Universidade de São Paulo
Instituto de Matemática e Estatística

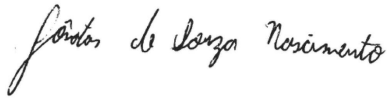
Relatório Científico

Visualização em tarefas de classificação de imagens baseada em aprendizado profundo

Aluno: Jônatas de Souza Nascimento
Professora Orientadora: Nina S. T. Hirata

Processo FAPESP: 2020/07089-0
Vigência: 01/08/2020 a 31/07/2021
Período descrito no relatório: 01/08/2020 a 31/07/2021

São Paulo, julho de 2021



1 Introdução

As Redes Neurais Convolucionais (CNNs) vem crescendo em relevância a cada dia. Mais e mais elas estão se tornando a ferramenta preferencial para diversos tipos de problemas na área de Visão Computacional, desde classificações simples e complexas, até localização de objetos, segmentação semântica e reconhecimento de imagem.

Porém, apesar dos excelentes desempenhos, as CNNs carregam consigo o problema de interpretabilidade. O fato de possuírem muitos parâmetros torna o entendimento sobre o funcionamento dessas redes muito difícil, muito mais que o de outras técnicas de aprendizado de máquina. Na prática, os modelos de Redes Neurais Convolucionais são muitas vezes utilizados quase como uma "caixa preta".

Interpretabilidade é uma característica desejável na maioria das aplicações que empregam técnicas computacionais inteligentes. No caso de técnicas de aprendizado de máquina, em muitas situações deseja-se saber não apenas o escore de predição mas também por que o algoritmo atribuiu tal escore. Ademais, mecanismos que permitam a interpretação do funcionamento de uma rede convolucional podem ser úteis para a melhoria do modelo de predição.

Com o objetivo de entender o que as CNNs "aprendem"a partir dos dados, ao longo dos últimos anos surgiram técnicas de visualização. Essas técnicas têm como objetivo central a geração de imagens entendíveis ao olho humano, de modo a tentar esclarecer quais são os fatores importantes para o aprendizado de uma rede desse tipo.

Neste projeto, alguns desses métodos de visualização foram estudados. O estudo foi restrito a problemas de classificação usando CNNs, e o foco foi dado ao estudo de técnicas que geram mapas de atenção. Os mapas de atenção indicam quais são as regiões da imagem mais relevantes para uma determinada predição.

O estudo envolveu a leitura de artigos referentes a quatro técnicas (Saliency Maps [SVZ14], Smooth Grad [STK+17], CAM [ZKL+16] e Grad-CAM [SCD+17]), a implementação de código para o treinamento de CNNs e visualização dos mapas, e o teste das implementações realizadas.

O bolsista vem estudando conceitos e algoritmos de aprendizado de máquina desde março de 2020 e foi contemplado com bolsa FAPESP a partir de agosto

de 2020. Essa bolsa está vinculada ao projeto de pesquisa regular (processo Microsoft-PITE 2017/25835-9), intitulada *Understanding images and deep learning models* e coordenada pela orientadora.

2 Atividades Realizadas

Em período anterior ao início da vigência da bolsa, o bolsista realizou atividades que visaram a familiarização com alguns conceitos e algoritmos básicos de aprendizado de máquina. Em particular, estudou e implementou os seguintes algoritmos em linguagem Python:

- Perceptron
- Regressão Linear
- Regressão Logística
- Redes Neurais

A seguir, estão detalhadas as atividades realizadas entre agosto de 2020 e julho de 2021, o período coberto neste relatório.

Encontros com grupo de pesquisa Há encontros semanais do grupo ImageU do IME-USP, formado pela orientadora e seus orientandos. Durante esses encontros ocorre a discussão sobre o andamento dos diferentes projetos dos orientandos, permitindo assim que ocorra colaboração mútua entre os alunos. Esses encontros estão ocorrendo de forma virtual e o bolsista participa deles sempre que possível.

Leitura de artigos Para a compreensão das técnicas de visualização estudadas, foram realizadas leituras dos artigos que propuseram as técnicas. Uma descrição das técnicas estudadas é apresentada mais adiante.

Experimentos A parte prática da IC envolve a aplicação das técnicas estudadas. Desta forma, foram realizadas atividades de implementação de redes convolucionais e visualização de mapas de atenção usando o framework `Keras`[1] e a biblioteca de visualização `tf-keras-vis`[2].

[1] `https://keras.io/`
[2] `https://keisen.github.io/tf-keras-vis`

Elaboração de artigo O bolsista colaborou na elaboração de um artigo, juntamente com um bolsista de treinamento técnico e um aluno de mestrado. O artigo trata da aplicação de redes convolucionais para a detecção de COVID-19 em imagens de tomografia de pulmão. O artigo não foi aceito e atualmente está sendo retrabalhado para resubmissão.

3 Progressos

Nesta seção são descritas as técnicas de visualização estudadas, quais sejam:

1. Saliency Map (descrito no artigo "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps" [SVZ14])

2. *SmoothGrad* (descrito no artigo "SmoothGrad: removing noise by adding noise" [STK+17])

3. CAM (*Class activation maps*, descrito no artigo "Learning Deep Features for Discriminative Localization" [ZKL+16])

4. GradCAM (Descrito no artigo "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization" [SCD+17])

Para testar essas técnicas, foram realizadas implementações para treinamento de CNNs e visualização dos mapas de ativação. As implementações foram todas feitas em *notebooks Python*, usando o framework `Keras` (`https://keras.io/`) e a biblioteca de visualização `tf-keras-vis` (`https://keisen.github.io/tf-keras-vis`). Esses *notebooks* assim com outros feitos antes da vigência da bolsa serão disponibilizados posteriormente via `github`, após documentação.

A seguir são descritas as técnicas de visualização estudadas e ao final são ilustrados alguns resultados obtidos com essas técnicas.

3.1 Mapas de Saliência

3.1.1 Motivação

O Mapa de Saliências em Redes Neurais Convolucionais foi primeiramente proposto em 2013 por Simonyan e outros [SVZ14]. A ideia dos autores era encontrar,

dada uma rede neural convolucional já treinada, os pixels de uma imagem que são mais relevantes para a CNN classificá-la como sendo de uma determinada classe.

Mais formalmente, seja $S_c(I)$ a função que define o escore atribuído pela CNN a uma dada classe c. O objetivo é identificar a influência de cada pixel de uma imagem de entrada I_0 para o escore $S_c(I_0)$. O mapa com os escores de cada pixel foi denominado mapa de saliência.

3.1.2 Técnica

A técnica surge da extrapolação da aplicação de conceitos simples de um modelo linear para um modelo mais complexo. Considerando que a classificação é definida por um modelo linear, temos a função de escore:

$$S_c(I) = w^T I + b_c$$

sendo w o vetor de pesos e b_c o "bias"do modelo. Nesse caso, uma possível interpretação é a de que os pesos w definem a importância de cada pixel para o escore relativo à classe c. Também podemos identificar que $w = \frac{\partial S_c}{\partial I}\Big|_{I=I_0}$, ou seja, o vetor de pesos é a derivada de S_c em relação a I para a imagem I_0.

Passando para o caso das CNNs, é sabido que a função escore $S_c(I)$ é altamente não linear, portanto não seria simples de se fazer um paralelo com o resultado anterior. Contudo, dada uma imagem I_0 é possível calcular uma aproximação da função S_c nos arredores de I_0 usando a expansão de Taylor de primeiro grau, isto é, temos:

$$S_c(I) \approx w^T I + b$$

Considerando essa aproximação, podemos fazer a mesma interpretação acima, tendo w como uma derivada:

$$w = \frac{\partial S_c}{\partial I}\Big|_{I=I_0}$$

O peso relativo a cada pixel pode ser disposto em um mapa de mesmo tamanho da imagem, resultando em um mapa de saliência. Uma outra maneira de

interpretar o cálculo do mapa de saliências por meio dessa derivada é entender que a derivada de S_c indica qual pixel da imagem precisa de menos variação para ter um grande efeito no escore. Mais adiante veremos que os pixels mais salientes correspondem em geral à posição do objeto classificado na imagem.

A principal vantagem dessa técnica é que o problema de se identificar quais são as regiões da imagem que mais são relevantes para a predição feita por uma CNN resume-se ao cálculo de uma simples derivada, que pode ser feita pela técnica de retropropagação.

3.1.3 Extração do Mapa de Saliências

Partindo do entendimento da última seção, podemos computar o Mapa de Saliências $M \in \mathbb{R}$ como em [SVZ14]. Primeiro calcula-se w por retro propagação e se reorganiza os valores de w de modo que o número de elementos de M seja o número de pixels de I_0.

No caso de imagens em escala de cinza, o mapa de saliências pode ser calculado como $M_{ij} = \left|w_{h(i,j)}\right|$ sendo que $h(i,j)$ representa o pixel de linha i ecoluna j. Já para imagens com vários canais de cores, para calcular um único Mapa de Saliência para uma determinada imagem, calcula-se $M_{ij} = max_c(\left|w_{h(i,j,c)}\right|)$ sendo c o índice do canal de cor do determinado pixel $h(i,j)$ da imagem.

3.2 Smooth Grad

3.2.1 Motivação

Várias técnicas de visualização, cujo objetivo é salientar áreas da imagem que são relevantes para a classificação realizada pelas redes neurais convolucionais, geram mapas que são bastante ruidosos. Por vezes, o mapa salienta regiões da imagem que ao olho humano parecem ser selecionadas randomicamente.

Portanto, em [STK+17] foi proposta uma técnica que visa reduzir o ruído visual nos mapas, em especial no Mapa de Saliências descrito acima. A ideia dessa técnica é, dada uma imagem de interesse, gerar novas imagens adicionando-se ruído à imagem original, e em seguida calcular um mapa de saliência que consiste da média dos mapas de cada imagem ruidosa gerada. A essa técnica os autores deram o nome de *SmoothGrad*.

3.2.2 Técnica

Em [STK+17], os autores propõem uma possível explicação para o alto ruído dos Mapas de Saliência. De acordo com os autores, o ruído está relacionado ao fato do mapa ser calculado através de uma derivada. É conhecido que o cálculo de derivadas discretas especialmente em escalas pequenas pode acarretar um alto ruído. Sendo assim, o ruído que vemos nos mapas de saliência podem ser simplesmente variações sem significado na derivada local de S.

Como S tem a tendência de possuir variações bruscas que não tem necessariamente significado, o gradiente de S em qualquer ponto da imagem terá menos significado que a média local do gradiente de n imagens geradas com um ruído. Isso sugere que uma maneira de diminuir os pontos sem significados dos mapas de saliência é basear esses mapas não no gradiente direto de S, mas em uma suavização do gradiente.

Como o cálculo direto de uma média local de um espaço de várias dimensões como uma imagem tem um custo muito alto, a maneira utilizada para lidar com esse problema foi o cálculo de uma aproximação estocástica dessa média. Na prática se calcula o Mapa de Saliências M_{ij} para n amostras aleatórias nos arredores da imagem I, e no fim toma-se a média desses mapas resultantes. Isso pode ser expresso matematicamente da seguinte maneira:

$$\hat{M}_{ij}(I) = \frac{1}{n}\sum_{1}^{n} M_{ij}(I + \mathcal{N}(0, \sigma^2))$$

na qual $\mathcal{N}(0, \sigma^2)$ é uma distribuição gaussiana com média 0 e desvio padrão σ, correspondendo ao ruído adicionado à imagem I.

3.2.3 Escolha dos Hiper-parâmetros

A técnica *SmoothGrad* envolve 2 hiper-parâmetros, sendo eles o número de amostras n e o desvio padrão σ do ruído. Como na escolha dos hiper-parâmetros de redes neurais comuns, não há regra e nem demonstração que nos guie para essas escolhas. As decisões devem ser tomadas de maneira empírica e devem variar para cada tipo de imagem/modelo.

Segundo os testes realizados em [STK+17], bons valores para σ variam entre $0,2$ e $0,3$. Já para n a tendência é que quanto maior for o número de amostras,

menos ruído o mapa de saliências terá (como o esperado), porém, os resultados mostram que a partir de $n = 50$, as mudanças deixam de ser significativas.

3.3 Class Activation Mapping

3.3.1 Motivação

Os autores de [ZKL+16], percebendo o alto desempenho das Redes Convolucionais em tarefas de reconhecimento e localização de objetos, implementaram uma técnica que ia além de apenas classificação, mas também identificava quais são as regiões da imagem que estão sendo utilizadas na discriminação.

Os autores baseiam essa ideia em trabalhos anteriores que tentavam responder o mesmo problema. Os autores de [BBAT16] propõem, por exemplo, uma maneira de ocluir regiões da imagem, e assim gerar mapas que mostram qual é a relevância de cada pedaço da imagem para a classificação final. Já [OBLS14], mostrou ser possível chegar em resultados de localização de objetos através de uma técnica que transfere os parâmetros de uma CNN fonte para uma CNN alvo, e ao calcular a saída da CNN alvo por múltiplos pedaços dessas imagens.

Os autores de [ZKL+16] discutem que apesar de os resultados anteriores a ele serem promissores, eles requerem muitos passos para chegar na localização das áreas relevantes da imagem, tornando difícil sua escalabilidade para as grandes aplicações comuns de CNN. Por isso, surge a proposta do Class Activation Mapping, que tem a pretensão de chegar em um resultado melhor com um cálculo mais simples.

O Class Activation Mapping, diferente das outras técnicas estudadas até aqui, é aplicado apenas a Redes Convolucionais com um tipo de arquitetura. Apesar das aplicações mais comuns atualmente utilizarem camadas densas ao fim das camadas convolucionais em CNNs, os autores desenvolveram um método de visualização baseado em redes neurais que possuam apenas camadas convolucionais e que utilizam um Global Average Pooling (GAP) ao fim destas, de modo a gerar as classificações.

3.3.2 Global Average Pooling

Global Average Pooling (GAP) é uma técnica que substitui a necessidade de camadas densas no fim de uma Rede Neural Convolucional. Especificamente, é calculada a média de cada feature map da última camada convolucional as mesmas são diretamente ligadas à camada de ativação softmax.

Algumas vantagens surgem por conta desse método, como a diminuição de overfitting e robustês a translações de imagens na entrada. Apesar dessas questões, o principal motivo da escolha do GAP para os autores, é a percepção de que vários trabalhos anteriores [ZF13] [ZKL+15] estavam se preocupando em visualizar apenas as camadas convolucionais das redes neurais, ignorando as camadas densas.

Julgando que a perda de desempenho ao se utilizar o GAP em detrimento de camadas convolucionais era pouco significativa, Zhou fundamenta o Class Activation Mapping com o GAP, permitindo assim que se visualize completamente suas redes neurais, entendendo-as portanto, do início ao fim.

3.3.3 Cálculo do Class Activation Mapping

Sendo $f_k(x, y)$ a ativação do mapa k da última camada convolucional de uma rede neural para uma imagem I, o resultado do GAP é $F_k = \frac{1}{Z} \sum_{x,y} f_k(x, y)$

Sendo w_k^c o peso entre o mapa k e ao nó de saída referente à classe c, a entrada S_c do softmax é $S_c = \sum_k w_k^c F_k$. Portanto, esse termo w_k^c mostra a importância do mapa k para essa classificação da imagem. Sendo assim, define-se o mapa de ativação de classes (CAM) pela seguinte fórmula:

$$M_c(x, y) = \sum_k w_k^c f_k(x, y)$$

Cada f_k é um mapa que representa um padrão visual que colabora para a ativação de S_c. Então CAM é simplesmente uma combinação linear ponderada por w_k^c desses padrões visuais. Portanto, $M_c(x, y)$ indica a importância do pixel (x, y) para a classificação da imagem I como sendo da classe c.

3.3.4 Desvantagens

A principal limitação do método de Class Activation Mapping é que ele só pode ser calculado para Redes Convolucionais com uma arquitetura específica, a saber, os feature maps devem preceder diretamente a camada final de classificação.

Para aplicar o CAM em Redes que não seguem essa estrutura, a opção é modificar a arquitetura. Por exemplo para casos de CNNs que possuem camadas densas após as camadas convolucionais, a solução é substituí-las por camadas convolucionais convencionais, e após isso treiná-las novamente.

A despeito dessa solução permitir a aplicação desse método em diversas Redes Neurais diferentes, ela carrega consigo limitações. Além dessa modificação na arquitetura frequentemente diminuir a acurácia das CNNs para tarefas simples (como classificação de imagens), existem problemas que essa arquitetura não é capaz de resolver (como detecção de objetos).

3.4 Grad CAM

3.4.1 Motivação

Das limitações da técnica de Class Activation Mapping, surge a necessidade de uma melhoria. Os autores de [SCD+17] propõem então, uma técnica baseada no CAM, porém generalizada de modo que ela possa ser aplicada a uma CNN de arquitetura qualquer, resolvendo a necessidade de alteração na arquitetura que existia na técnica anterior.

3.4.2 Técnica

Para o caso de uma arquitetura qualquer de CNN, a "relevância"de cada *feature map* é calculada como sendo o gradiente da saída y^c com respeito a cada *feature map*. Definimos α_k^c como o GAP desse gradiente, ou seja:

$$\alpha_k^c = \sum_{x,y} \frac{\partial S_c}{\partial f_k(x,y)}$$

Portanto, de modo semelhante ao CAM, o grad-CAM é uma combinação linear ponderada dos feature maps, com a diferença de que é seguida por um ReLU:

$$M_c(x,y) = ReLU\left(\sum_k \alpha_k^c f_k(x,y)\right)$$

Portanto, com esse método chegamos a um mapa de calor da mesma natureza do gerado pelo CAM, com a diferença de que este pode ser calculado para CNNs de arquiteturas diversas.

3.5 Exemplos

Para gerar os exemplos de mapas adiante, considerou-se o problema de classificação entre cães e gatos. Sendo assim, foi treinada uma rede utilizando o modelo pré treinado VGG-16 como base, adicionando apenas uma camada densa conectada com a última camada convolucional do VGG, e todos os parâmetros foram treinados.

A arquitetura do modelo portando, ficou da seguinte forma:

- Camada de entrada (150,150)
- Duas camadas convolucionais (150, 150, 64)
- Uma camada de Max Pooling
- camadas convolucionais (75, 75, 128)
- camada de Max Pooling
- camadas convolucionais (37, 37, 256)
- camada de Max Pooling
- camadas convolucionais (18, 18, 512)
- camada de Max Pooling
- camadas convolucionais (9, 9, 512)
- camada de Max Pooling
- camada densa com dropout de 0.5 (256)
- Saída

O dataset utilizado para o treinamento da CNN, foi o dataset da competição `Dogs vs. Cats`[3] do kaggle. Ele possui 25 mil imagens de treinamento, sendo metade de cada classe.

[3] `https://www.kaggle.com/c/dogs-vs-cats`

A figura 1 mostra exemplos de Mapas de Atenção calculados para 3 imagens do dataset do problema de classificação entre cães e gatos.

Podemos ver no caso do Mapa de Saliência, que os pontos mais claros e mais salientados concentram-se sobre os animais, e apesar de visivelmente esse método possuir muito ruído, é possível observar uma tendência da CNN de dar uma importância maior aos pixeis na região da cabeça dos animais.

Como o esperado, o SmoothGrad é uma melhoria do método anterior. Comparando as imagens 1d e 1g por exemplo, é visível como o ruído diminui consideravelmente, fazendo com que os pixels mais claros se concentrem melhor, e a área relevante para a classificação fique mais clara.

O grad-CAM também demonstra o resultado esperado. Por meio do mapa de calor gerado, vemos que nas imagens 1j, 1k e 1l a rede neural destaca as regiões da cabeça do animal para realizar a classificação. Isso pode revelar uma tendência do aprendizado da rede.

Embora os métodos de visualização utilizem abordagens bem diferentes, as imagens obtidas demonstram que eles entregam resultados parecidos. As imagens 1h e 1k dão destaque a regiões parecidas da imagem do gato.

4 Plano de trabalho

Estamos solicitando a renovação da bolsa para dar prosseguimento a este projeto de iniciação científica. O período de extensão solicitado é de 6 meses, para o desenvolvimento das seguintes atividades:

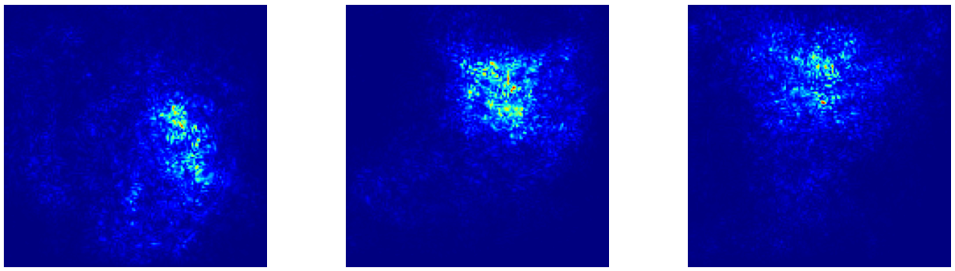
1. Realizar o treinamento de uma CNN para a classificação de imagens de tomografia do pulmão em COVID-19 positivo ou negativo.

2. Aplicar as técnicas de visualização estudadas sobre a CNN treinada e compará-las.

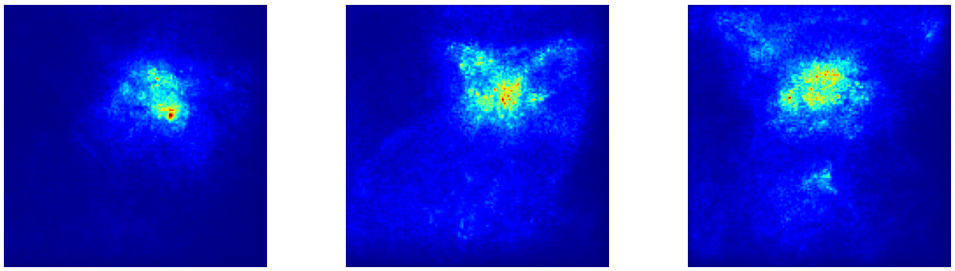
3. Avaliar as visualizações do ponto de vista da aplicação, isto é, avaliar se a CNN está de fato considerando características consideradas indicativos de COVID-19 (por exemplo, presença de regiões com aspecto de vidro fosco).

4. Colaborar na elaboração do artigo sobre classificação de imagens de tomografia quanto a COVID-19 positivo ou negativo.

(a) Imagem Original (b) Imagem Original (c) Imagem Original



(d) Saliency Map (e) Saliency Map (f) Saliency Map



(g) SmoothGrad (h) SmoothGrad (i) SmoothGrad

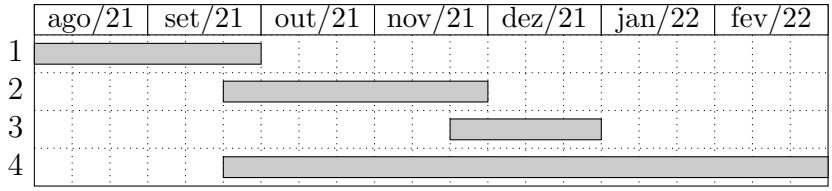
(j) Grad-CAM (k) Grad-CAM (l) Grad-CAM

Figura 1: Mapas de Atenção gerados para o problema de classificação entre Cachorros e Gatos

As atividades 1 a 3 serão úteis para aprofundar, consolidar e validar o conhecimento adquirido até este momento. Em especial, o bolsista terá a oportunidade de trabalhar um problema real de classificação de imagens integralmente, desde o treinamento da CNN até a avaliação de seu desempenho, incluindo possíveis interpretações.

Por outro lado, a atividade 4 permitirá que o bolsista tenha contato com um processo de pesquisa, desenvolvido por um time (no caso, a orientadora e mais dois bolsistas). De fato, o bolsista já participou das discussões e elaboração de uma primeira versão do artigo. Uma vez que o escopo da pesquisa está sendo estendido visando uma segunda versão do artigo, será uma excelente oportunidade para o bolsista ampliar sua experiência em pesquisa em equipe.

Cronograma



Referências

[BBAT16] Loris Bazzani, Alessandro Bergamo, Dragomir Anguelov, and Lorenzo Torresani. Self-taught object localization with deep networks, 2016.

[OBLS14] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 06 2014.

[SCD+17] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[STK+17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. SmoothGrad: removing noise by adding noise, 2017.

[SVZ14] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

[ZF13] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013.

[ZKL+15] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns, 2015.

[ZKL+16] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, 2016.