BEYOND PIXELS: A SAMPLE BASED METHOD FOR UNDERSTANDING THE DECISIONS OF NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interpretability in deep learning is one of the largest obstacles to more widespread adoption of deep learning in critical applications. A variety of methods have been introduced to understand and explain decisions made by large neural networks. A class of these methods are algorithms that attempt to highlight which input or feature subset was most influential to model predictions. We identify two key weaknesses in existing methods. First, most existing methods do not provide a formal measure of which features are important on their own, and which are important due to correlations with others. Second, many of these methods are only applied to the most granular component of input features (e.g., pixels). We partially tackle these problems by proposing a novel Morris Screening based sensitivity analysis method using input-partitioning (MoSIP). MoSIP allows us to quantify local and global importance of less granular aspects of input space, and helps highlight which parts of inputs are individually important and which are potentially important due to correlations. Through experiments on both MNIST with spurious correlations (Biased-MNIST), and the large scale ImageNet-1K dataset, we reveal several new and interesting findings. *Our key finding is that newer CNN architectures (e.g., ResNet) compared to older architectures (e.g., VGG) do not extract fundamentally more relevant features, but simply make stronger use of non-linearities and feature interactions.* This can manifest itself in the use of spurious correlations in the data to make decisions.

1 INTRODUCTION

Deep learning models are becoming endemic in various applications. As models are increasingly used for critical application such as detecting lung nodules (Schultheiss et al., 2021) or autonomous driving (Li et al., 2021), it is important to to either create interpretable models, or to make opaque models human interpretable. This paper focuses on the latter.

Over the past decade, much progress has been made on interpreting deep models. These methods can be broken down into model agnostic vs model dependent, and/or local vs global. Model agnostic methods, such as Shapley values (Kononenko et al., 2013), weigh the importance of input features without relying on the structure of the model. In contrast, methods such as GradCam (Selvaraju et al., 2017) and GradCam++ (Chattopadhay et al., 2018) are heavily dependent on model architecture. Local methods, such as the Integrated Gradients method proposed by Sundararajan et al. (2017) and Taylor Decomposition proposed by Montavona et al. (2018), focus on understanding the importance of features for specific inputs. Global methods, such as breakDown presented by Staniak & Biecek (2018), attempt to explain features across a collection of inputs.

While these methods yield valuable information about models, they share two common gaps. The first is that none of them report metrics that distinguish between the *features in input space that are individually important, and features that are important because of their interaction with other features.* The second is that the above methods are generally applied to inputs at their most granular level. The combination of these two gaps limits the conclusions that Machine Learning practitioners can make about individual model predictions, as well as limits the ability of these methods to analyze the evolution of models.

There exist methods in the statistics/ML literature that explicitly measure feature interaction, but have not been applied to large deep learning problems. Two such methods are Sobol indices (Sudret, 2008), and Shapley interaction values (Agarwal et al., 2019). These methods are costly, requiring the analysis of pairwise inputs in the method.

The Morris method (Morris, 1991) is a model agnostic method that compromises between measuring feature interaction and cost. It measures which features are individually important and which are important due to *feature interactions and/or non-linearities*, and can be used both as a global and local method, much like Shapely (Linardatos et al., 2021). We present, what is to the authors' knowledge, first application of this method to Deep Learning models.

The number of model evaluations required for Morris scales linearly with input dimension. For deep learning applications involving images and text, this is prohibitive. This computational burden can be greatly reduced if, rather than focusing on interpretability at the most granular level of inputs, such as pixels, we perform sensitivity analysis at more *semantic* levels. Combining sensitivity analysis of models on semantic levels of inputs with the model agnostic methods like Morris, opens the door to a formal method for answering a host of questions about how models use inputs. We demonstrate this with two applications

First, we perform a global analysis on a MNIST dataset that is biased in digit color, background color, and digit position. The semantic partitions here correspond to this metadata. Second, we performed local and global analysis on different architectures applied to a subset of ImageNet. These architectures were AlexNet (Krizhevsky et al., 2012), VGG-16 (Simonyan & Zisserman, 2014), Inception-V3 (Szegedy et al., 2016), and ResNet-50 (He et al., 2016). The semantic partitions here are regions of the images. Our experiments demonstrate that all of the CNN models relied more heavily on non-linearities/interactions than individual features globally. Moreover, we quantitatively demonstrate that the main impact of the evolution of architectures from AlexNet to ResNet-50 was to make more use of feature interactions/non-linearities. Although the Morris Method cannot distinguish between interactions and non-linearity, we use a heuristic to test top-K most important regions for the existence of interactions.

In summary, we make the following contributions:

1. We present, to our knowledge, the first application of the Morris Method to deep learning models.
2. We propose a novel application of the Morris Method that performs sensitivity analysis at the level of semantic partitions of the input, Morris Sensitivity-analysis on Input Partitions (MoSIP).
3. We demonstrate the ability of MoSIP to correctly reveal the most important semantic components of inputs.
4. We show that newer CNN models do not learn fundamentally new features. They primarily increase their exploitation of semantic feature interactions/non-linearities.

2 Methodology (MoSIP)

In this section we describe how MoSIP was used to analyze the importance of semantic representations of inputs. While the methodology is general, specifics will vary with application. This variation will be made clear during our experiments. MoSIP consists of two building blocks. The first is Morris screening which allows us to identify features that are most important to a model's prediction, both individually and as part of interactions/non-linearities. The second is input partitioning, which divides input into semantically coherent chunks.

2.1 Morris Screening

The Morris method is a sensitivity analysis method that is typically used as an initial screening method to reduce the number of parameters to be evaluated using more expensive methods such as the Sobol method (Ge & Menendez, 2017).

Given a model, $f(\boldsymbol{x})$, that acts on an input, $\boldsymbol{x} \in \mathbb{R}^n$, the Morris method samples N values $\boldsymbol{x}$. It then creates designs from each of these sampled vectors. The goal of these designs is to construct a series of points that differ in only one component of the input $\boldsymbol{x}$, x_i. The two most common type of designs are trajectory and radial. The details of these designs are described in Appendix A.1.

The generated designs are subsequently used to calculate elementary effects. Elementary effects are a measure of the contribution of each feature component to the model output. An elementary effect is calculated for each component of the input, i, and for each sampled vector, $\boldsymbol{x}_r$.

$$EE_{i,r} = \frac{f(x_{1,r}, ..., x_{i-1,r}, x_{i,r} + \Delta, x_{i+1,r}, ...x_{n,r}) - f(\boldsymbol{x})}{\Delta_i} \tag{1}$$

where $\Delta = \frac{m}{2(m-1)}$ and m is the number of levels used when sampling for the trajectory design (Sobol, 1976).

The mean and standard deviation of these elementary effects are used as measures of importance for each individual and non-linear/interactive importance of input components respectively.

$$\mu_i = \frac{1}{N}\sum_{r=1}^{N} EE_{i,r},\ \sigma_i = \sqrt{\frac{1}{N-1}\sum_{r=1}^{N}(EE_{i,r} - \mu_i)^2} \tag{2}$$

While μ is effective for models that aren't heavily non-linear/interactive, Campolongo et al. (2007) found that it often fails for non-linear models. To combat this, they introduce μ^* which is the mean of the absolute value of elementary effects.

$$\mu_i = \frac{1}{N}\sum_{r=1}^{N} EE_{i,r} \quad \rightarrow \quad \mu_i^* = \frac{1}{N}\sum_{r=1}^{N} |EE_{i,r}| \tag{3}$$

μ_i^* states how significant the input component, i, is to a model prediction and σ_i states the extent to which the model is non-linear in i, or makes use of interactions between i and other features. There are three possible scenarios with μ_i^* and σ_i.

1. μ_i^* is low. The feature is not relevant.
2. μ_i^* is high and σ is low. The feature is important, but only linearly.
3. μ_i^* is high and σ is high. The feature important and has significant interactive and/or non-linear effects.

In MoSIP each component being analyzed, i, is a real valued number that represents a semantic input partition. We detail this more in the following section.

2.2 Input Partitioning

Input partitioning is key to our methodology. It involves extracting semantic features from raw inputs and representing them as real numbers. These real numbers are used as the input, $\boldsymbol{x}$, into MoSIP and are sampled for Morris screening. For Morris screening to provide trustworthy sensitivity analysis for these semantic features, it is critical to provide an accurate method for mapping from these semantic features back to the input space used for model predictions. In the next section, we demonstrate useful input partitioning for two datasets. However, we note that the general principles can be applied to other data modalities such as language, audio and video.

2.3 Algorithm

We present a diagram of our methodology in Figure 1. The **Input partitioner** is responsible for generating semantically partitioned components of the input and a logic to map the partitions back to raw input space. The Input partitioner is used by the **Model Wrapper** to transform samples of the partitioned objects back to raw space, before feeding these samples to the runnable model. The output of the runnable model is provided to the **Morris Screening Method** which computes sensitivity results for the output with respect the input partitions. The results are visualized and validated.

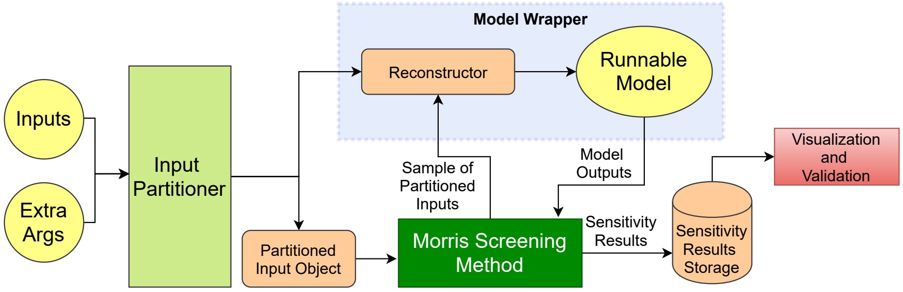
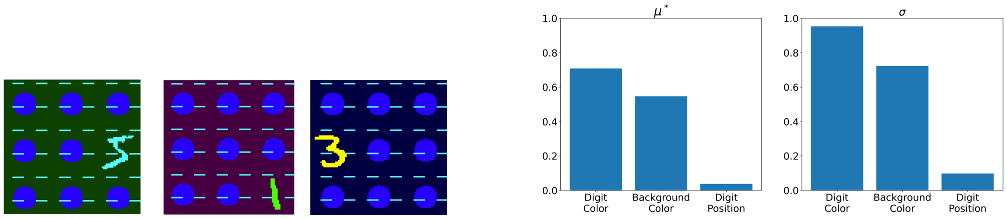


Figure 1: Schematic of MoSIP building blocks.



(a) Example of a Biased-MNIST dataset. The displayed digit colors, background colors, and positions are indicative of the manner in which the data is biased for each displayed digit.

(b) Normalized standard morris method results for a Biased-MNIST with a correlation level of 0.7. Digit color and background color are both significant, with non-linear/interactive effects, while digit position is not.

Figure 2: Biased-MNIST and relative significance of different confounding variable according to MoSIP.

3 Data, Models, and Partitioning

In this section we discuss the data and models we will apply the Morris method to, as well as the how we partition this data.

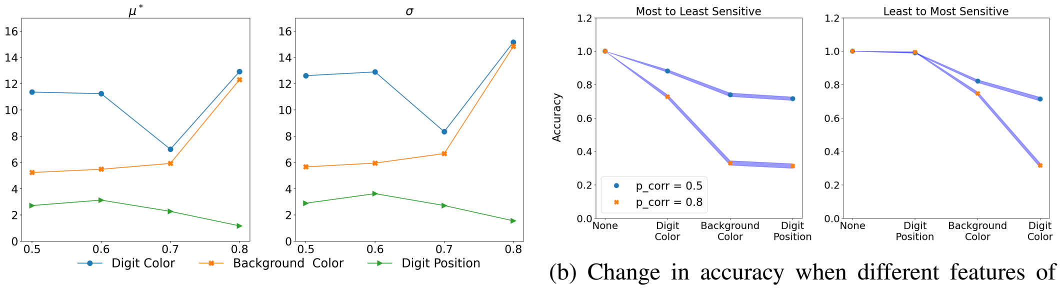
3.1 Biased-MNIST

Standard MNIST is a test set for image classification algorithms consisting of handwritten digits from 0-9. Recent work has created variations of MNIST that are biased in order to study CNN's tendencies to use spurious correlations. One such variation was an MNIST dataset which created a correlation between digits, background color, digit color, distractor shapes, background texture, and digit position (Shrestha et al., 2021). We test the ability of the Morris method to detect CNN model's use of these correlated features. Using code from Shrestha et al. (2021), we create Biased-MNIST that has spurious correlations only in digit position, digit color, and background color. Figure 2a shows three samples of this dataset. The displayed digit position, digit color, and background color are set to the values that are correlated with the digit class.

Biased-MNIST was partitioned as follows. The digit color and background color had 10 possible values, while the digit position had 9 possible values. An image generator function takes these three integers and produces corresponding Biased-MNIST images. With reference to Figure 1, these integers serve as the semantic partitions of the input samples, and the image generator serves as extra arguments capable of generating raw input space from the partitioned values. More concretely, the partition input is $\boldsymbol{x} \in \mathbb{R}^3$, and x_i is an integer representing digit color, background color, or digit position. The CNN model used for classification is presented in Appendix B.

3.2 ImageNet Data Partitioning and Models

ImageNet is the most common dataset for benchmarking modern deep learning classification models. ImageNet, and the CNNs benchmarked on it, provide an ideal test case for MoSIP. We apply



(a) Unnormalized standard morris method results for a Biased-MNIST at varying correlation levels.

(b) Change in accuracy when different features of Biased-MNIST are changed sequentially from most-to-least important (left) and vice-versa (right)

Figure 3: Trend in MoSIP sensitivity as a function of bias correlation and validation of MoSIP on Biased-MNIST metadata

MoSIP to ImageNet-1K (Russakovsky et al., 2015) and a subset of the CNN architectures. AlexNet, VGG-16, Inception-v3, and ResNet-50. In particular, we use MoSIP to explore how these CNNs evolved over time (Alom et al., 2018).

The partition used for performing sensitivity analysis on these images is image regions. For the purpose of testing, we use a simple partitioning scheme. Images are split into $w \times w$ grids of regions. The pixel values of these regions are aggregated, through summation, into a vector of lumped values, $\boldsymbol{l} \in \mathbb{R}^{w^2}$. A map between these lumped values and the original image is created by dividing the region images by their lumped values. Multiplying $\boldsymbol{l}$ by their corresponding regions in the map would yield the original image. The lumped value vector, $\boldsymbol{l}$, is sampled by the Morris Method, and multiplying these lumped vectors by the map is used to project these samples back to the image space. We use PyTorch (Paszke et al., 2019) pre-trained models for our sensitivity analysis experiments.

4 Biased-MNIST Experiments

In this section we present Biased-MNIST experiments. The parameters of the Morris method are $N = 100$, $n_{partition,x} \times n_{partition,y} = 3 \times 1$, and $m = 20$. N is the number of samples generated for each input, $n_{partition,x} \times n_{partition,y}$ is the dimension of the partition inputs, and m is the number of levels used for trajectory design sampling.

We train the CNN model, presented in Appendix B, on Biased-MNIST with varying levels of correlations, p_{corr}, in digit color, background color, and digit position. These levels were 0.5, 0.6, 0.7, and 0.8. We then use the models to evaluate the feature importance of test sets that correspond to these levels.

Figure 2b shows the results for the $p_{corr} = 0.7$ correlation data. These results are normalized per output, and averaged over all the outputs. Our analysis shows that digit position is unimportant. It further shows that the Morris method accurately captures the importance of digit and background colors to the model prediction. Figure 3a displays similar results as a function of correlation. The importance of background color reduces monotonically with correlation level, while digit color does not. In particular, at $p_{corr} = 0.5$, the σ of digit color is twice as large as for background color.

Although the Morris method cannot distinguish between feature interactions and non-linear effects out-of-the-box, we can use the results of this analysis to make a distinction through validation. Validating this result essentially consists of choosing the subset of test images for which our model correctly classified the digit, randomly changing the digit color, background color, and digit position levels sequentially, and observing the drop in accuracy. If σ is primarily a measure of non-linearity and not correlations, we would expect the resulting drop in accuracy to be proportional to differences in σ. For example, the σ of digit color is twice as large as for background color at $p_{corr} = 0.5$. If they are not correlated, the drop in accuracy for changing digit color will be twice as much as changing background color regardless of the order in which the changes are made. To confirm, we run the experiment for changing most-to-least important features and vice versa 100 times. The results along with the 95% confidence are reported in figure 3b. We see that changing the digit color or background color have similar impacts on accuracy at both correlation levels rather than expected $\approx 2\times$ impact for digit color if the σ values were primarily based on non-linearity alone. Hence, we can infer from this that the σ values of digit colors and background colors are primarily measures of feature interaction, rather than non-linearity alone.

These results verify the Morris method's ability to accurately capture when a model use's correlations in non-geometric semantic components of inputs.

5 ImageNet Experiments: Quantitative

In this section we present quantitative ImageNet sensitivity analysis experiments. For all of these experiments, we use a set of 80 images constructed by randomly sampling 20 validation set images from the randomly chosen classes of ”oscilloscope”, ”brambling”, ”grey fox”, and ”mobile home” classes. We explore two main questions. One, do more modern CNN architectures use fundamentally different features to make predictions. Two, how have CNN's use of non-linearities/feature interactions evolved over time.

To answer these questions, we use both local and global sensitivity analysis. The difference between global and local sensitivity analysis for the Morris method lays primarily in how the lumped partitions are sampled. For the local analysis lumped partitions are uniformly sampled for each input, with the lumped values of the inputs as the mean of this uniform distribution. For the global analysis, lumped partitions are sampled by using the maximum and minimum values of the lumped components of *all* inputs as bounds. The samples generated are used with each input map, and the sensitivity results are averaged.

The parameters used for the Morris method are $N = 40$, $n_{partition,x} \times n_{partition,y} = 8 \times 8$, amd $m = 8$. These quantities are defined as described in section 4.

Before exploring the evolution of ImageNet CNNs, we validate that the Morris method accurately selects the most, and least, relevant regions of images.

Validating the Morris Method: We quantitatively validate MoSIP as follows. We select the subset of inputs that each model accurately predicted from our sample. For these subsets, we sequentially masked the top 20% most relevant regions. Separately, we sequentially mask the 20% least important image regions. We report the average accuracy of this subset, as well as the average change in score of the ground truth outputs as a function of region masking. If the Morris method is accurately selecting regions of importance/non-importance, we expect that the score and accuracy should decrease by more than 20% when we mask the 20% most important regions. We additionally expect that masking the least sensitive regions will not significantly change the accuracy and or score.

Figure 4 shows the average change in accuracy when masking regions that are rank based on σ, μ^*, and $\mu^* + \sigma$ respectively. We see that, for all the models, the decrease in accuracy after masking the most important regions was at least 50%. For score the results were similar, figure 9 in the appendix. Masking the least important regions caused almost no change in score or accuracy. These results indicate the the Morris method is accurately identifying the most and least important regions in the image. Figure 10, in Appendix C, shows the results of this validation for global analysis. For global analysis, the same regions are masked in all inputs. While not as pronounced as the local analysis, the accuracy drops by more than 20% for all models. Similar to the local case, masking the least important regions causes very little change. This indicates two things. First, the Morris method is able to accurately pick regions of most/least importance globally. Second, since globally relevant regions exist, ImageNet likely has a bias towards relevant features in the data being located in certain regions of images.

Evolution of CNN Architecure's on ImageNet: With MoSIP validated, we explore the evolution of our architectures. In particular, we ask whether or not CNN model predictions on ImageNet are based on the use of fundamentally different features in the input space. We explore this question using both local and global analysis.

We use the local analysis to quantify the extent to which modern models use fundamentally different features. For the local analysis, we measure the portion of the top 20% most relevant regions in AlexNet that overlap with the top 20% most sensitive regions in VGG-16, Inception-v3, and

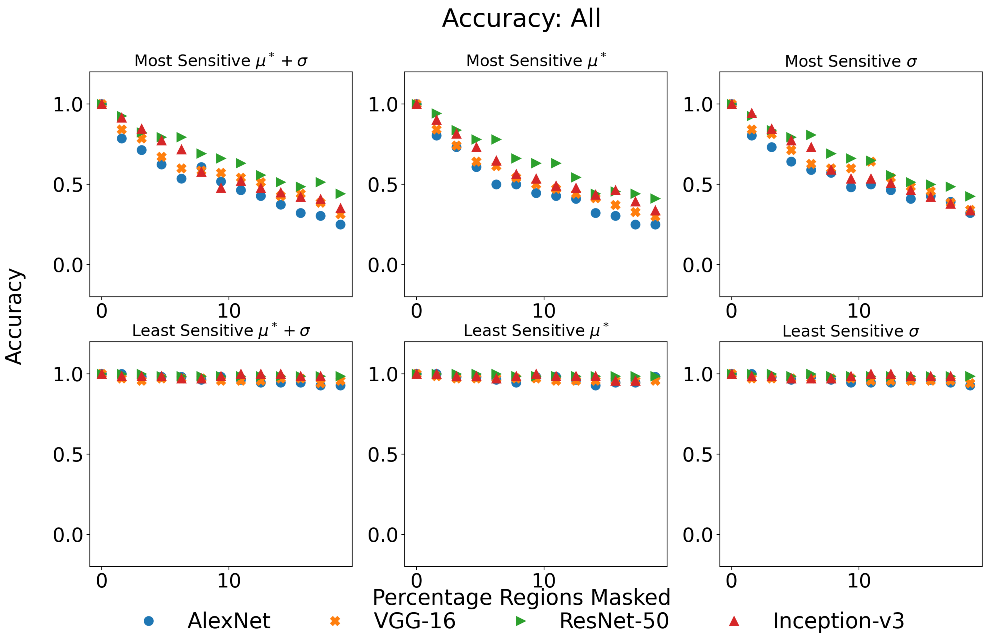
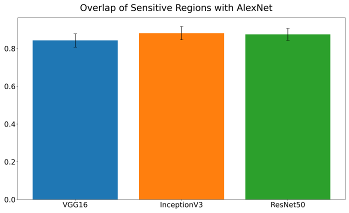
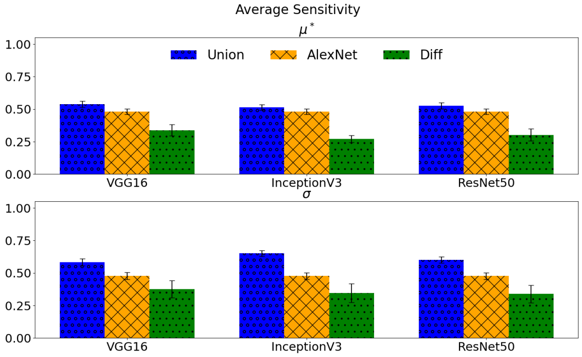


Figure 4: Average change in the accuracy of the models when certain percentage of regions are masked based on μ^*, σ, and $\mu^* + \sigma$. (TOP) Most important . (BOTTOM) Least important.



(a) Portion of top-20% of regions in each model that is also in the top-20% of regions of AlexNet.

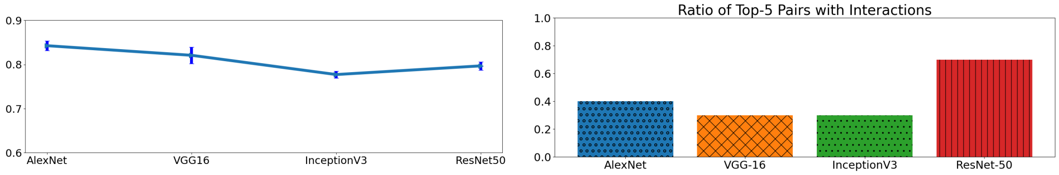


(b) Average sensitivity modern models and AlexNet for overlapping regions, and for differing regions.

Figure 5: Results from local experiments testing the difference in important regions between AlexNet and more modern CNNs.

ResNet-50. Regions are defined as overlapping if they are identical between AlexNet and the other models, or if they are immediate neighbors. We additionally calculate the average μ^* and σ of the regions of the VGG-16, Inception-v3, or ResNet-50 that overlap with AlexNet. We also calculate the average of these quanitities for regions that differ. For reference, the average sensitivity values for the overlapping regions are computed for AlexNet. The averaging is done over inputs, and the 95% confidence is reported.

Figure 5 shows the results of these experiments. For μ^*, σ, and $\mu^* + \sigma$ all of the models' most important regions overlap with at least 80% of the most important AlexNet regions. The models additionally have higher μ^* and σ for the overlapping regions, than the differing regions. This implies that the most important regions to decision making for modern CNNs overlap strongly with AlexNet. A key finding here is that *modern CNNs have higher values of σ for regions that overlap with AlexNet, than AlexNet has for those same regions.*



(a) (Average $\frac{\mu^*}{\sigma}$ calculated for regions that satisfy $\mu^* > 0.6$ or $\sigma > 0.6$.

(b) Ratio of all pairwise combinations of regions in the top-5 MoSIP results that show evidence of interactions

Figure 6: Evolution of CNN's over time with respect to use of correlated features

These results imply that the difference between AlexNet and modern CNNs is not the extraction of fundamentally different semantic features, but more pronounced use of non-linearities and interactions in these features. The stronger use of interactions can lead to stronger exploitation of correlations in the data when making decision.

We further explore whether or not modern CNN models make stronger use of interactions/non-linearties than older CNNs using global analysis. To measure the relative importance between individual and interactive/non-linear features, we calculate the ratio between the μ^* and σ, $\frac{\mu^*}{\sigma}$, for relevant regions in the image. This quantifies the extent to which each model weighs individual vs interactive/non-linear feature importance for all relevant regions.

In this analysis, relevant regions are defined as regions for which the normalized μ^* or σ are greater than 0.6. Figure 6a displays the results of these calculations. Figure 6a shows that $\frac{\mu^*}{\sigma}$ decreases overtime. This reinforces the local analysis result that the key effects of the evolution of CNNs was to weigh feature interaction/non-linearity more heavily for relevant regions, not to discover fundamentally new features.

Do newer CNNs exploit more correlations?: The final question to explore here is to what extent more modern CNNs use interactions vs non-linearities reletive to AlexNet. While MoSIP cannot distinguish beteen non-linearities and interactions out-of-the-box, we propose a top-K heuristic that can. We select all possible pairs of regions in the top-K regions from MoSIP. For each resulting pair, x, y, we mask x, yielding a change in accuracy, δ_1, and then mask y, yielding a change in accuracy of δ_2. We then perform this process in reverse order, yielding γ_1 and γ_2. If there are pairwise interactive effects, then masking x and y individually should yield a greater change in accuracy than when y is masked after x and vice-versa. Mathematically, an interaction is present in a pair, x, y, exists when $\delta_1 + \gamma_1 > \delta_2 + \gamma_2$. For each model we perform a top-5 version of this heuristic, and report the ratio of the resulting combinations that have interactive effects. A high ratio denotes the presence of more pairwise interaction effect rather than simple non-linearity.

Figure 6b shows our results. We see that, relative to AlexNet, VGG-16 and Inception-v3 appear to use interactive effects less. This makes it likely that the increase in σ of those models relative to AlexNet, figure 5, is primarily a result of a more non-linear use of image regions with respect to model outputs.

ResNet-50 uses interactive effects much more than other models. This implies that the increase in σ seen relative to AlexNet is primarily a result of an increase in feature interaction. While the use of feature interaction isn't necessarily negative, it makes ResNet-50 more vulnerable to spurious correlations in the data.

6 ImageNet Experiments: Qualitative

While many types of questions can be answered quantitatively, it is at times necessary to use interpretability methods on particular examples. We show the results of the Morris method on a mobile home and brambling example in figure 7. Both yield some evidence that CNN models make use of data correlations when performing classification. AlexNet, VGG-16, and Inception-v3 all correctly classify the mobile home image. However, ResNet-50 classified the image as a stretcher. The most important part of the image for this decision by ResNet-50 was the presence of firemen. Although not explicitly explored here, it is reasonable to assume that images that contain stretchers also usually contain emergency personnel such as fireman. In the brambling case, all of the models classified the image incorrectly. The most prominent bird in the image is a brambling. AlexNet, VGG-16 and ResNet-50 classified the image as a gold finch. The most critical regions for this decision making process for the models are the yellow portions of the birds in the background, and the bird feeder. In both of these examples, σ highlights the parts of the images that are important due to correlations.

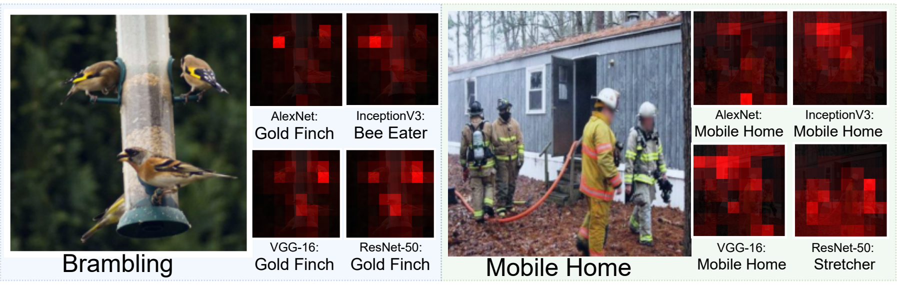


Figure 7: Local Morris sensitivity results when AlexNet, VGG-16, Inception-v3, and ResNet-50 are applied to two images, brambling (left) and mobile home (right).

The results of Biased-MNIST and these local experiments show that σ, which represents both feature interactions and non-linearities, is a reasonable initial measure of the extent to which models use correlations when making decisions.

Consistency with prior results: The findings in our experimental section are consistent with work done by others studying the behavior of CNNs for classification. In particular, our results are consistent Brendel & Bethge (2019) who found that early CNN models were well approximated using Bag of Words patches, but not later ones. This indicated that later CNNs used inputs more non-linearly (interactive). Our results are also consistent with Xiao et al. (2020) who found that image classifiers were susceptible to adversarial attacks involving modifying image backgrounds, an indicator that these models exploit spurious correlations

7 Discussion, Conclusion, Further Work

Through MoSIP, we have developed a tool that facilitates understanding the decisions of deep Learning models at a semantic level. MoSIP allows us to quantify the extent to which semantic representations of the input impact model decisions. This semantic level analysis facilitates the discovery of model bias, as well as the ability to answer interesting high level questions about the behavior of these models. Our experiments with Biased-MNIST demonstrated MoSIP's ability to detect CNN's use of spurious correlations. Moreover, our ImageNet experiments demonstrate MoSIP's ability to facilitate a deeper understanding of deep models. We found that the changes in the CNN architectures since AlexNet manifests themselves primarily through greater interactive and non-linear exploitation of image regions, not the use of fundamentally different regions.

While our studies were limited to image models, MoSIP can be easily extended to other domains, such as text and audio as well. Furthermore, the general principles of MoSIP can be easily extended sampling based sensitivity methods other than Morris. Additionally, although our naive input partitioning strategy yielded interesting results, expert-designed input partitioning could be used to address domain-specific questions (e.g in medicine, autonomous driving, etc...). Finally, our work could also be extended to study the interaction and importance of different layers of deep models in model predictions, rather than studying input-output relationship.

We hope that MoSIP assists researchers in interpreting otherwise opaque deep learning models, facilitates the development of future models based on a strong understanding of how current models use semantic information, and enables the construction of diverse datasets based on an understanding of which semantic features in the dataset are relevant.

Reproducibility

At the time of writing this, the authors are obtaining permission from their organization to open-source the code used for this paper. When available, this will be placed here.

As a placeholderfor that, the authors describe the steps to reproduce the work in this paper. In particular, the authors will describe where to obtain the data and models used in this paper, source code to use when implementing Morris, and where to find the parameters for partitioning the data in the paper. Given this information, the results of this paper should be reproducible.

Data and Models

The Biased-MNIST data can be generated using open source code created by Shrestha et al. (2021) and the ImageNet validation data can be downloaded from the website provided by Russakovsky et al. (2015). The CNN model used to classify the Biased-MNIST can also be found on the cite created by Shrestha et al. (2021), while the pre-trained ImageNet models used in this paper are readily available in PyTorch.

Morris

The Morris method is implemented using the open source sensitivity analysis package, SALIB, (Herman & Usher, 2017), along with Morris parameters described in sections 4 and 5.

Data Partitioning

The data partitioning for Biased-MNIST and ImageNet is described within section 3.

References

Ashish Agarwal, Kedar Dhamdhere, and Mukund Sundararajan. A new interaction index inspired by the taylor series. *CoRR*, abs/1902.05622, 2019. URL `http://arxiv.org/abs/1902.05622`.

Md Zahangir Alom, M. Tarek Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, C Brian Van Esesn, Abdul A S. Awwal, and K. Vijayan Asari. The history began from alexnet: A comprehensive survey on deep learning approaches, 2018.

Wieland Brendel and Matthias Bethge. Approximating cnns with bag-of-local-features models works surprisingly well on imagenet. volume 7, 2019.

Francesca Campolongo, Jessica Cariboni, and Andrea Saltelli. An effective screening design for sensitivity analysis of large models. *Environmental Modelling Software*, 22:1509–1518, 2007.

Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 839–847. IEEE, 2018.

Qiao Ge and Monica Menendez. Extending morris method for qualitative global sensitivity analysis of models with dependent inputs. volume 162, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jon Herman and Will Usher. Salib: An open-source python library for sensitivity analysis. *Journal of Open Source Software*, 2, 2017. URL `https://doi.org/10.21105/joss.00097`.

I. Kononenko, E. Štrumbelj, Z. Bosnic, Darko Pevec, M. Kukar, and M. Robnik-Sikonja. Explanation and reliability of individual predictions. *Informatica*, 37, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Ying Li, Lingfei Ma, Zilong Zhong, Fei Liu, Michael A. Chapman, Dongpu Cao, and Jonathan Li. Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32, 2021.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23, 2021.

Grégoire Montavona, Wojciech Samekb, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73, 2018.

M.D. Morris. Factorial sampling plans for preliminary computational experiments. *Techometrics*, 33:161–174, 1991.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

Manuel Schultheiss, Sebastian Schober A., Marie Lodde, Jannis Bodden, Juliane Aichele, Christina Müller-Leisse, Bernhard Renger, Franz Pfeiffer, and Daniela Pfeiffer. A robust convolutional neural network for lung nodule detection in the presence of foreign bodies. *Scientific Reports*, 32, 2021.

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 *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Robik Shrestha, Kushal Kafle, and Christopher Kanan. An investigation of critical issues in bias mitigation techniques, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

I.M. Sobol. Uniformly distributed sequences with an additional uniform property. *USSR Computational Mathematics and Mathematical Physics*, 16(5):236–242, 1976.

Mateusz Staniak and Przemysław Biecek. Explanations of model predictions with live and breakdown packages. *The R Journal*, 10:395–409, 2018.

Bruno Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering System Safety*, 93(7):964–979, 2008.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *International Conference on Machine Learning*, 34, 2017.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *CoRR*, abs/2006.09994, 2020.

Table 1: Trajectory and radial designs

Point	**Trajectory**	**Radial**
$\boldsymbol{x}_{1,r}$	$a_{1,r}, a_{2,r}, a_{3,r}, ..., a_{n,r}$	$a_{1,r}, a_{2,r}, a_{3,r}, ..., a_{n,r}$
$\boldsymbol{x}_{2,r}$	$b_{1,r}, a_{2,r}, a_{3,r}, ..., a_{n,r}$	$a_{1,r}, b_{2,r}, a_{3,r}, ..., a_{n,r}$
$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{x}_{i,r}$	$b_{1,r}, b_{2,r}, ...b_{i-1,r}, a_{i,r}..., a_{n,r}$	$a_{1,r}, a_{2,r}, a_{3,r}, b_{i-1,r}, ..., a_{n,r}$
$\boldsymbol{x}_{i+1,r}$	$b_{1,r}, b_{2,r}, ...b_{i-1,r}, b_{i,r}..., a_{n,r}$	$a_{1,r}, a_{2,r}, a_{3,r}, ..., b_{i,r}, ..., a_{n,r}$
$\vdots$	$\vdots$	$\vdots$
$\boldsymbol{x}_{n+1,r}$	$b_{1,r}, b_{2,r}, ...b_{i-1,r}, b_{i,r}..., b_{n,r}$	$a_{1,r}, a_{2,r}, a_{3,r}, ..., b_{n,r}$

A MORRIS SCREENING

A.1 MORRIS RADIAL DESIGN

Here we describe in more detail radial and trajectory design used for the Morris screening method. Table 1 shows examples of these designs. The designs are essentially constructed by sampling N values of $\boldsymbol{z} \in \mathbb{R}^{2n}$. The r^{th} sampled vector is split into two vectors, $\boldsymbol{a}_r \in \mathbb{R}^n$ and $\boldsymbol{b}_r \in \mathbb{R}^n$. Both the trajectory and radial design construct $n+1$ points. A trajectory design is created by iteratively replacing $\boldsymbol{a}_{k,r}$ with $\boldsymbol{b}_{k,r}$ in such a way that the i^{th} and $i+1^{th}$ points differ only in the i^{th} element. A radial design is created by replacing components of the sampled point such that the i^{th} and 1^{st} point differ only by the i^{th} component. Table 1 shows an example of this.

The sampling strategy used for the trajectory design is based on sampling on fixed grid levels (Morris, 1991), while the samples for the radial design were generated with Sobol's quasi-random sequence (Sobol, 1976). The trajectory design method focuses on constructing a series of points such that the i^{th} point differs from the $(i-1)^{th}$ point only in the $(i-1)^{th}$ element. The radial design constructs series of points such that the i^{th} point differs from the 1^{st} point in the $(i-1)^{th}$ element.

B BIASED-MNIST MODEL

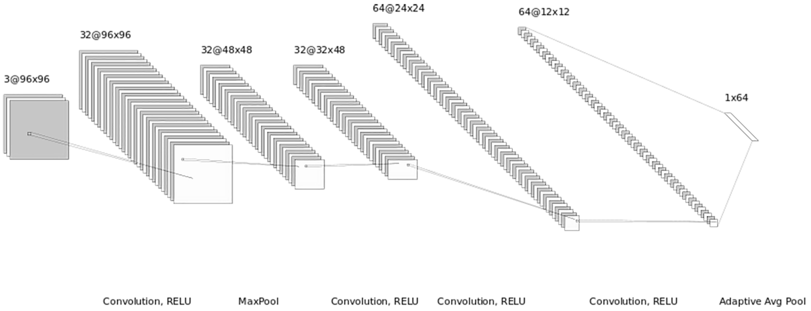


Figure 8: Architecture that is trained on Biased-MNIST data, and used to perform sensitivity analysis on test data.

The model used for classifying Biased-MNIST is a simple CNN architecture. It consists of a batch norm layer, followed by a series of convolution and RELU layers. The architecture is shown in figure 8.

C ImageNet Results

This section of the appendix contains results for the quantitative experiments of MoSIP on different architectures applied to CNN.

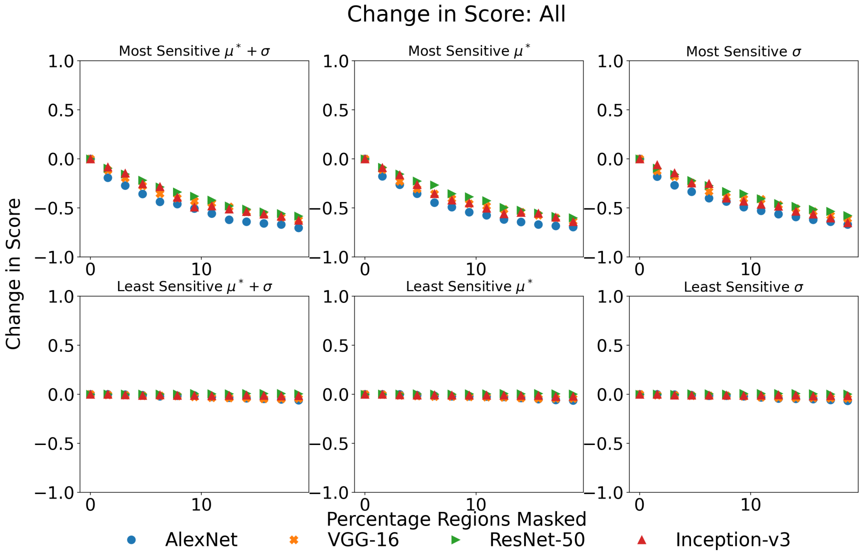


Figure 9: Average change in score of the ground truth component of the model output when certain percentage of regions are masked. The top graphs show when the most important μ^*, σ, and $\mu^* + \sigma$ are masked. The bottom shows when the least important are changed. All the inputs used in this test are those for which the model made accurate predictions.

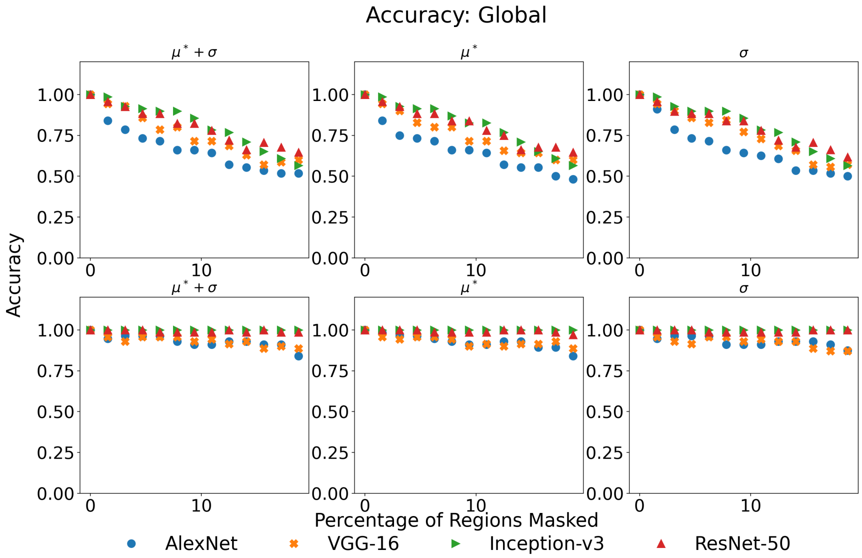


Figure 10: Average change in the accuracy of the models when certain percentage of regions are masked using *global* values of μ^*, σ, and $\mu^* + \sigma$. (TOP) Most Important. (BOTTOM) Least Important.

We also performed experiments to validate that the Morris method accurately selected the most and least important regions of the images both locally and globally. In order to do this, we masked the top-20% of regions in local analysis and observed the change in accuracy, shown in figure 4 in the text, and figure 9, shown here. We finally observe the change in accuracy of masking based on the global results. We note that in the global experiments the same regions are masked in all inputs. Figure 10 shows these results.