國立臺灣大學電機資訊學院資訊工程學系
碩士論文
Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

藉由覆蓋雜訊了解深度網路所見之方法

Uncovering What Network Sees by Noise Covering

廖建棋
Chien-Chi Liao

指導教授：吳家麟 博士
Advisor: Ja-Ling Wu, Ph.D.

中華民國107年7月
July 2018

doi:10.6342/NTU201801411

國立臺灣大學碩士學位論文
口試委員會審定書

藉由覆蓋雜訊了解深度網路所見之方法

Uncovering What Networks See by Noise Covering

本論文係廖建棋君（學號 R05922002）在國立臺灣大學資訊工程學系完成之碩士學位論文，於民國 107 年 7 月 5 日承下列考試委員審查通過及口試及格，特此證明

口試委員：吳家麟

（指導教授）

朱威達　鄭文皇

胡敏君　陳祝嵩

系主任　莊永裕

doi:10.6342/NTU201801411

摘要

我們還不夠瞭解深度學習是如何做決策的，以至於我們沒辦法完全信任它。因此，我們提出一個以最佳化為基礎的方法，試著以視覺化的方式找出深度網路對影像分類時的依據。此方法將不斷變化的雜訊覆蓋在輸入影像上，使得被辨識到的特徵可以被標示出來。我們用類似ImageNet自然影像的資料集來與其他方法進行比較。結果顯示，我們的方法得出來的顯著圖在視覺品質上較佳，以及與辨識結果有較高的相關性。我們將此方法套用到三種知名的深度卷積網路中間層，探索影像辨識的過程，並得到一些見解。此外，我們的方法在實作上並不需要修改現有的模型，所需的最佳化目標函數可以容易地用現有的深度學習套件實作。

doi:10.6342/NTU201801411

Abstract

The mystery of how deep neural networks (DNNs) make decision has discouraged us to fully trust them. This writeup presents an optimization-based method for visualizing the clues that may explain the reason behind the DNN classified results of an image. Our method masks the inputs with varying noises to extract the truly effective and recognizable features. We did the empirical comparisons with related works on ImageNet like dataset, and the obtained saliency maps provide better visual quality and higher relevance score in general. We found some insights into the recognition processes of three notable CNNs by applying our approach to the corresponding intermediate layers. Besides, the realization of our approach doesn't require any modification of existing models and the cost function of the optimization process can be easily formulated based on modern deep learning libraries.

doi:10.6342/NTU201801411

List of Contents

doi:10.6342/NTU201801411

List of Figures

doi:10.6342/NTU201801411

List of Tables

doi:10.6342/NTU201801411

Chapter 1 Introduction

For the past few years, due to big data and the great progress in hardware, DNNs have become popular and successful in image classification, speech recognition, and natural language processing, just named a few. However, an effective deep network is usually big and complex. Its inner representation of data often is in the form of high dimensional tensors and processed by composite nonlinear operators, which makes the direct explanation of the result difficult, so DNN is usually considered as a black box. People won't trust an outcome that is not explainable, especially in applications that the mistakes are costly such as self-driving cars or medical imaging classification [8]. Without an explanation, we won't know what is wrong in the network and have no clues to debug. Therefore, the understanding of a network's decision is necessary and worthy of doing.

In this work, we tried to understand how decisions are made by convolutional neural networks (CNN). Some existing works [5, 6, 7, 11] used backpropagation-like mechanisms to distribute the classification scores back to inputs. The corresponding computation is fast but the associated saliency maps (will be formally defined in sub-section 3.23) are often scattered and noisy. On the other hand, Perturbation-base methods [8, 12] often produce blobby and vague saliency maps. Most importantly, it is observed that saliency maps generated from both types don’t match the structures of the associated input images. This can be an issue when researchers are figuring out what geometric structures that CNNs actually focuses on. In our experiments, we also noticed that some of the obtained results can be misleading.

doi:10.6342/NTU201801411

In this thesis, we proposed an optimization-based method called “Noise Mask” to uncover what a CNN actually see. Our approach regards this uncovering problem as a minimization problem which tries to progressively discard information from the input but keeps the outcomes unchanged so as to reveal the underlying recognized features. The residual information is kept as the saliency map. Interestingly, it is found that Noise Mask provides better visual quality and higher relevance accuracy on average than that of other methods. Noise Mask can also be applied to a CNN’s intermediate layers, which provide useful insights to understand the corresponding recognition processes.

The realization of Noise Mask employs the backpropagation computation provided by modern deep learning libraries. The only efforts are inserting two assistance layers into the network and defining the corresponding cost function. It's much easier than that of the methods given in [5, 6] which involving the backpropagation details.

This thesis is organized as follows: Some background and the evolution of decision explanation on CNNs are introduced in Chapter 2. Chapter 3 preludes the intuition of our solution, and then crafts and polishes the idea progressively to define the adopted strategy. In Chapter 4, we did the empirical comparisons with related approaches to show the latent flaws in them. In addition, we addressed the general recognition processes in notable CNNs with different architectures. Chapter 5 summarizes the observations we obtained in this thesis and points out our possible future improvements.

doi:10.6342/NTU201801411

Chapter 2 Related Works

This chapter reviews some previous works relating to the understanding of neural networks. Here, we assume the networks are well trained rather than discuss the evolution of model training from scratch.

The operations conducted in neural networks are mostly simple. The inputs are fed into a set of linear functions followed by another set of nonlinear functions, which are usually ReLUs or sigmoids, this forms a single layer. Repeating this process forms a sequence of layers. In CNNs, we replace the set of linear functions with convolution filters and insert the pooling operation between layers to reduce the dimension of intermediate outputs. Though each operation seems simple, the composite nonlinearity makes the networks hard to be analyzed. Besides, millions of model parameters make precise explanation of network's behavior nearly impossible. Therefore, developing other strategies to comprehend CNNs, instead of finding direct interpretation seems worthy of doing. As addressed in [1], there are two major categories of CNN behavior's analysis in the literature.

2.1 Network Interpretation

The first category is Network Interpretation. Network interpretation provides a meaningful mapping between the network units, which can be a group of layers or a set of filters, and a understandable target media, which can be a set of words or images [2]. This mapping explains the function of each unit and gives us insights into how the networks organize high level abstract concepts. In their reverse engineering works, researchers tried to synthesize an image which optimizes the state of the targeted neuron

doi:10.6342/NTU201801411

[4, 14, 15]. To improve the visual quality of the synthesized representatives, they tried to reduce high-frequency artifacts by blurring [4], adding regularizers [14], or even training a generator which produces pleasing samples [18]. Researchers also found that a single neuron can respond to diverse visual concepts [9, 10], which means there is no particular visual or semantic representation that can characterize a specific neuron, completely.

2.2 Decision Explanation

The second category is Decision Explanation, which is the main focus of this thesis. Here, decision means the classification result when an input was given. The goal of decision explanation is to highlight input features that are responsible for the obtained classification results. A heatmap, which is also known as “saliency map” [15], will be produced to explain the outcome of an image classification in this work. Two prominent types of methods will be addressed in the rest of this sub-section.

2.2.1 Backpropagation-based Methods

Simonyan et al. [15], the pioneer of decision explanation of CNNs, tried to simplify the corresponding nonlinear model by linear approximation. Each weight of the linear function proportions to the importance of each input pixel. Therefore, the saliency map is obtained by taking the gradient of the leading class function with respect to pixels of the input image. Some other similar approaches, such as Deconvolution [12] and Guided Backpropagation [16] modified the gradient computation of the nonlinearity to control the gradient paths. However, the physical

doi:10.6342/NTU201801411

meaning of the gradient indicates the fact that the input adjustment just forcing the image becomes more similar to a specific class of images, rather than finding the answer to which part of the image makes it belong to the class [1]. Moreover, when the input falls into the flat region of the decision function, there is no significant gradient. This problem was termed "saturation" in [7].

The notion of contribution or attribution comes up after the availability of the simple gradient approach [6]. The goal of this kind of methods is to distribute the interested output back to the input. Each pixel is assigned a value (or score) to quantify how relevant this pixel is to the output or how much this pixel contributes to the output, and the sum of these contributions is equal to the output score. Such kind of approaches includes the Layer-wise Relevance Propagation (LRP) [6], DeepLIFT [5], and Integrated Gradients [7, 11].

LRP and DeepLIFT defined rules for score distribution. Each kind of operations corresponds to different rules. These rules receive contribution values (or scores) from their upper layers and then redistribute them to their lower layers. In other words, the algorithms redistribute one specific class's score or the relative score to a baseline score, which is the case of DeepLIFT, back to the inputs layer-by-layer. There are different choices of linear and/or nonlinear rules since they all make sense in some degree. However, it is not trivial to implement them because we have to customize the backpropagation mechanisms adopted in the existing deep learning libraries.

There is a notion called "reference" in both Integrated Gradient and DeepLIFT. The idea is that we don't directly distribute the prediction score, but instead, we compare the input and output with some reference input and the corresponding output to analyze the difference. This makes the contribution analysis more rational because this

doi:10.6342/NTU201801411

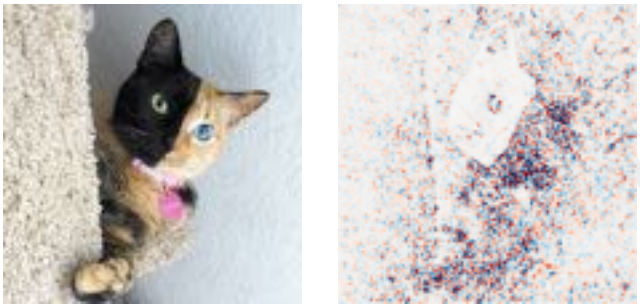


Figure 1: Integrated Gradients with black reference input produces a confusing saliency map.

approach does introduce the comparison with a baseline. In the image classification task, we often choose a black image as the reference, since it looks neutral in the visual sense. However, as long as the input has the same value as the reference input, we will get zero contribution from the difference. For example, if a black cat image is analyzed with a black reference image, there is no input difference over the cat itself, so it would result in wrong explanation. Figure 1 shows this drawback on applying Integrated Gradients.

The Idea behind Integrated Gradients comes from the observation of experiments. Sundararajan et al [7] did the interpolations between the reference input and the actual input, and these interpolations are termed as "counterfactual images". These images form a straight path from the reference input toward the actual one, so it will cross the decision boundary of the classifier. They recorded the gradients with respect to these counterfactuals, which are called "interior gradients". Gradient becomes stronger when the path is crossing the decision boundary, and going weaker and approaching to saturation when it is away from the boundary. We can summarize the interior gradients by averaging them; this is why the summary is called the Integrated Gradients. Each component of integrated gradient maps to each pixel of the image and serves as the corresponding contribution score. Mathematically, it can be showed that

doi:10.6342/NTU201801411

the sum of integrated gradient components is equal to the relative output, which obeys the contribution axiom given in [11]. Integrated gradients would be dominated by the gradients at the decision boundary, so intuitively, they figured out the key features of the input.

2.2.2 Perturbation-based Methods

These approaches alter an input or a set of inputs and then measure the change of the output. When they are applied to CNNs, a patch is prepared to occlude with the input image. When the patch slides through the image, we record the output changes with respect to the occluded locations which then form the saliency map. The early approach [12] chose grey as the patch color, whereas the recent approach [8] drew patch samples from a multivariate normal distribution and did rigorous statistics to simulate the absence of targeted pixels. However, unlike backpropagation-based methods, these methods are slow since the number of forward passes is proportional to the image size. The rigorous approach even needs more passes, since the patch is sampled more than once at the same location. Furthermore, the resulting saliency map is sensitive to the patch size, so the analysis has better be done in multiple trials with different patch sizes.

2.2.3 Evaluation Methods

A saliency map is good or not depends on whether it captures the key feature for conducting the recognition task. The easiest way is to evaluate the corresponding effects by eyes, but the features identified by humans may not be the same as the ones by machines. Bach et al. [17] introduced the Most Relevant Fist (MoRF) algorithm, which

doi:10.6342/NTU201801411

replaces features in the descending sorted order of contribution values and records the output changes iteratively. The original choice of the replacement was a patch drawn from the uniform distribution, whereas recent practice replaced a few pixels in each iteration. Over iterations, a curve is constructed from the records. If a saliency map is good, this curve should decline rapidly. Additional measures such as the Least Relevant First (LeRF) and the Area Between Perturbation Curves (ABPC) will also be examined in our experiments.

A good saliency map should only highlight the relevant regions. Saliency maps generated from backpropagation-based methods often contain irrelevant noise, which is confusing and misleading. As suggested in [17], we can measure the noisiness or randomness of an image by applying lossless compression and compare the size of the compressed file. On the basis of this performance index, we can show that our saliency map is less noisy than those obtained by other related works.

doi:10.6342/NTU201801411

Chapter 3 Our Approach

3.1 Motivation

We don't follow the contribution axiom obeyed by backpropagation-based methods. Instead, the goal of our approach is to find the most influencing regions in the input image. The most influencing region means that other regions are less effective and whose constituents can be replaced by other values. Here, we call the less influencing regions backgrounds. This idea is similar to that of the occlusion-based methods, but the latter approaches only consider a small region at a time. The main idea of our work based on the assumption that the CNN model can tolerate some changes in the input image except the target subject, and therefore, those non-subject modifications won't affect the output much.

3.2 Formulation

We transform the above-mentioned idea into an optimization and/or a compression problem. The problem is: where are the least regions that must be preserved, i.e. others can be altered, such that the outputs won't change too much? In other words, we are trying to discard as much information from the input as possible, but still keeping the outputs nearly untouched.

We consider this problem with two different performance terms. The first is the preservation term, which measures the amount of the influencing area, and the corresponding original input kept by this area. The second term is the loss term, which measures the difference between original outputs and the outputs after the modification.

doi:10.6342/NTU201801411

Clearly, it is a minimization problem, since we are trying to minimize the influencing area and the output difference, but these two terms are against each other. Given an input image X and the model output function f, the system's characteristic function can be formulated as:

$$\min_{\hat{X}} \mathscr{P}(X, \hat{X}) + \beta \cdot \mathscr{L}(f(X), f(\hat{X})), \tag{1}$$

where $\mathscr{P}$ is the preservation term and $\mathscr{L}$ is the loss term. β is the weighting constant between these two terms. We try to find an optimal input $\hat{X}$* which minimizes (1). $\hat{X}$ is a synthesized image that discards the background but preserves the subject. In the following sub-sections, we will discuss the details of these two terms, respectively.

3.2.1 The Preservation Term

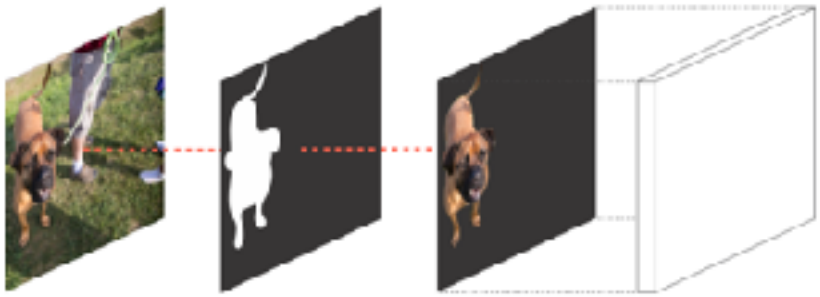


Figure 2: Masking the input image before feeding it into a CNN.

This term measures the area of the influencing region. However, we must define how $\hat{X}$ is generated first. The intuition is inserting a mask which controls what regions should pass into the classifier and what regions can be discarded. Figure 2 illustrates an example of the mask, where the white region preserves the subject while the black region blocks the background. Therefore, the white area of the mask is the influencing

doi:10.6342/NTU201801411

area. We use the sigmoid function to realize the mask function [19] since it maps real values into the interval between 0 and 1, so the constrained specification can be ignored. That is,

$$M = \sigma(V).$$

Where V is the matrix of variables used to minimize (1) and M is the resulting mask. Therefore, the preservation term can be defined as:

$$\frac{1}{n^2}\sum_{i,j} M_{i,j} \quad , \tag{2}$$

where we assume the image dimension is $n \times n$. We do the average over the area since the input size varies along with the networks.

Originally, the black region of the mask represents removing the effect of the associated inputs, but in practice, we cannot provide nothing to the network. We choose the noise drawn from a uniform distribution as the filler because it can reproduce high variety and randomness properties demanded to be the background. We also experimented on normal distributions in Appendix A, and the results are similar. Then, we can synthesize $\hat{X}$ by consolidating the input image and the filling noise according to the chosen mask. Figure 3 illustrates this consolidation.

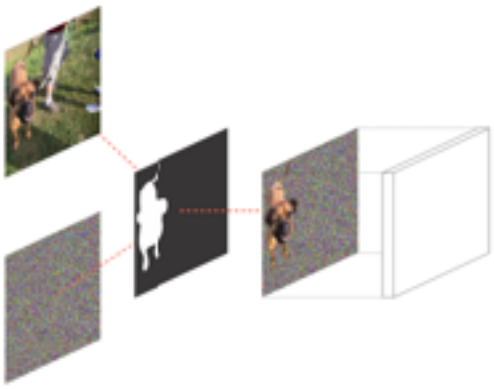


Figure 3: Blending the input image and the noise background.

doi:10.6342/NTU201801411

However, if we use the same noise background during the whole optimization process, dependency on this specific chosen background will be produced. We want the background to differ at each iteration, so there is no consistent background providing a certain pattern that the optimization will make use of. Therefore, the synthesized input $\hat{X}$ can be computed as:

$$\hat{X} = X \odot M + \texttt{uniform}_t(0,255) \odot (1 - M), \tag{3}$$

Where the function $\texttt{uniform}_t(\cdot, \cdot)$ produces a colored noise image and t denotes the iteration number indicating the uniqueness of the noise at each iteration.

3.2.2 The Loss Term

The loss term measures the difference between the original outputs and the outputs after the modification. We not only consider one output that represents a specific class since all outputs represent a specific state that the inputs associated with. Without loss of generality, we simply apply the mean squared errors (MSE) to measure the difference; that is

$$\frac{1}{N}\sum_{i}^{N}(\texttt{FeatMap}_i^l(X) - \texttt{FeatMap}_i^l(\hat{X}))^2. \tag{4}$$

where N is the number of outputs. It should be noted that we can't apply this to the softmax outputs directly since softmax does the normalization over its inputs. Notice that it is possible to have the same outputs when the inputs are very small, which is not what we want. Therefore, `FeatMap`, in (4), stands for the outputs of a layer before the softmax is applied, and l is the index of that layer. We will see what happens when we apply it to different network layers in our experiments.

doi:10.6342/NTU201801411

3.2.3 The Balancing Weight

We noticed that the scales of the two terms are different. The range of the preservation term is between 0 and 1, but the loss term is an average of squared differences and its scale depends on the layer index and the network architecture. The weight β seems hard to decide; therefore, we must set a baseline which suits different networks such that we can adjust β accordingly.

A recommended baseline is that we compute the largest loss value at first and then use it to normalize the loss term. This upper bounding rescales the loss term into the level as the preservation term. The largest loss value occurs when no subject exists in the inputs. Thus, we feed a noise image into the network to compute the baseline of β. That is,

$$\beta_0 = 1/(\frac{1}{N}\sum_{i}^{N}(\texttt{FeatMap}_i^l(\texttt{uniform}(0,255)) - \texttt{FeatMap}_i^l(X))^2).$$

Finally, by combining explicit definitions of the preservation term (2) and the loss term (4), we can reformulate (1) into the following:

$$\arg\min_{M} \frac{1}{n^2}\sum_{i,j} M_{i,j} + \beta\frac{1}{N}\sum_{i}^{N}(\texttt{FeatMap}_i^l(X) - \texttt{FeatMap}_i^l(\hat{X}))^2. \qquad (5)$$

The optimal mask M^*, obtained from (5), is the saliency map that we're looking for. The initialization of M should be close to 1 since it's not easy to recover the subject from the outputs that contain no spatial information.

We name our approach "Noise Mask". Noise Mask avoids some disadvantages existed in related works. It doesn't require the customization of backpropagation

doi:10.6342/NTU201801411

calculation [5, 6] and prevents the misleading caused by the reference input [5, 7]. Also, the iteration number till to convergence is usually smaller than that of the perturbation methods [8, 12] need. We will demonstrate other advantages through experiments, in the next section.

doi:10.6342/NTU201801411

Chapter 4 Experiments

4.1 Settings

We implemented the proposed Noise Mask approach via Tensorflow [20] and ran on the machine with an Nvidia GTX1060 Graphics card. The balancing weight is set to β_0 and `FeatMap` is chosen to be the last output just before the softmax layer. We adapted LazyAdamOptimizer with learning rate 0.015 and the rest of parameters are set to their default values. In average, it can find a saliency map with dimension 224 by 224 in 5000 to 15000 iterations, which is equivalent to 0.7 to 2 minutes on a small model like Inception V1 [21].

Figure 4 shows some experimental outcomes resulted from Inception V1. Besides, there are some findings worthy of mentioning: First, the saliency map of the library image is the striped pattern of bookshelves. Second, the chin and mouth related areas are reserved in the saliency pattern of lab coat image, and it implies there is a high occurrence of a person wearing the lab coat in the training data. Finally, though the color of the background in the lynx image is similar to that of the lynx itself, Inception can still recognize and locate the lynx precisely.

4.2 Comparison

In this sub-section, we compare the experimental outcomes of Noise Mask with those of the other methods. We use the toolkit provided by Ancona et al. [3] to conduct our comparison because it realized most of the decision explanation methods, and

doi:10.6342/NTU201801411

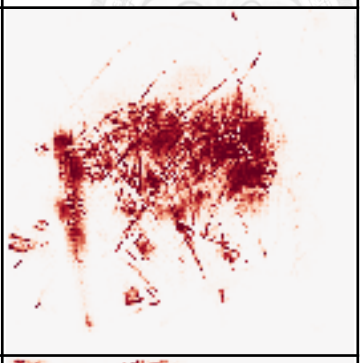
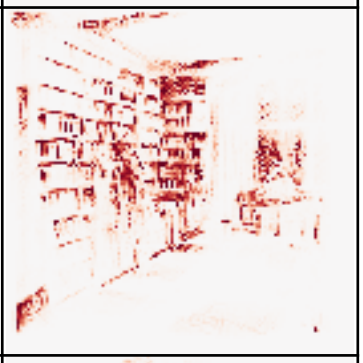
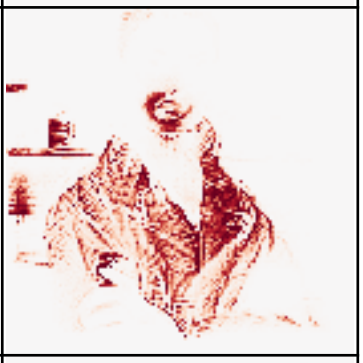
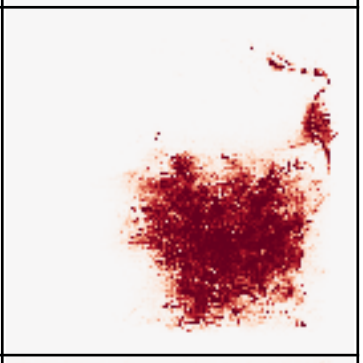
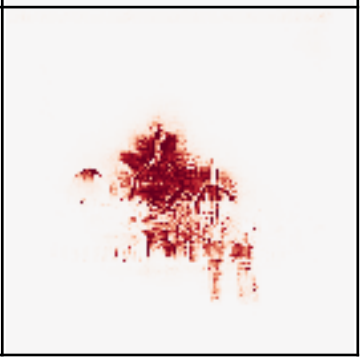
top class: score	original image	saliency map
European gallinule: 99.9%		
library: 98.9%		
lab coat: 99.9%		
lynx: 77.3%		
mosque: 100%		

Figure 4: Example snapshots of saliency maps generated by Noise Mask.

doi:10.6342/NTU201801411

the recommended configurations are adopted. We do the following comparisons on Inception V3 [23] with the image of size 299 by 299 pixels. The dataset we used is provided by Kaggle [22], and we randomly chose 200 images to do the statistics.

4.2.1 Visual Comparison

Figures 5 and 6 give some example snapshots obtained in our experiments. At the first glance, most of the saliency maps obtained from backpropagation-based methods contain noisy saliency that is scattered and not relevant to the subject, but the saliency maps generated from Noise Mask are much clearer. Notice also that the results obtained from the perturbation-based method are blurred since the size of the recommended occluding patch cannot be too small.

4.2.2 The Average Nosiness and The Structural Similarity

According to our previous discussion, we want to measure the average nosiness and the structural similarity of the results to verify that Noise Mask does provide better visual quality than other methods. As we addressed in the related works, lossless image compression can help us to measure the noisiness. Table 1 shows the average PNG file sizes on grayscale saliency maps of each method. Noise Mask indeed produces the smallest file size, meaning that it generates less high-frequency patterns such as scattered spots.

Table 1: Average PNG file sizes of grayscale saliency maps of different methods.

Method	deeplift	intgrad	elrp	sensitivity	perturbation	noise mask
average PNG file size(KB)	75.3±5.5	77.5±4.5	76.8±4.7	75.5±3.6	32.3±3.8	**30.2±7.5**

doi:10.6342/NTU201801411

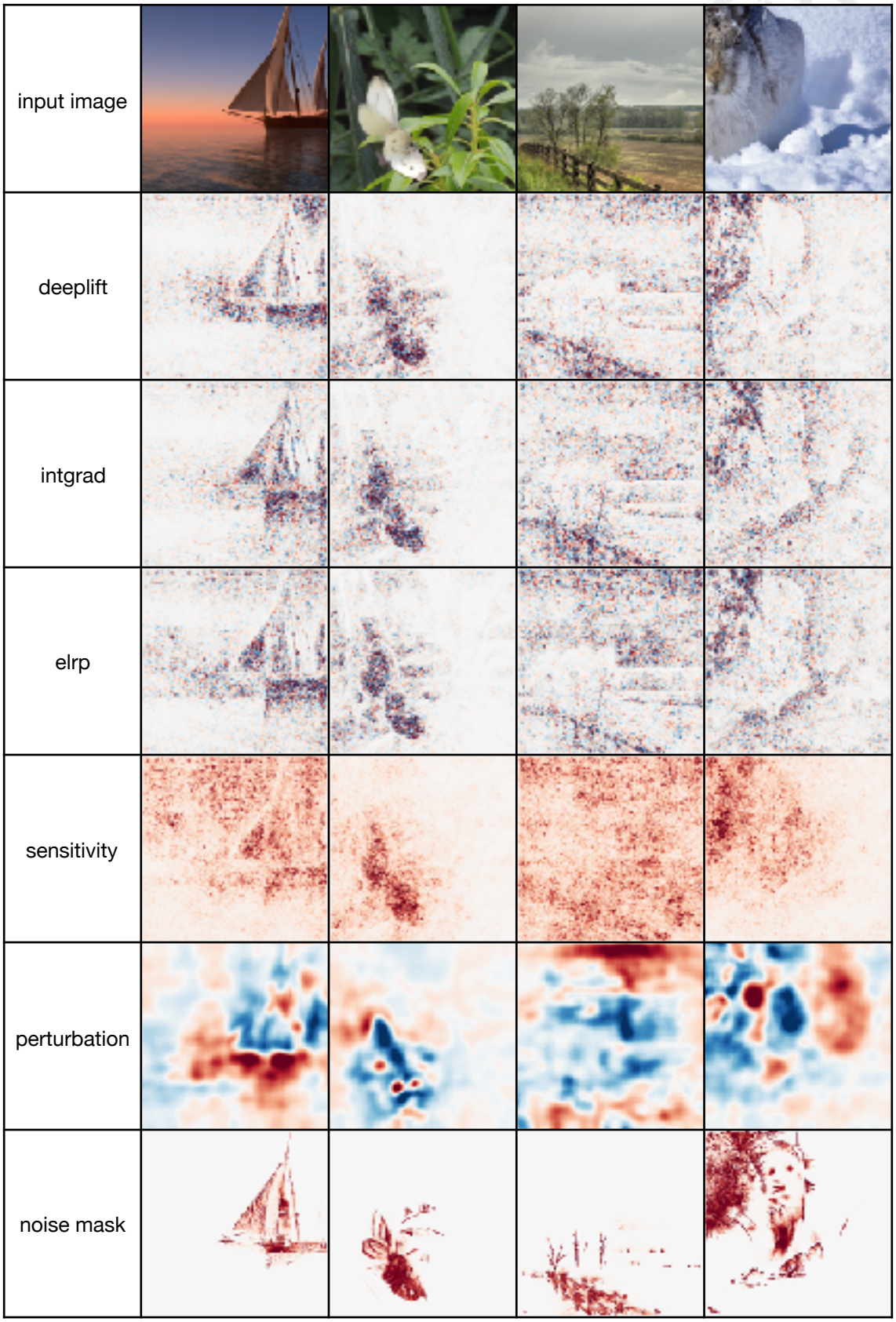


Figure 5: Saliency maps generated from different methods. Except the saliency of sensitivity and noise mask, negative contributions are represented by the blue color and positive contributions are represented by the red color.

doi:10.6342/NTU201801411

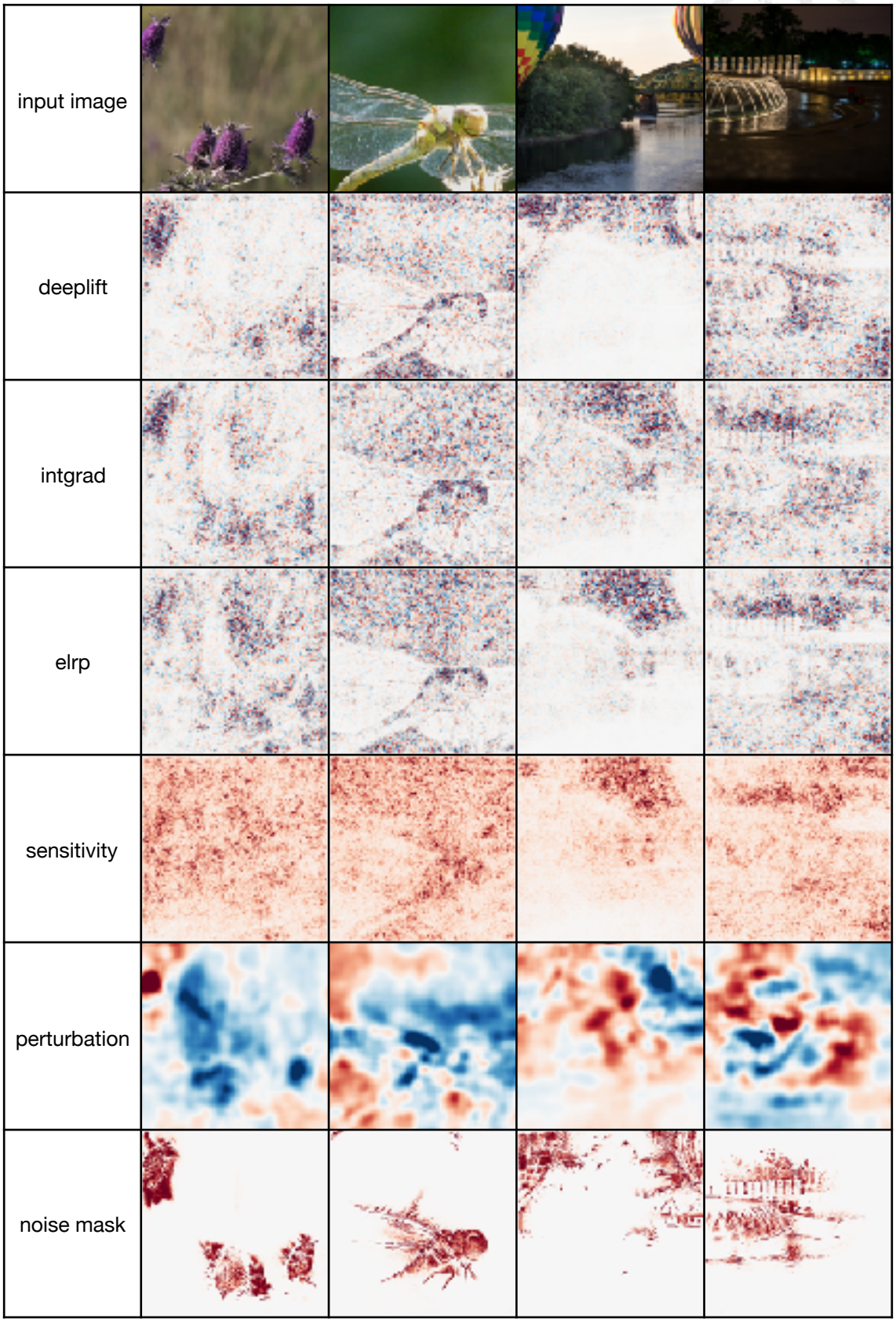


Figure 6: Saliency maps obtained from different methods. Except the saliency of sensitivity and noise mask, negative contributions are represented by the blue color and positive contributions are represented by the red color.

doi:10.6342/NTU201801411

We convert the saliency map and the input image into grayscale ones in order to measure the structural similarity between them. If a saliency map has higher structural similarity, it can describe the specific feature from the input image better, and therefore, it can save the effort of guessing and matching the structure to highlighted regions. To verify this claim, the average SSIMs of each method are calculated and shown in Table 2 . Clearly, from Table 2, Noise Mask gets the overwhelming score of 0.59, and others are below 0.1. That is, The proposed Noise Mask provides better visual quality in constructing saliency map than that of the other related works.

Table 2: Average SSIMs between the grayscale saliency maps and the grayscale input images over different methods.

Method	deeplift	intgrad	elrp	sensitivity	perturbation	noise mask
SSIM	0.039±0.05	0.022±0.04	0.030±0.05	0.007±0.01	0.039±0.039	**0.599±0.12**

4.2.3 Relevance Accuracy

The quality of a saliency map should also be evaluated by the machine itself. After all, humans' perspective cannot represent the view of machines. We evaluate the machine's relevance accuracy by the variation of [17]. As we mentioned in Chapter 2, MoRF sorts the input pixels in descending order according to the saliency value, replaces some pixels in each iteration, and passes the modified image to the network for recording the sore change. We replace top 100 pixels with random color values at each iteration instead of a patch. We also evaluate LeRF, which replaces pixels in ascending order. A better explanation method will cause the MoRF curve steeper at the beginning since it is more accurate on capturing the most important features. In the case of LeRF, a better explanation method won't make LeRF affect the prediction score until the end of

doi:10.6342/NTU201801411

the procedure. We ran 800 iterations over these two experiments, so around 90% of pixels are replaced at the end.

Figure 7 shows the average MoRF curves with respect to all methods listed in Table 1, we usually have a baseline that replaces the pixels in random order. The best methods are DeepLIFT and Integrated Gradients, whereas Noise Mask is as good as Sensitivity but getting worse after 500 iterations. This result is reasonable since Noise Mask is not a contribution analysis method but it can still capture the most influencing regions. However, we must evaluate them on LeRF to see whether the performance of each method is still the same.

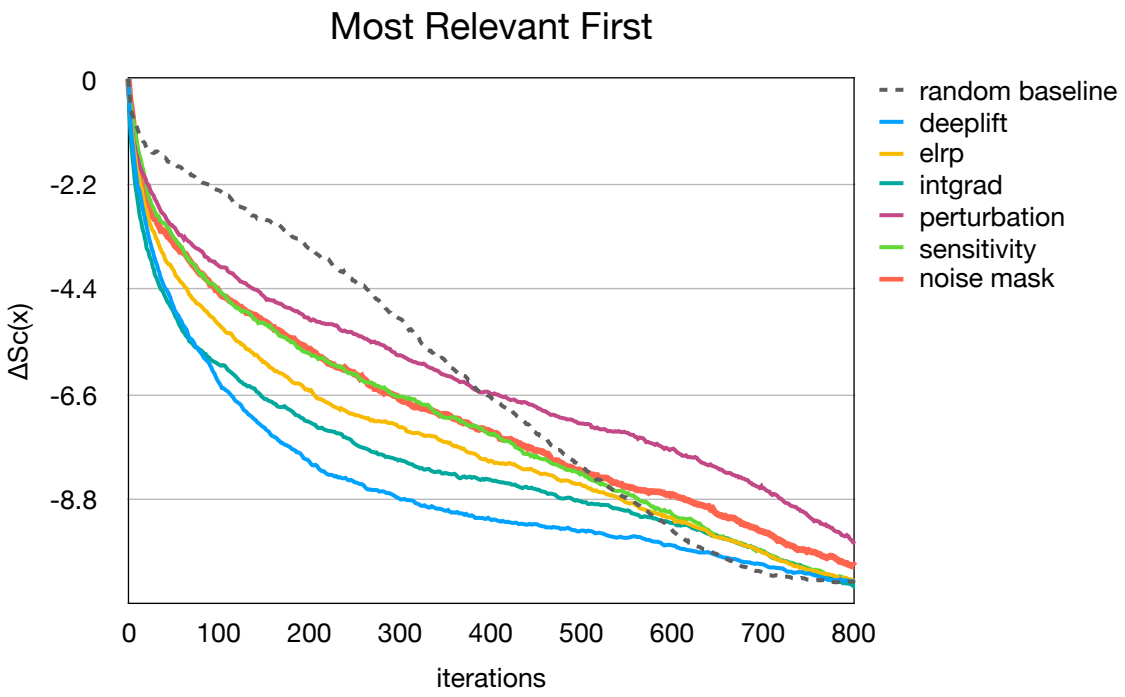


Figure 7: MoRF results w.r.t different methods.

doi:10.6342/NTU201801411

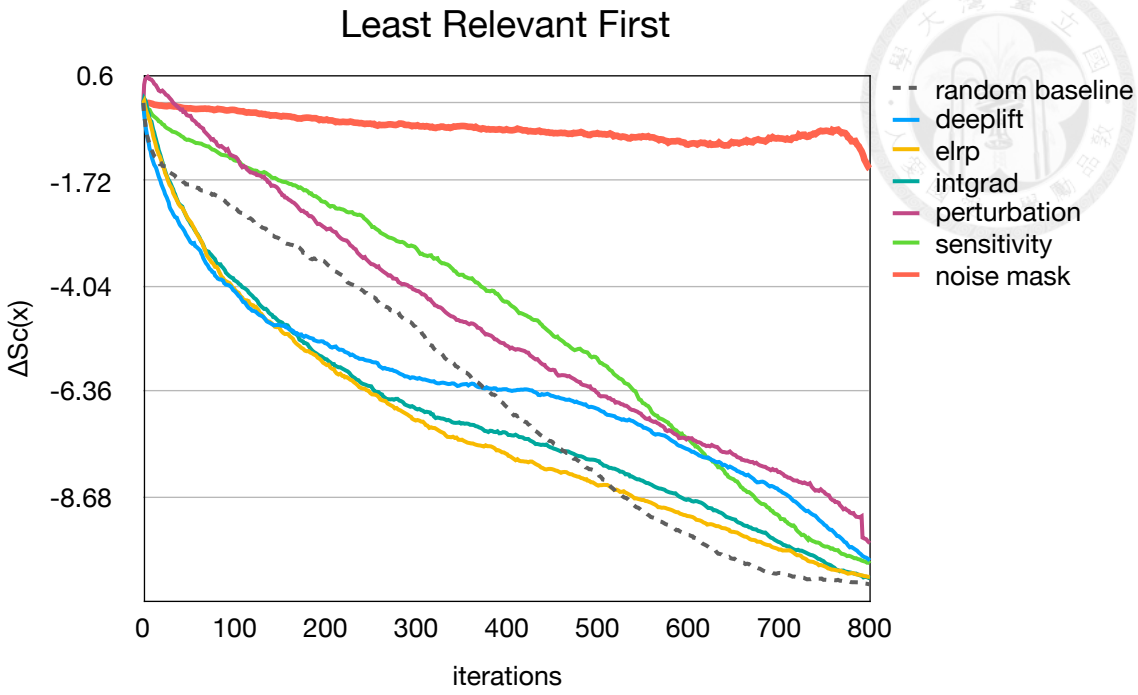


Figure 8: LeRF results w.r.t different methods.

Figure 8 shows the average LeRF curves. DeepLIFT and Integrated Gradients perform closely to the results of MoRF at the first 100 iterations. We doubt about the reliability of both methods since they might produce unrecognizable images in both LeRF and MoRF. On the contrary, Noise Mask performs reasonably, the curve only drops rapidly at the end. We claim that Noise Mask guarantees to preserve the subject and which is always recognizable.

In Table 3, we compute the area between MoRF and LeRF to get the summary called Area Between Perturbation Curve (ABPC). Noise Mask gets the largest area, generally, which implies that Noise Mask performs well in both testing phases and therefore is the most reliable one among all tested methods.

Table 3: ABPC of different methods.

Method	deeplift	intgrad	elrp	sensitivity	perturbation	noise mask
ABPC	1783.62	885.839	345.742	1926.34	1092.98	**5243.72**

doi:10.6342/NTU201801411

4.3 Model Discovery

In this sub-section we are trying to use Noise Mask to discover the recognition process of different CNN architectures. We tested 3 networks with different architectures, which are VGG 16 [22], Inception V1, and ResNet V1 50 [25]. All of them are relatively shallow but with good accuracy. Also, these models have comparable numbers of layers, so we can compare their layer's saliency maps at the resembling positions. We feed an image into the network and apply the Noise Mask to the intermediate layers. By observing the resultant saliency maps, we try to analyze the general behavior of these three models.

These models can be separated into modules on the basis of pooling layers. We examined only some layers in the last half of modules since we found that within the first half of modules, it is hard to find a balanced solution between the preservation term and the loss term, so we deduce that they are responsible for finding low-level features. The layers we chose to examine are the last layer of the third last module, the middle and the last layers of the second last module, the last two layers of the last module, and the fully connected layer before the softmax. We compare these 6 layers of 3 models at roughly the same positions of 3 models with respect to the two optimized terms which have been set to balance each other. Figures 9 to 11 show the testing results of 3 different images.

We also measured the values of the loss term at the optimal stage. Since we normalized the loss, we could compare the changes through layers. The observation is that the saliency points first spread over the image in the first two layers and then gathered together on the structural regions. Finally, it gradually concentrates on the

doi:10.6342/NTU201801411

recognized subject in the last modules. We also found that the losses gradually decrease at the last module. We believe this phenomenon also happens in the previous modules since the loss rises drastically when a small portion of the image has been masked. ResNet has more losses in the final few layers and keeps more background. This may explain the role of residual connections: they deliver information from the lower layers to the higher layers, so will enhance the classifying ability.

If we consider both the saliency map and the loss, we will find that the information about the image is gradually discarded but the losses on the layer's outputs are also decreased. This means the network progressively transfer the activation from the background to the subject. We conclude that networks in the same pooling procedure, in spite of different module architectures, have the similar recognition behavior.

doi:10.6342/NTU201801411

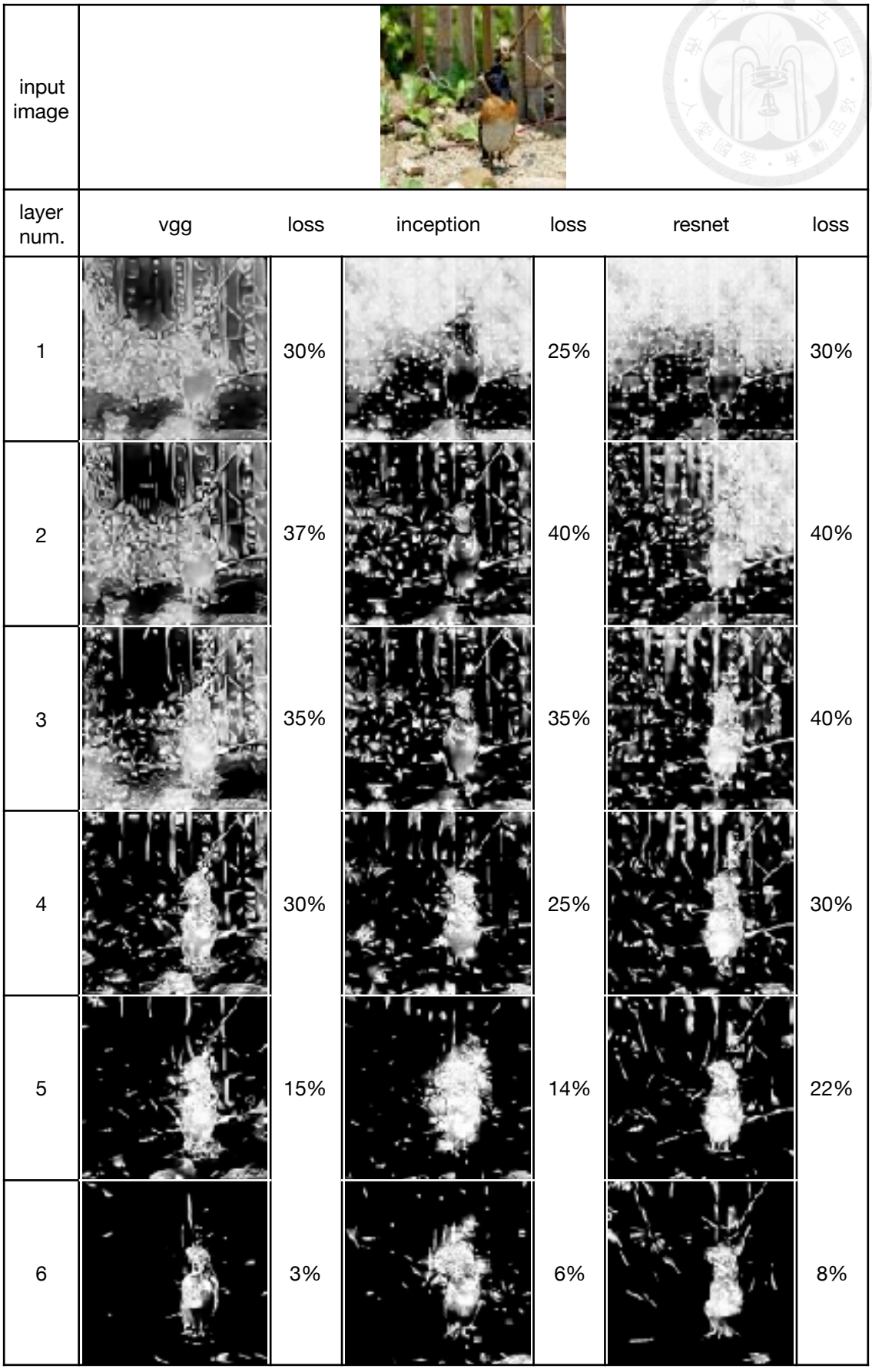


Figure 9: Saliency maps of a brambling image on different layers and models. The percentages report the values associated with the loss term.

doi:10.6342/NTU201801411

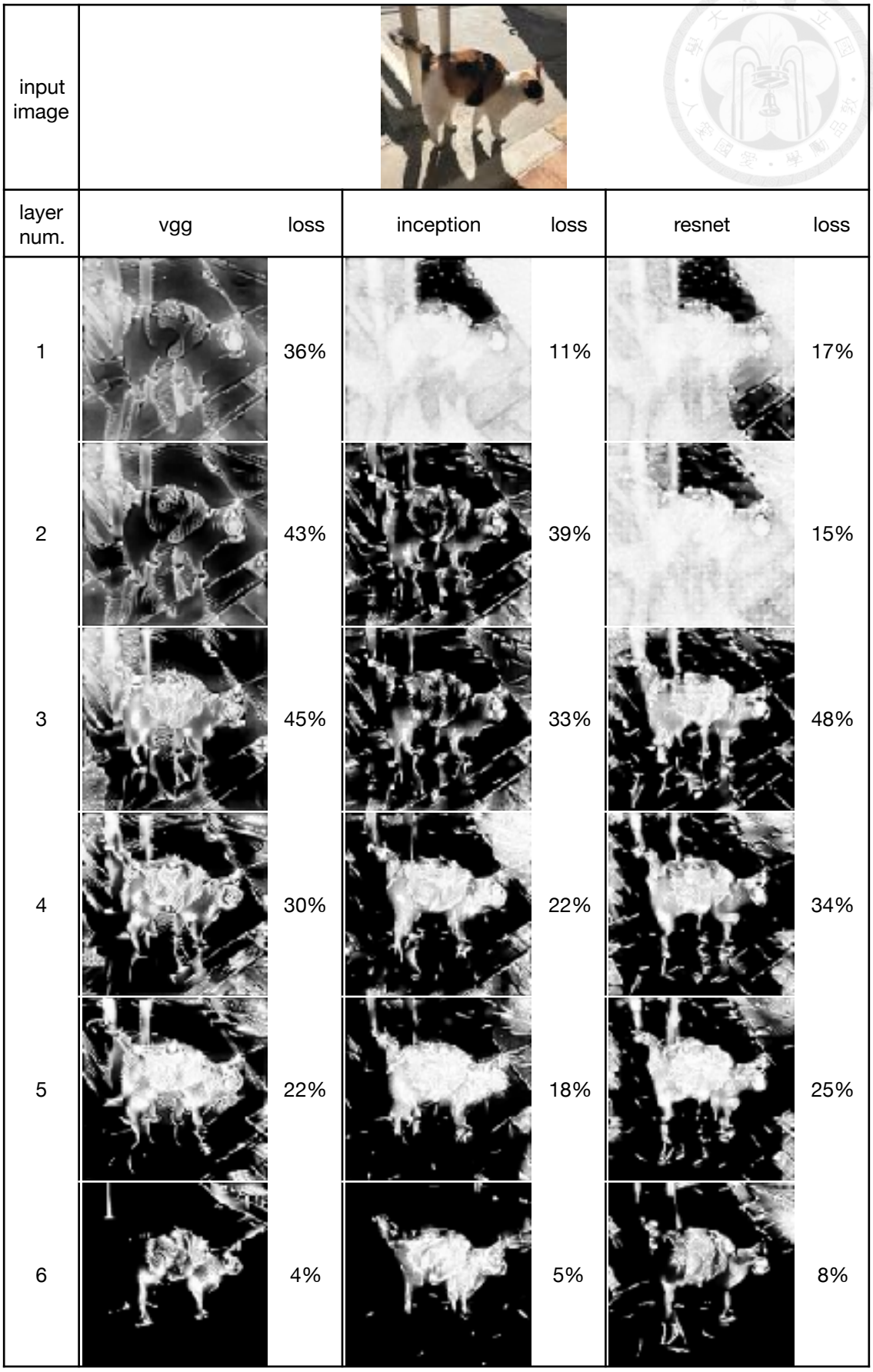


Figure 10: Saliency maps of a cat image on different layers and models. The percentages report the values associated with the loss term.

doi:10.6342/NTU201801411

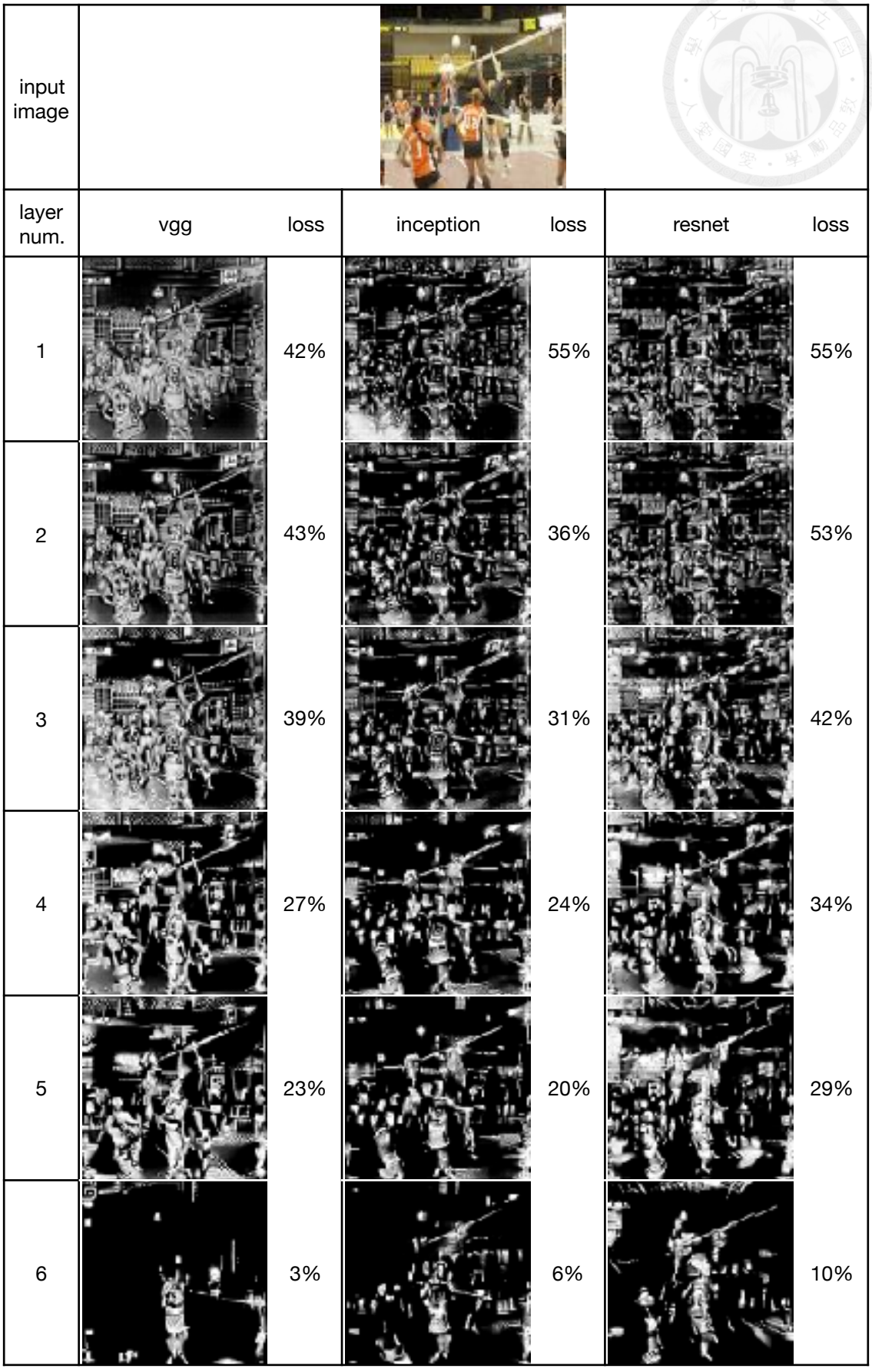


Figure 11: Saliency maps of a volleyball image on different layers and models. The percentages is the values associated with the loss term.

doi:10.6342/NTU201801411

Chapter 5 Conclusion and Future Works

We purposed the “Noise Mask”, an optimization-based approach for finding the saliency and/or most influencing regions of the input image and for explaining how the decision is made by a CNN. Noise Mask saves the unnecessary network modification and reliefs the need of choosing good reference input. Although we don't follow the contribution axiom, through LeRF tests, Noise Mask won’t mistake important features as irrelevances while other approaches might produce misleading explanations. In terms of visual quality, saliency maps constructed from Noise Mask are less noisy and have better structural consistency with the input images. Through the attempt to discover different CNN models, we found that they have similar recognition processes at similar layer positions while having their unique convolution modules. CNNs prepare structural features in most of the beginning layers and start to recognize the subject at the last few modules, which progressively transfers the activation from the input to the subject.

Further possible improvements include accelerating the optimization and the background sampling from a realistic background generator rather than the noise drawn from the uniform distribution.

doi:10.6342/NTU201801411

Reference

[1] G. Montavon, W. Samek, and K.-R. Müller. Methods for Interpreting and Understanding Deep Neural Networks. arXiv:1706.07979, 2017.

[2] Olah, et al., "Feature Visualization". Distill, 2017.

[3] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. A Unified View of Gradient-based attribution Methods for Deep Neural Networks. ICLR 2018, 2017.

[4] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding Neural Networks Through Deep Visualization. ICML Deep Learning Workshop 2015, 2015.

[5] A. Shrikumar, P. Greenside, and A. Kundaje. Learning Important Features Through Propagating Activation Differences. arXiv:1704.02685, 2017.

[6] S. Bach, A. Binder, G. Montavon, F. Klauschen, K. R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One, vol. 10, no. 7, pp. 1–46, 2015.

[7] M. Sundararajan, A. Taly, and Q. Yan. Gradients of Counterfactuals. arXiv: 1611.02639, 2016.

[8] L. Zintgraf, T. Cohen, T. Adel, and M. Welling. Visualizing Deep Neural Network Decisions: Prediction Difference Analysis. ICLR 2017, 2017.

[9] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. CVPR 2017, pp. 6541-6549, 2017.

[10] A. Nguyen, J. Yosinski, and J. Clune. Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks. arXiv:1602.03616, 2016.

doi:10.6342/NTU201801411

[11] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic Attribution for Deep Networks. arXiv:1703.01365, 2017.

[12] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. ECCV 2014, pp. 818-833 Springer, 2014.

[14] A. Mahendran and A. Vedaldi. Visualizing Deep Convolutional Neural Networks Using Natural Pre-images. arXiv:1512.02017, 2016.

[15] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. ICLR 2014 Workshop, 2014.

[16] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for Simplicity: The All Convolutional Net. ICLR 2015 Workshop, 2015.

[17] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. Evaluating the Visualization of What a Deep Neural Network Has Learned. IEEE transactions on neural networks and learning systems, 2016.

[18] Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T. and Clune, J. Synthesizing the Preferred Inputs for Neurons in Neural Networks Via Deep Generator Networks. Advances in Neural Information Processing Systems, pp. 3387–3395, 2016.

[19] Ziwei Liu, Raymond Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala

. Video Frame Synthesis using Deep Voxel Flow. Proceedings of International Conference on Computer Vision (ICCV) 2017, Oral Presentation, 2017.

[20] TensorFlow. https://www.tensorflow.org/.

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. CVPR 2015, pp. 1-9, 2015.

doi:10.6342/NTU201801411

[22] Kaggle Dataset. https://www.kaggle.com/c/nips-2017-non-targeted-adversarial-attack/data.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z.Wojna. Rethinking The Inception Architecture For Computer Vision. In Proceedings of CVPR 2016, pages 2818–2826, 2016.

[24] Karen Simonyan and Andrew Zisserman. Very deep Convolutional Networks For Large-scale Image Recognition. ICLR 2015, 2015.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. CVPR 2016, 2016.

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

[27] Tiny ImageNet. https://tiny-imagenet.herokuapp.com/.

doi:10.6342/NTU201801411

Appendix

A. Noise Drawn from Normal Distributions

We also tested the noise drawn from normal distributions which are estimated from a large natural image dataset [27] as the background, since the normal distribution maximizes the pixel entropy in the continuous domain. The pixel means of RGB channels respectively are 122.46, 114.26, 101.374, and the associated standard deviations are 70.63, 68.61, 71.93, respectively. Under these settings, we examined a few images in Figure 12 and compared saliency maps with those of the noise drawn from the uniform distribution. We noticed that there is no big difference between the saliency maps, but the average noise color tends to be yellow in the cases of normal distributions. Both types of distributions are acceptable, but we chose the uniform distributions for avoiding color bias.

doi:10.6342/NTU201801411

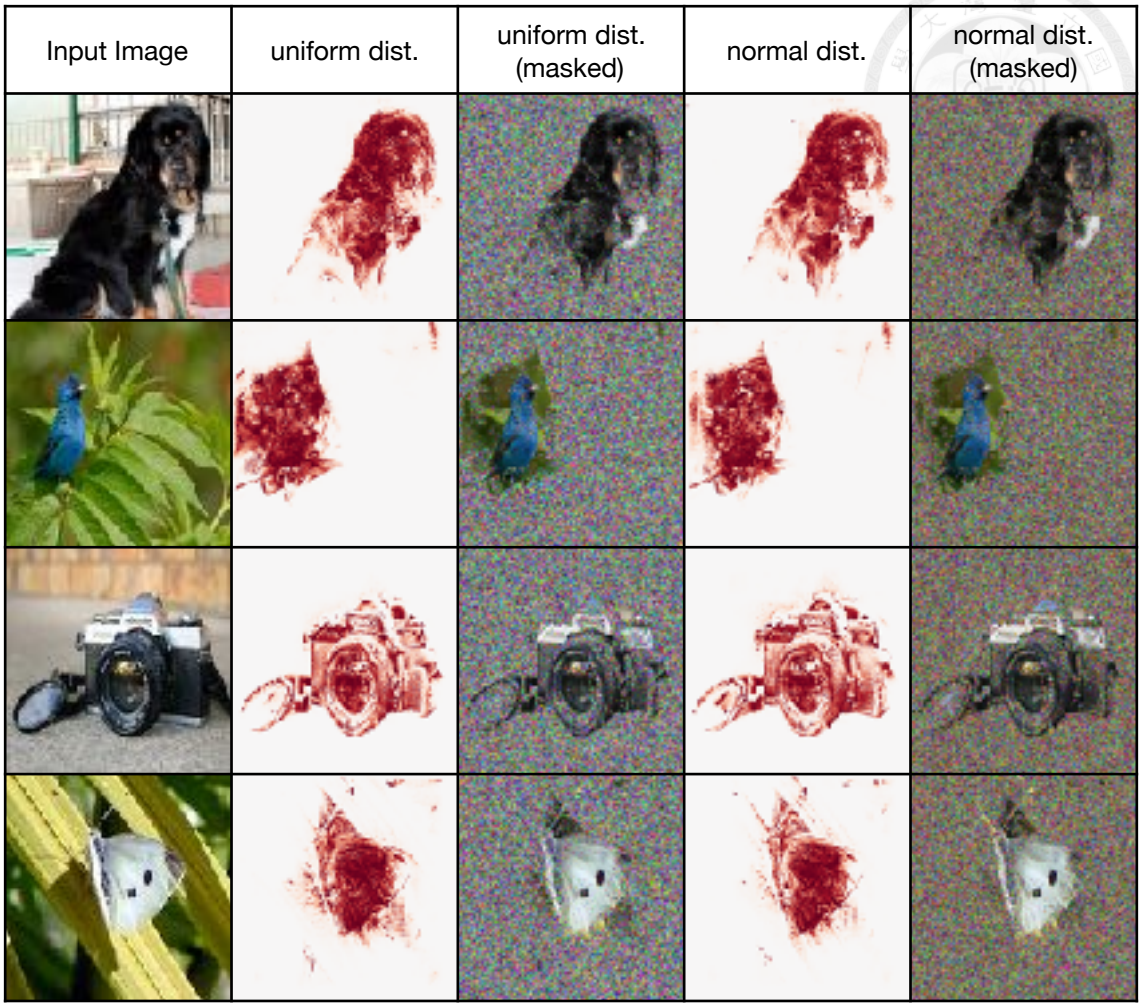


Figure 12: Comparing saliency maps of the background generated from the normal distributions and the uniform distribution. The second column and the fourth column show the saliency maps respectively obtained by the noise drawn from the uniform distribution and the normal distributions. The third column and the fifth column show the associated masked inputs.

doi:10.6342/NTU201801411