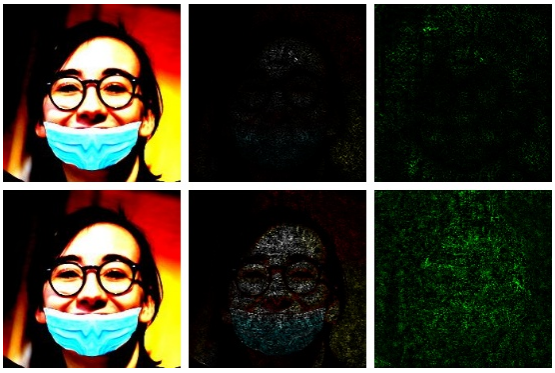

Fig. 16. Implementation of Integrated Gradient on Figure 10

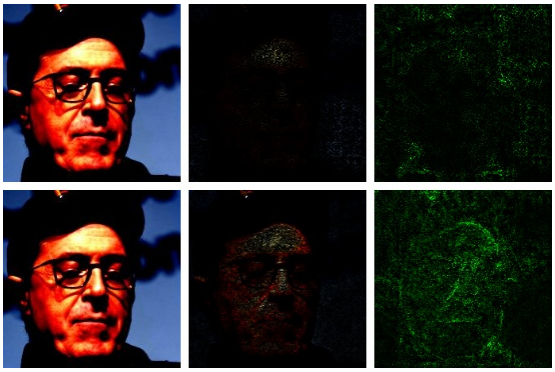

Fig. 17. Implementation of Integrated Gradient on Figure 11

the powerful nature in learning features from ResNet50. In Figure 17, the pixels that are in green constitute area such as the edge of the face which connects to the cap, the node, and the upper lip. This is a very interesting case because not only does it learn other features such nose and edge of the forehead, but it also pays attention to the lip which is important in distinguishing whether a mask is worn. Once again, we believe this is happening because ResNet50 is powerful enough to recognize each facial feature, which includes the upper lip. In summary, while the results of integrated gradient isn't as straightforward as the Grad-CAM, we are still able to connect the result with the success of our model.

Overall, we are satisfied with the high accuracy of the model, and the visualization results which validate our model's approach to classifying the images.

C. Error Analysis

In this section, we would like to provide an example where our model has failed to predict the image correctly. Figure 18 shows the original image.

This image is classified as correctly masked. However, our model has classified it as incorrectly masked. We believe that this is likely due to the fact that some part of the person's nose is not fully covered, resulting in our model detecting the presence of the nose and labeling it as incorrectly masked. Next we'll look at the results of Grad-CAM and Integrated

Fig. 18. An image where our model has incorrectly classified

Gradient on this particular image. Figure 19 and 20 show the result:

Fig. 19. Grad-CAM result on the misclassified image

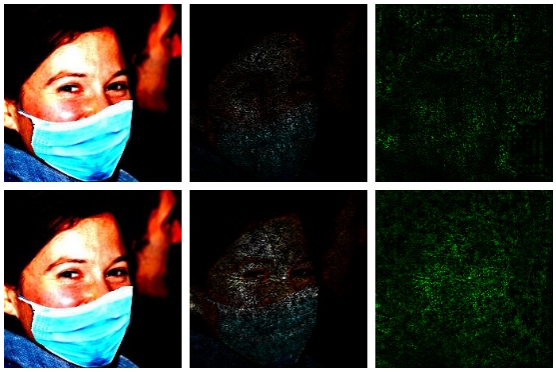

Fig. 20. Integrated Gradient result on the misclassified image

Looking at Grad-CAM result, we observe that the heat map is placed correctly onto the mask. However, it is also noticeable the area of the map has extended to the nose area, which indicate the when considering whether this image contains a person with correctly worn mask or incorrectly worn mask, the presence of nose is a major factor in assisting the model to make the decision, thus resulting in the model predicting this image as incorrectly masked.

Now looking at the result of integrated gradient, though not very easy to locate, we discover that the cheek of this person is being highlighted. When a person is wearing a mask, it's very unlikely for this person's cheek to expose, unless this person is smiling. In this case, the person is smiling. Therefore, we deduce that perhaps our model might not be able to classify an image where a person is smiling but still has his mask properly worn. This is something definitely worth considering in the future for further development and refinement of the model.

D. Discussion and Future Improvement

As mentioned in the previous section, the error example demonstrates that a person smiling might result in model's failed attempt to correctly classify the image. Therefore we definitely need to take this into consideration if we want to further improve our model. Some possible approaches include adding more layers to fine tune the model or training the model with more data, especially images like this.

Although we mentioned the addition of other models in our code, there are also other improvements that can be made. Currently one caveat of implementation is the requirement of an image to make a classification. That being said, an improvement would definitely be to create this classification through a video setting due to settings in which people constantly move around making this task difficult to accomplish. However, the ability to adapted for motion capture is currently far beyond the scope of any of the author's capabilities. To able to do so will require people who are skilled in IT.

VI. Conclusion

In this report, we address a current concern shared by people across the world and point out a common bad habit practiced by the general public, which is wearing mask incorrectly and has jeopardized a lot of business. Then we attempt to mitigate this issue by proposing a face mask detector that will help business owners comply with laws and ensure the survival of their business. Our approach to building such a face mask detector is successful and is proven to work in the intended way. We hope that more face mask detectors can be implemented so that business owner can survive in this rather difficult time and that people can protect not only themselves but also others.

References

[1] A. Rosebrock, *COVID-19: Face Mask Detector with OpenCV, Keras/TensorFlow, and Deep Learning*, PyImageSearch, May 4, 2020. Accessed on: Feb 7, 2021. [Online]. Available: https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/

[2] Cabani, Adnane, et al. "MaskedFace-Net–A dataset of correctly/incorrectly masked face images in the context of COVID-19." Smart Health 19 (2020): 100144.

[3] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[4] Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE international conference on computer vision. 2017.

[5] Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." International Conference on Machine Learning. PMLR, 2017.