Technische Universität Berlin

Faculty of Electrical Engineering and Computer Science
Dept. of Computer Engineering and Microelectronics
Remote Sensing Image Analysis Group

On the Prevalence of Texture Bias in Deep Learning Models for Remote Sensing

Master of Science in Computer Science

April 14, 2025

Oliver Vincent Leon Stoll

Matriculation Number: 382829

Supervisor: Prof. Dr. Begüm Demir

Advisor: Tom Oswald Burgert

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Sämtliche benutzten Informationsquellen sowie das Gedankengut Dritter wurden im Text als solche kenntlich gemacht und im Literaturverzeichnis angeführt. Sofern generative KI-Tools verwendet wurden, habe ich Produktnamen, Hersteller, die jeweils verwendete Softwareversion und die jeweiligen Einsatzzwecke (z.B. sprachliche Überprüfung und Verbesserung der Texte, systematische Recherche) benannt. Ich verantworte die Auswahl, die Übernahme und sämtliche Ergebnisse des von mir verwendeten KI-generierten Outputs vollumfänglich selbst. Die Satzung zur Sicherung guter wissenschaftlicher Praxis an der TU Berlin vom 8. März 2017. https://www.static.tu.berlin/fileadmin/www/10000060/FSC/Promotion_-__Habilitation/Dokumente/Grundsaetze _gute_wissenschaftliche_Praxis_2017.pdf habe ich zur Kenntnis genommen. Die Arbeit wurde bisher nicht veröffentlicht und keiner Prüfungsbehörde vorgelegt.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, April 14, 2025

.....

Oliver Stoll

Abstract

Deep learning models have demonstrated remarkable performance across various domains, including remote sensing (RS). However, the reliance of these models on specific image features, such as spectral content, texture, and shape, remains an open question, particularly in comparison to the established understanding of feature biases in computer vision (CV). In recent years, the long-lasting notion that deep neural networks rely on high-level shape information in classification tasks has come into question, by works reporting that convolutional neural network (CNN) architectures trained on CV datasets exhibit a bias towards texture features. This thesis investigates the prevalence of texture bias, as well as the relative reliance on different image features in deep learning models applied to the RS domain. In a theoretical analysis, we find that the previously reported texture bias in CV might be influenced by theoretical and methodological issues in the applied evaluation protocol. By addressing these limitations, this work introduces a novel dataset-agnostic protocol for assessing feature reliance utilizing a set of feature-suppressing image transformations. Through the proposed protocol, we systematically evaluate the individual importance of spectral content, texture, and shape features across datasets of the RS domain and contrast them with datasets of the CV domain. Our findings indicate that RS datasets are predominantly reliant on spectral features, with texture and shape features playing an equal secondary role. A class-wise analysis further shows this reliance on spectral features to be dominant across all classes of both datasets, highlighting the universal importance of spectral features. In contrast to previous findings on texture bias in CV, our results reveal that models trained on CV datasets exhibit a lower reliance on texture and spectral content features, but a higher reliance on shape features. Overall, our results underscore fundamental differences in feature reliance between RS and CV, providing a foundation for future work in optimizing model design and training strategies tailored to the unique characteristics of the RS domain.

Zusammenfassung

Deep-Learning-Modelle haben in vielen Bereichen der künstlichen Intelligenz basierten Bildverarbeitung, darunter die Fernerkundung (RS), bemerkenswerte Leistungen erzielt. Dennoch ist unklar, inwieweit sie verschiedene Merkmale von Bildern wie Form, Textur und spektrale Inhalte verwenden, um Objekte zu erkennen. In jüngerer Zeit wurde die langjährige Annahme, dass tiefe neuronale Netze vor allem kantenbasierte Form-Merkmale für Klassifikationsaufgaben nutzen, von Studien infrage gestellt, die zeigen, dass CNN-Architekturen, die auf CV-Datensätzen trainiert wurden, eher eine Präferenz für Textur- als für Form-Merkmale aufweisen. Diese Arbeit untersucht die Verbreitung einer Texturpräferenz und die relative Abhängigkeit von unterschiedlichen Bildmerkmalen in Deep-Learning-Modellen für RS. Eine theoretische Analyse legt nahe, dass die zuvor berichtete Texturpräferenz in CV durch theoretische und methodologische Probleme im angewandten Evaluationsprotokoll verursacht sein könnte. Durch Verbesserung dieser Einschränkungen führen wir ein neuartiges, datensatzunabhängiges Protokoll ein, das anhand einer Reihe von merkmalsunterdrückenden Bildtransformationen die Merkmalsabhängigkeit von Modellen auf einem Datensatz ermittelt. Mit diesem Protokoll wird die individuelle Bedeutung von spektralen, Textur- und Formmerkmalen in RS-Datensätzen systematisch bestimmt und mit CV-Datensätzen verglichen. Die Ergebnisse deuten darauf hin, dass Modelle auf RS-Datensätze in hohem Maße auf spektrale Merkmale angewiesen sind, während Textur- und Formmerkmale eine gleichwertige, nachgeordnete Rolle spielen. Eine klassenweise Analyse bestätigt die Dominanz spektraler Merkmale für alle Klassen in beiden Datensätzen. Im Gegensatz zu früheren Berichten über eine Texturpräferenz in CV weisen Modelle, die auf CV-Datensätzen trainiert wurden, eine geringere Abhängigkeit von Textur-Merkmalen, aber eine stärkere Präferenz für Form-Merkmale auf. Insgesamt verweisen die Ergebnisse auf grundlegende Unterschiede in der Merkmalspräferenz von Modellen zwischen RS und CV und schaffen eine Basis für zukünftige Arbeiten zur Optimierung von Modellarchitekturen und Trainingsstrategien, die auf die besonderen Eigenschaften der RS-Domäne zugeschnitten sind.

Contents

List of Tables

List of Figures

1 Introduction

In recent years, deep neural networks (DNN) have demonstrated remarkable capabilities, establishing themselves as powerful tools for image recognition tasks on various image domains, including remote sensing (RS). However, because of their immense inherent complexity, DNN are often regarded as "black-box" models with limited understanding of how they exactly utilize image features to process and classify objects. A long lasting hypothesis that convolutional neural networks (CNNs), a standard class of DNNs used for image task, understand complex object shapes (e.g., the outline of an object) by hierarchically assembling smaller edges in their earlier layers, similar to the human visual processing system. This notion has, however, seen counterarguments in the last years. Previous works find that CNNs exhibit a bias towards texture features when presented with conflicting features for shape and texture [1], and can perform classification tasks comparably well even without the ability to utilize the global shapes of objects [2, 3]. These findings could suggest that texture features play a more central role than shape features in decision processes of CNNs in image recognition tasks. Nonetheless, despite substantial progress in exploring such behavioral phenomena of CNNs, a comprehensive understanding of how these networks combine and quantitatively depend on different types of image features during object recognition remains elusive. Directly investigating the relative importance of different image features is, therefore, crucial for advancing our understanding of CNNs decision-making processes and improving their effective application, interpretability and robustness.

Although the topic of feature bias has been studied in the CV domain, limited research has focused on feature bias within the RS domain or a direct comparison of the two, although there appear to be substantial differences in the predictive strength of different image features. While the importance of spectral information is shown for the RS domain [4, 5, 6], differences in predictive strength of spectral characteristics have seen little direct comparison with other domains such as CV. Additionally, it could be conceived that shape features, due to differing data characteristics, play a less important role for classification tasks in the RS domain, compared to the CV domain. RS imagery is predominantly captured from a top-down perspective, which contrasts with the diverse viewpoints present in CV datasets. Furthermore, RS data is often affected by varying spatial resolutions, ranging from low resolution (greater than 10x10 m) to very high resolution (less than 1x1 m, as in aerial photography). These varying resolutions introduce unique challenges: at low spatial resolutions, shape features of many objects of land use classes such as roads, buildings, or vehicles become indistinct, while naturally formed land cover classes exhibit randomness in their object boundaries that might reduce the predictive value of shape-based features. In contrast, image recognition in the CV domain is commonly assumed to benefit significantly from shape features [7]. Humans have evolved to rely heavily on shapes for object understanding [8], making shape a likely versatile feature category for recognizing objects like animals, devices, or vehicles, with many distinctly recognizable unique shape features (e.g., paw of a cat, wheel of a car) .

This disparity raises questions about the transferability of insights about image features from CV to RS. For instance, the phenomenon of texture bias as reported by Geirhos et al. [1] for CNNs trained on ImageNet [9] (as a representative dataset for the CV domain), where models may prefer texture features for predictions, could manifest differently in RS data. Given the higher relevance of spectral content in RS, it is plausible that less texture-based information is used for predictions, and the specific biases of models trained on RS data could differ significantly from those observed in CV. However, without explicit quantification of these relative differences in the importance of image features in different domains, it is commonly assumed that the findings of the domain CV hold up in some similar fashion for CNNs trained on other domains such as RS. Quantifying differences of feature importance between the RS and CV domains could prove essential for designing domain-specific training strategies, such as specifically choosing data augmentations or pretext tasks for self-supervised learning aligned with relative feature importance, which could lead to significant improvements in model performance and robustness on RS data. Furthermore, insights into the importance and bias of image features across domains and model architectures can guide the selection or development of models better suited for RS applications, ultimately advancing our ability to extract meaningful information from RS data.

Finally, despite significant advancements, existing research specifically on the relative preferability of image feature comparisons has focused largely on CNN architectures. However, the rise of transformer-based architectures for vision applications has brought new perspectives to the field. Transformers, unlike CNNs, lack the convolutional layers that introduce inductive biases favoring local spatial relationships, such as textures, which are central to CNNs. This fundamental architectural difference may influence the importance of various image features, including low-level details. Transformers have achieved state-of-the-art performance in many CV tasks following the introduction of the initial vision transformer (ViT) by Dosovitskiy et al. [10]. Their rapid adoption in recent years underscores the need to better understand how they process and prioritize image features compared to classical CNNs. While a lower exhibited bias towards texture was found for vision transformers compared to CNN architectures, such comparisons were not extended towards other feature types or to reliances on feature types for classification. A deeper investigation into such differences could shed light on how different inductive biases of both architecture categories affect their relative reliance on different image features. This could highlight whether transformers, commonly assumed to not share the same inductive bias of CNNs toward localized information, exhibit a reduced reliance on texture-based features or adopt alternative mechanisms to rely on such texture.

By addressing these gaps in understanding the role of different image features, this thesis aims to investigates the prevalence of texture bias, as well as provide a comprehensive quantification of the relative importance of spectral, texture, and shape features for classification tasks in the RS domain. In a theoretical analysis of previous evaluation methods, our work finds potential limitations with previous studies reporting a texture bias of CNN. By addressing these limitations, this work introduces a novel dataset-agnostic evaluation protocol for assessing feature reliance of models on spectral, texture, and shape features by systematically applying feature suppressing transformations at test time. This approach is applied to various model architectures and two datasets in the RS domain, BigEarthNet-S2 [11] and DeepGlobe [12], and compared to datasets from the CV domain, ImageNet [9] and Caltech [13], to quantify the relative importance of these feature types within RS and compare them with the observed

relative importance of features in the CV domain. Finally, this analysis is conducted class-wise for the BigEarthNet and DeepGlobe datasets, comparing differences in importance image features exhibit for correct classifications of specific classes.

This thesis is structured into seven chapters, each addressing specific aspects of research into the importance of image features across different domains, such as RS, and their evaluation. Chapter 1 introduces the motivation behind this thesis, outlining the significance of studying image feature reliance in the RS domain and its potential applications. Chapter 2 reviews related work, providing a foundation for the research by defining key terms and concepts essential for understanding the subsequent chapters. Chapter 3 examines the limitations of existing research on feature bias, identifying methodological and theoretical gaps that this study aims to address. Chapter 4 proposes a novel feature-reliance evaluation protocol designed to empirically assess feature reliance across different datasets, offering a robust framework for evaluating image feature importance. Chapter 5 details the datasets, models, and experimental setup used in the study. This includes a comprehensive description of dataset characteristics and an overview of the chosen model architectures employed in the evaluation. Chapter 6 presents the experimental results, including a quantitative analysis of feature reliance when single feature categories are impaired and the importance of individual remaining features. Additionally, the chapter explores class-wise results for selected RS datasets and examines the effect of model architecture on feature reliance. Finally, Chapter 7 concludes the thesis by summarizing the key findings, discussing the limitations of the study, and proposing future research opportunities in the area of feature reliance for RS and other domains.

2 Related Work

The interpretation of image feature importance in deep learning has undergone a significant shift, sparking growing interest in understanding how DNNs utilize different image features to correctly identify and classify objects. An earlier long-lived intuition was that CNNs combine low-level features to increasingly complex shapes to gain an intrinsic understanding of objects. This understanding encapsulated by the work of Kriegeskorte et al. [14], was supported by additional findings that CNNs "implicitly learn representations of shape that reflect human shape perception" [15]. Additionally, CNNs were the currently most predictive models for human ventral stream image recognition [16, 17] and that visualization techniques like Deconvolutional networks [18] often highlight shape information of entire object parts, in high-level learned features of CNNs.

However, this concept is contested by emerging studies on the insignificance of global shape, the silhouette or outline of an object, which discovered that CNNs are capable of accurately classifying images even when global shape features are not applicable for the classification process. Baker et al. demonstrate that CNN models could maintain comparable performance levels even when the global shape structure of objects was disrupted [2]. This was achieved by evaluating model performance on images shuffled into 4x4 patches, effectively destroying the overall global shape, a feature-suppressing transformation also employed in this thesis. Brendal et al. show that constraining the model's perceptive field does not have an adverse effect on classification performances [3]. The authors introduce BagNet, a Bag-of-local-features DNN architecture designed to classify images based solely on small local features by processing only limited patches of the image at a time, disregarding their spatial arrangement. In contrast to the previous assumption, BagNet achieved notable accuracy scores on ImageNet comparable to current state-of-the-art models, without utilizing the global shape of the objects.

In a parallel line of study, Geirhos et al. [1], in a foundational work for relative comparisons of image feature importance, demonstrate that when faced with images exhibiting conflicting features, models exhibit a preference for texture over shape cues. To investigate this, a cue-conflict image dataset was devised, which was constructed by combining individual texture and shape images through style transfer techniques into images with conflicting shape and texture features. When tasked with classifying these cue-conflict images, CNNs categorize them predominantly according to the original texture classes, while humans mostly classify the images with conflicting cues according to the original shape classes. This unexpected finding, that CNN contrary to previous assumptions exhibit a bias towards texture-cue based classification, led to the coinage of the term texture bias. This reported phenomenon is cited as a potential explanation for the lack of prediction robustness and is creating intense research interest in determining the intrinsic image features models prefer for image classification.

Numerous subsequent studies followed, adopting both the intuition of Geirhos et al. [1] that CNNs exhibit a preference for texture rather than shape features in their classification

processes, and the proposed cue-conflict evaluation protocol to determine the bias of models towards texture or shape features. Hermann et al. investigate the origin of this feature bias toward texture, finding that the properties of the data, as well as the applied data augmentations for training models, play a major role in how prevalent it is exhibited [19]. Mummadi et al. explore the connection between feature bias and classification robustness and found that feature bias toward shape is more a correlation with increased robustness, instead of causing factor [20]. By harnessing a redesigned label space, Chung et al. [21] are able to mitigate the found feature bias towards texture in favor of a more balanced feature preference. Furthermore, Naseer et al. extend the research on feature bias to vision transformer architectures [22], finding the reported bias towards texture features to be less prevalent for such compared to CNN. Additionally, for an application to the RS domain, Tang et al., following the intuition of Geirhos et al., train models using additional edge images obtained by using edge detection methods as data augmentation, to increase classification performance by allowing models to exhibit a stronger bias towards shape features [23].

Others proposed different evaluation methods for feature bias or the importance of different image features. Kalischek et al. propose a different method for assessing feature bias, using a combination of different data sets, including ones that feature a reduced amount of texture or shape features [24]. However, this evaluation method still relies on the perspective of a binary feature bias on image features, which we do not adopt for this thesis, as will be described in Chapter 3. Ge et al. [25] quantitatively investigate the importance of different image features by emulating the human visual system, encoding each feature type separately. By this, each feature categories contribution to predictive capabilities could be measured, an approach close to the goal of this thesis. However, object shape features were represented by global shape alone, disregarding the predictive strength of local shape contributions.

Hermann and Lampinen conducted an evaluation of learning behavior of individual image features, color, texture, and shape, both independently and in combination, by directly controlling their predictive power through a custom-generated dataset [26], closely related to the approach of this thesis. Using this, they reported findings on the behavior of models trained on the dataset in regards to the image features utilized, such as that models prefer to learn easier features even when those do not exhibit the same predictive power and that weakly predictive features can suppress more strongly predictive, but more difficult ones. However, the evaluation is based on a synthetic data set to offer the possibility of deliberately controlled predictive power. In contrast, this thesis employs an evaluation protocol designed for application across real-world datasets, with the aim of evaluating feature reliance within specific domains.

3 Limitations of Feature-Bias as a Research Perspective

This chapter critically examines the limitations of feature bias as a research perspective by introducing an alternative definition of image features, highlighting issues in the current understanding of image features in the context of feature bias, identifying methodological issues in the foundational work of Geirhos et al., and addressing misinterpretations regarding feature bias. To achieve clarity when discussing **(1)** biases towards specific image features and **(2)** the significance of various image features within the RS domain for classification tasks, we introduce two terms to represent these phenomena. The tendency of models to exhibit a preference of a particular image feature type over others for classification tasks, as observed on a specific dataset or within a specific domain, is referred to as *feature bias*. This term extends previously established terminology of a dichtonomy of either texture or shape bias, introduced by Geirhos et al. [1], to a more general notion that also includes a bias towards spectral features. In addition, we define a dependence of models on the availability of certain image feature types for classification tasks, as *feature reliance*. Unlike *feature bias*, *feature reliance* directly highlights the importance of image features for an effective classification on certain datasets. The necessity of distinguishing between *feature bias* and *feature reliance*, as well as their implications for the evaluation of the importance of image features in the RS domain, will be discussed during this chapter.

3.1 Image Feature Definition

To highlight limitations in the current literature on feature bias in DNN, this section revisits a general definition of image features for object recognition, emphasizing a clear distinction among features to assess their relative importance for classification performance on a given dataset. We identify three different types of image features that a model trained on an image recognition task may use for prediction: shape, texture and spectral features. To provide a clear understanding of these features, we will define the scope each one individually.

We define shape features as a combination of descriptors at different scales, referred to as global and local shape, primarily expressed through unique continuous edge information that form subparts within an class object or its outline. Global shape descriptors are defined as the silhouette or outline of an class object , while local shape descriptors are defined as unique geometric traits around distinctive points [27]. Global shape conveys the macroscopic form of an object, representing its overall outline or silhouette. This is achieved through edge information that defines the object's boundary, separating it from adjacent objects, and through the relative proportions and spatial arrangement of the silhouette. In contrast, local shape describes finer-scale geometric details within the object or along its boundary. These details often

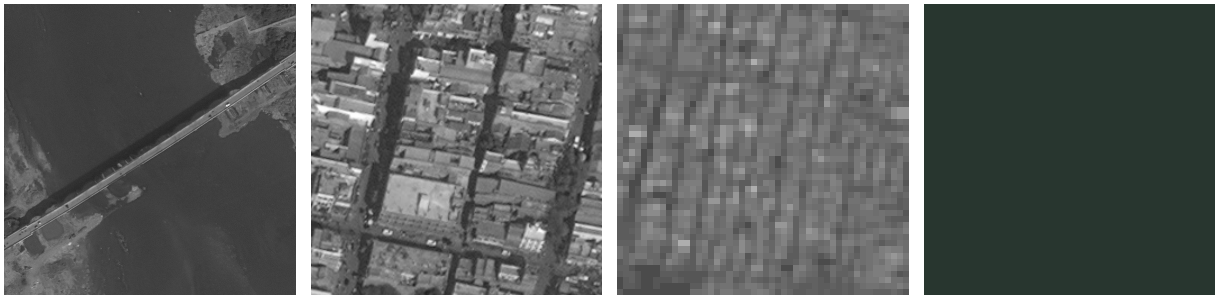
correspond to non-repetitive unique edge patterns that provide additional insight into the object's structural intricacies. Together, global and local shape descriptors form a comprehensive representation of an object's geometry solely based on unique continous edges that compose subparts of the class object, balancing holistic form with localized detail.

We define texture features as local variations in pixel intensity changes (i.e., contrasts), which are characterized by spatial relationships between pixels. Several definitions capture this concept. Armi and Fekri-Ershad define texture as "a function of spatial variation of the brightness intensity of the pixels" [28]. Similarly, Russ describes image texture as "a descriptor of local brightness variation from pixel to pixel in a small neighborhood through an image," forming identifiable repetitive patterns within an object [29]. These patterns are not determined by absolute pixel values, as incorporating such values would include absolute brightness within the definition, making it dependent from individual lighting conditions. Instead, texture is observed through the relative differences between pixel intensities, which form recurring arrangements. This ensures that texture is described by its spatial and relational properties, distinguishing it from other image features such as spectral content.

We define spectral content as the distribution of an object's reflectance or emittance across various wavelengths of the electromagnetic spectrum [30]. In CV images, this is represented as color information encoded in the red, green, and blue (RGB) channels. To maintain clarity and consistency, this information is referred to as spectral content throughout this work. Spectral content can be utilized as image features in various ways, such as through direct spectral analysis of individual absolute pixel values across image channels or by leveraging general statistical properties of the image, such as image histograms.

Following these definitions, the distinction between global shape and texture features is straightforward: global shape features refer to the singular silhouette or outline of an object, while texture features are characterized by repetitive patterns that can occur across any portion of the object. However, the differentiation between local shape features and texture features requires a more nuanced understanding, as both feature types exist at similar scale and can overlap in their spatial dimensions. Additionally, their interpretation may shift depending on spatial resolution, or the context of the class represented by the features. For example, in the RS domain, the edges of the roof of a house may represent a local shape feature in very high spatial resolution imagery (e.g., 0.5x0.5 m per pixel). In such cases, the edges are unique, distinguishable from other edges in the image, and by themselves indicative of a land use land cover class like "human built" or "houses". Conversely, in lower spatial resolution imagery (e.g., 10x10 m per pixel), the same housing roof may appear as only a few pixels. An example of this difference can be seen in Figure 3.1. At this scale, its individual edges are no longer distinguishable. In this case, it becomes part of a larger repetitive pattern formed by neighboring roofs and streets, collectively defining a texture feature of an urban area. To differentiate between these features, it is therefore essential to consider the size of the feature in relation to the object it represents. We define the scale of local shape features to occupy at least 10x10 pixels or approximately 1% of the class object's area to be uniquely distinguishable and predictive of a specific class, to be defined as such. Conversely, edges that contribute to repetitive patterns with relative pixel contrasts, without extending over larger dimensions or exhibiting uniqueness within the pattern, are classified as texture. This distinction ensures a consistent interpretation of both feature categories across varying spectral resolutions and class contexts.

We can find many examples for these three feature types being distinguishable image fea-

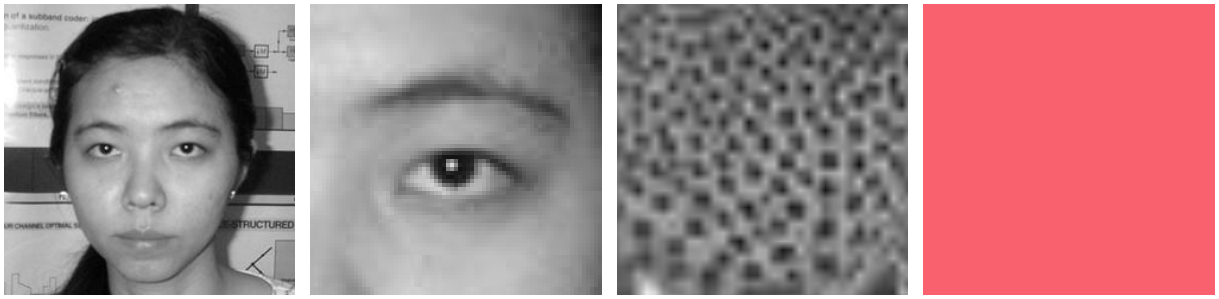


(a) (b) (c) (d)

Figure 3.1: Examples of feature types in the DeepGlobe dataset, demonstrating distinctiveness of various spatial and spectral characteristics. Shape features are classified into **(a)** global shape, represented by the outline of a bridge, and **(b)** local shape, illustrated by the unique forms of individual houses at very high spatial resolution of 0.5x0.5m. Texture is captured through **(c)** repetitive patterns of houses and streets, emphasizing spatial organization when recorded at a lower spatial resolution of 10m×10m. Spectral content is illustrated by **(d)** the spectral value of a water body as recorded by the given RGB channels. Non-spectral features are shown in monochrome to highlight their distinction.

tures in images of the RS domain, seen in Figure 3.1. An example of recognizable global shape features are the distinct outlines of certain classes, such as bridges or airplanes in high spatial-resolution images. For instance, the elongated linear structure of a bridge, often spanning over a water body as seen in Figure 3.1a, serves as a defining characteristic of its global shape. Similarly, the unique silhouette of an airplane, with its wings and fuselage, can differentiate it from other objects. Such global shape features remain distinguishable regardless of variations in spectral content or texture patterns of the object, making them a potential generalizable image feature to depend on. An example of local shape features being recognizable includes the inner edges of houses or industrial buildings as seen in Figure 3.1b, which indicate the presence of an urban area or even a specific type of urban area. Similarly to global shape features, such local shape features remain identifiable even when texture patterns or spectral content vary. Their unique geometric characteristics can indicate class-specific features, such as the type of urban area or the function of the structures. An example of recognizable texture features is the pattern formed by housing roofs and streets as seen in Figure 3.1c, which can indicate a suburban or urban area at high to medium spatial resolution values. Unlike global or local shape features, no singular unique edges are present to predict the area, and the spectral values may be similar to those of other classes. However, the dense and repetitive pattern of streets and housing roofs can be distinctive across all classes, making it indicative for an urban area class. An example of spectral content being a predictive image feature is the reflectance values of a given land cover class that can be easily differentiated by its spectral signature, such as a water body of the DeepGlobe dataset as seen in Figure 3.1d. In such cases, neither texture nor shape features are necessary for classification, as the distinct spectral values of the class allow it to be directly and linearly decodable. This makes spectral content a highly effective feature for identifying classes with unique spectral characteristics.

In the CV domain, comparable examples of global shape, local shape, texture, and spectral content can be observed, as illustrated in Figure 3.2 showing examples of the Caltech dataset. An example of a recognizable global shape is the outline of a human face as seen in Figure



(a) (b) (c) (d)

Figure 3.2: Examples of image feature types from the Caltech dataset, illustrating distinct spatial and spectral characteristics. Shape features are divided into **(a)** global shape, exemplified by the face outline, and **(b)** local shape, represented by the detailed form of an eye. Texture is characterized by **(c)** the repetitive pattern of leopard fur, capturing local spatial organization. Spectral content is demonstrated through **(d)** skin color of flamingos, emphasizing its unique spectral properties. Non-spectral features are shown in monochrome to highlight their distinction.

3.2a, where the shape is determined by the head contour, jawline, ears, and hairline. These features provide a recognizable structure distinguishable from others. An example of local shape features could be the distinct shape of a human eye as seen in Figure 3.2b. Local shape focuses on small-scale, unique contours, such as the oval-shaped outline of the eye, the curve of the eyelid, or the sharp edge of the iris. These localized edge information combined provide a feature that is distinct for the object class. An example of recognizable texture features could be the fur of a leopard as seen in Figure 3.2c. The repetitive pattern of irregularly shaped spots scattered across the fur forms a distinctive texture. Again, the texture feature captures this pattern independently of spectral content, enabling reliable detection even across variations in fur coloration or under different lighting conditions. An example of spectral features being solely distinguishable can be the pink or reddish hues which are distinctive and serve as strong indicators for the flamingos class in the dataset, as seen in Figure 3.2d. When pink hues are not commonly present in other classes or in significant amounts, the spectral reflectance properties can provide a unique signature that distinguishes flamingos from other objects in the scene.

3.2 Current Understanding of Image Features in the Context of Feature-Bias

This section contrasts our proposed definition of image features as described in the previous Section 3.1, with the current interpretation of image features in the context of the research of feature bias toward texture or shape features, where we observe possible shortcomings with regard to the clear distinction between feature types.

In the field of image feature analysis, only two major image feature categories are commonly regarded: texture and shape. Furthermore, shape features are predominantly understood as being defined by global shape. This perspective is reflected in the current literature on feature bias, where various experimental techniques, such as the use of silhouettes [24, 25, 2] or the disruption of global shape by splitting it into smaller patches [20, 3], are employed to analyze the use of shape features. Such technique however, do not target local shapes in their disruption,

therefore leaving shape features partially available. Still, these experiments are often either directly referred to by authors or are later interpreted by others as illustration of the effects of missing shape features. For instance the findings of Brendal et al. [3] and Baker et al. [2] finding a low importance of global shape features for classification tasks, were commonly interpreted as evidence of a lower importance of all shape features. In line, the measured differences in classification performances when destroying global shape are subsequently interpreted as illustrating the effects of absence of shape features [1, 21, 31, 32]. However, this represents a significant misinterpretation of the scope of shape features, as such research merely demonstrated the relative importance of global shape features, which may not be as crucial for a model's understanding of objects. The potential importance of local shape features however is often overlooked, even though smaller-scale edges may be easier for models to process and could provide stronger predictive power than complex global shapes.

Finally, texture is often conceptualized as encompassing the interior of an object, effectively including everything except the global shape and silhouette [1, 33, 25, 20]. Within this broad understanding of texture, spectral content is almost always subsumed, resulting in the combination of two distinct image features under the single term "texture." Additionally, by understanding texture as the inside of an object, local shape features were sometimes incorrectly incorporated into texture too, as for instance directly observable in applications that utilize style-transfer techniques. Consequently, spectral features with a significant importance for the RS domain are not recognized as an independent and distinct, which obscures the potential for models to exhibit bias toward it.

This highlights two primary issues: Firstly, the possibility that spectral content alone can contain valuable information for object recognition, independent of its combination with repetitive patterns, is often overlooked. This omission significantly reduces the understanding of the importance of specific image features. For example, in the RS domain, where spectral content plays a critical role, adopting a simplistic binary perspective, choosing between texture and shape, as proposed by Geirhos et al., could lead to misleading conclusions. Such an approach might neglect the potential insignificance of both texture and shape features compared to spectral content. As a result, one might incorrectly conclude that models trained on RS data are predominantly biased toward texture features. However, if spectral features were analyzed as an independent feature category alongside texture and shape, the findings would likely provide an entirely different feature understanding of the same domain. Secondly, the possibly significant role of local shape information, as a component of shape features, is largely overlooked and remains unexplored in current research regarding feature bias between texture and shape. If local shape features were regarded as a primary image feature alongside global shape, representing overall shape information, then insights regarding bias toward texture or shape features could be fundamentally altered.

3.3 Methodological Issues with Feature-Bias Evaluation

The following section highlights how the foundational work of Geirhos et al.[1] may contain two significant methodological issues that potentially limit the validity and applicability of their reported findings and evaluation methods. Research on feature bias has gained significant attention following the findings reported by Geirhos et al. As outlined in Chapter 2, subsequent research has widely adopted the understanding that models are inherently texture-

biased. Furthermore, studies have employed the same cue-conflict evaluation protocol proposed by Geirhos et al. to assess feature bias in their experiments. As a result, a substantial body of work builds on the findings and methods introduced by Geirhos et al., relying on them for investigating the relative bias of models towards image features.

Geirhos et al. conducted experiments to analyze two image feature categories, texture and shape, and their relative favorability for classification tasks of CNN. This analysis was carried out using cue-conflict images, which contain conflicting shape and texture features of two merged input images, allowing the researchers to test whether CNN and humans would favor one feature type over the other, when presented with both. For these images of conflicting feature cues, both humans and models were tasked with classifying them. The predictions were then compared with the original class label of the texture and shape input image. The percentage of "correct" predictions, those that classified the conflicting image based on either the original texture or shape label, was then computed and compared. For instance, if participants classified 10 out of 100 images by the original shape category and 30 out of 100 by the original texture category, this result was interpreted as a texture bias of 75%. A bias value of 50% was considered neutral, while any deviation in favor of one feature category was labeled as shape or texture bias, depending on the preferred feature type.

The cue-conflict images used in the experiments by Geirhos et al. were created using the style transfer method originally proposed by Gatys et al. [34]. In this process, an original object image, whose class label was later determined by the underlying class label of the "shape" feature, was taken as the base input. Style transfer was then applied to the original shape image using another input image serving as style input to be applied, which object category was subsequently labeled the underlying "texture" category. A visualization of the cue-conflict experiment can be seen in Figure 3.3. Through this method, the authors aimed to create images in which the original object retained its shape features, while the texture of the object was replaced with the texture information derived from the style input image. This approach was intended to generate images containing conflicting shape and texture features associated with different object categories, allowing for a direct evaluation of any preference of models towards either feature type for classification.

The results of the experiment reported that CNNs, when presented with the conflicting feature cues primarily predicted the original texture label, whereas humans primarily chose the original shape label in their predictions. This divergence in category labeling between humans and models highlighted a striking difference in preference for image features. Due to the strong difference in predictive behavior, these findings were interpreted as evidence that models, in contrast to humans, exhibit a bias towards texture features. This interpretation highlighted an apparent fundamental difference in the way humans, previously known to be shape biased, and CNN process visual information.

3.3.1 Issues with Image Features in Cue-Conflict Dataset

However, when analyzing the generated images within the cue-conflict dataset, we observe behavior not in line with the original aims of the preservation of pre-existing shape features and inclusion of other image features by the style transfer process. Example images of the cue-conflict dataset shown in this section illustrate different issues, including application of texture features outside of object boundaries, distortion of shape features, and introduction of

(a) Texture image
81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image
71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% **Indian elephant**
26.4% indri
9.6% black swan

Figure 3.3: Example images of the cue-conflict experiment taken from [1], visualizing both input images for **(a)** texture cues and **(b)** shape cues, and **(c)** the by style-transfer generated image of conflicting cues. Classification predictions of a standard ResNet-50 shown below.

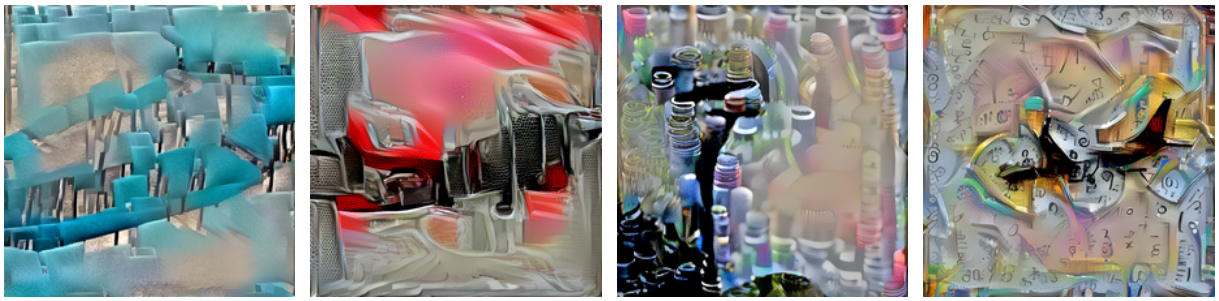
spectral content of texture cues, all which are contrary to original design goals of the cue conflict dataset. This behavior could be due to the fact, that style transfer was never explicitly designed to selectively replace only texture features of objects while leaving other features unaffected. The found discrepancies of image features in the resulting cue-conflict images might influence why models exhibit a preference for texture cues over shape cues.

(a) (b) (c) (d)

Figure 3.4: Example images taken from [1], of cue-conflict dataset reducing usability of shape features. Images show examples of **(a)** cat shape merged with elephant texture with reduced contrast of global outline, **(b)** bird shape merged with oven texture with original shape features partially removed, **(c)** keyboard shape merged with clock texture with original shape features obscured, and **(d)** airplane shape merged with tractor texture cues with original shape features not clearly differentiable.

In Figure 3.4, we can observe the global outline of objects in the cue-conflict images negatively affected by the style transfer process. The overall shape of the objects is only partially preserved, and style transfer significantly reduces the contrast of the outline. This reduction in contrast arises because the background surrounding the object, as well as the interior regions of the object, are altered by the introduction of texture features. Consequently, pixel values inside and outside the object become more similar, leading to diminished contrast between the object and its background. Even in the example image provided by the authors, a cat shape with an

elephant texture cue exhibits a gray background instead of the original white. The cat itself also appears gray, illustrating how the contrast between objects and their background can be reduced by style transfer.



(a) (b) (c) (d)

Figure 3.5: Example images of cue-conflict dataset taken from [1], where shape features of original texture-cue image were unwillingly introduced. Images show examples of **(a)** boat shape merged with chair texture-cue, **(b)** airplane shape merged with truck texture-cue, **(c)** bird shape merged with bottle texture-cue, and **(d)** airplane shape merged with clock texture-cue.

In some cases, the style transfer process introduces not only texture features but also shape features from the style input image. In the first example of Figure 3.5, the outline of a boat was not filled with the texture of a chair, as intended, but instead with multiple instances of the shape of a chair. This introduces an unwanted type of conflict, where the image does not merely present a contrast between texture and shape features but instead contains conflicting shape information. Specifically, the global shape corresponds to one class (e.g., boat), while local shape elements from another class (e.g., chair) are simultaneously present. Such conflicts complicate the interpretation of feature preference as they do not align with the intent of isolating differing shape and texture cues for evaluation.

(a) (b) (c) (d)

Figure 3.6: Example images of cue-conflict dataset taken from [1], where features of the original texture-cue were applied to the image background beyond object boundaries. Images show examples of **(a)** cat shape merged with elephant texture-cue, **(b)** bicycle shape merged with knive texture-cue, **(c)** bear shape merged with clock texture-cue, and **(d)** chair shape merged with car texture-cue.

In other instances, the style transfer process often applied the texture cue indiscriminately, affecting not only the original object but also the background, as can be seen in all images of Figure 3.6. This results in the texture cue covering a significantly larger portion of the image than what would typically occur if the texture were constrained to the object's boundaries.

By applying the texture cue across the entire area of the image, the signal strength associated with the texture cue was likely increased beyond what would be expected under normal image conditions. This disproportionate presence of the texture cue may have further contributed to a stronger preference for texture-based classification in the models.

Finally, spectral features of the shape cue object were often completely replaced by those of the texture cue during the style transfer process, as seen in virtually all examples of Figures 3.5, 3.4 and 3.6. As a result, the signal strength of the spectral content associated with the original texture class is present alongside texture features, which may inadvertently bias the classification towards texture-cues. This further complicates the intended isolation of shape and texture features, as the spectral properties of the image may disproportionately favor texture-cues predictions.

Overall, the style transfer process appears to disproportionately favor classification decisions based on texture cues. This bias arises from several factors: the reduction of usability of the preexisting shape cue, the introduction of additional spectral content and shape features of the original texture class, and the indiscriminate insertion of texture features across both the object and the background. These unexpected alterations likely enhance the signal strength of the texture cue. As a result, it can reasonably be suspected that the texture cue dominates the overall signal strength present in the cue-conflict images. This imbalance undermines the intended parity between shape and texture features, calling into question the robustness of the reported results of the cue-conflict experiment. Consequently, the main findings of a texture bias in CNNs, as reported by Geirhos et al., may be less conclusive than initially assumed.

3.3.2 Unintentional Influence on Human Decisions

However, contrary to models, humans predominantly base their decisions on shape features, an observation in contrast with our argument that the cue-conflict dataset was biased towards texture-cue based classification. We attribute this discrepancy to two main reasons, with the second being a significant issue in the observed experimental design:

(i) Firstly, we contend that humans are inherently biased towards shape features in their object recognition process, a phenomenon that has been widely established [8]. This natural preference for shape likely influenced classification decisions, even when texture cues were prominent in the images.

(ii) Secondly, in a possibly severe methodological issue, we observe that the human participants in the experiment conducted by Geirhos et al. may have been inadvertently biased toward shape-based decisions. The authors explicitly used neutral language when instructing participants, neither mentioning texture or shape. The original instruction given to participants stated:

"Click on the object category that you see in the presented image; guess if unsure. There is no right or wrong answer, we are interested in your subjective impression."

However, participants gave their classification predictions by clicking on one of sixteen options representing the object categories, which were **annotated by corresponding shape icons**, as depicted in Figure 3.7. By using shapes as representatives for the answer categories, humans may have been unintentionally presented with a context cue on which type of image

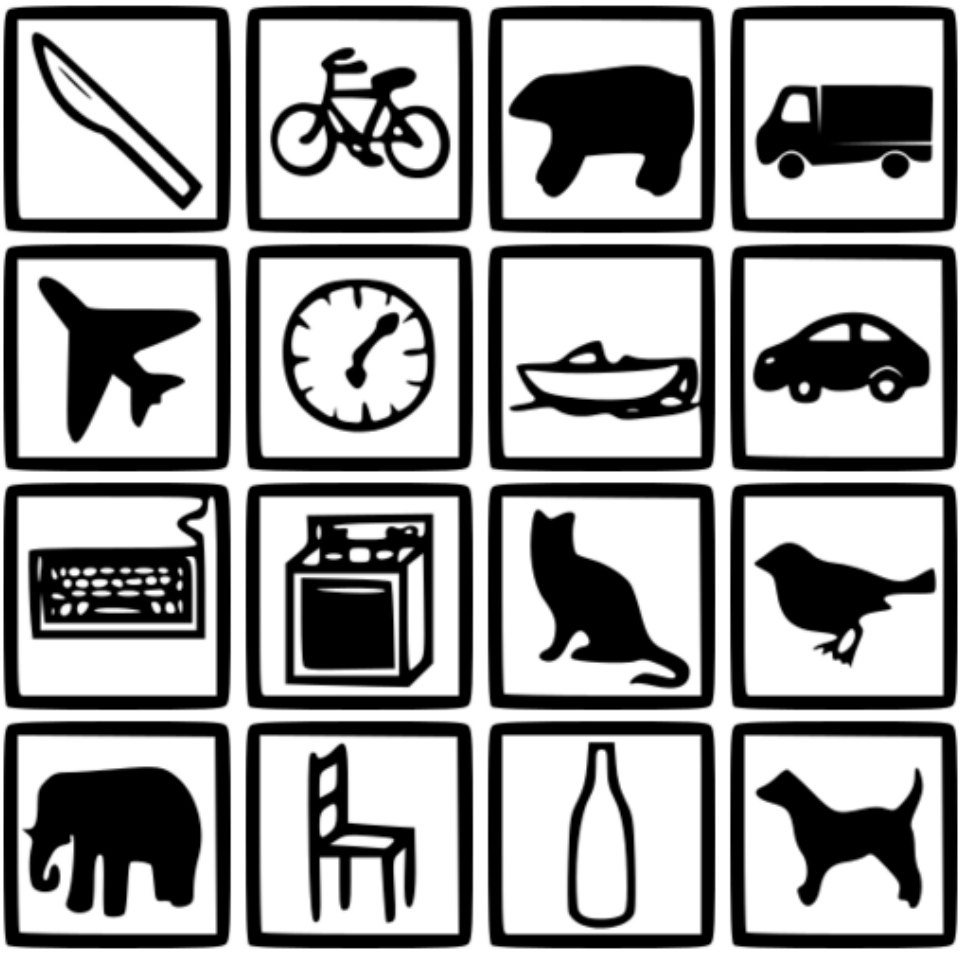


Figure 3.7: Response screen shown to participants after each cue-conflict image, taken from [35]. Possible answer categories are represented by object shapes.

feature was sought after. With this experimental design, participants for instance recognizing the texture of an elephant, would have to click on the shape of an elephant to give this texture-cue-based answer. It is plausible that participants recognized both the texture and shape cues in the images and, when unsure which feature to prioritize, defaulted to a shape-cue prediction, the feature represented by the answering mechanism. This potential bias diminishes the informative value of the reported results indicating a strong shape-based bias in human decisions. Furthermore, it raises questions about the magnitude of the difference between humans and models in feature preference, suggesting that this difference, reported as significant, may be smaller than previously assumed.

Seeing the issues observed with the experimental design, including the cue-conflict evaluation protocol being potentially biased toward texture-cue-based decisions for models, and conversely introducing a bias towards shape-cue decisions for humans, the usefulness of the reported findings that humans and models classify images based on entirely different feature biases comes into question. The analysis highlights how the cue-conflict evaluation protocol may, in general, be unsuitable for evaluating conflicting image features due to the lack of clear

separation between these features in the generated images.

3.4 Common Misinterpretations of Feature-Bias

Besides issues in the cue-conflict experiment, described in the previous Section, the results of Geirhos et al.'s cue-conflict experiment were further misinterpreted in two significant ways:

(1) The results were interpreted by the authors as indication of a general bias of CNNs towards texture features, evident in the title of their work "ImageNet-trained CNNs are biased towards texture; [...]" [1], leading to many subsequent works adopting this simplified perspective on feature bias. However, the cue-conflict experiment itself did not reveal a statistically significant general bias toward texture in CNNs, only a higher texture bias than measured in humans. The possibility that the experimental setup itself may have introduced bias, favoring one feature, was largely overlooked. Instead, the results were interpreted as evidence that models are inherently texture-biased, with the experiment regarded as a neutral evaluation protocol for feature bias. This interpretation could be due to the remarkable reported difference between models and humans, making the experiment favoring texture-cues seem unlikely. However, such a view is shortsighted, given the well-established fact of humans shape-bias [8].

(2) The feature bias toward texture, which was mistakenly assumed to be universally valid, was subsequently interpreted as a direct indication that texture is the image feature most relevant for a model's object classification performance. However, this interpretation may not necessarily hold true. The feature bias reported by Geirhos et al. merely reflects how models classify when features are simultaneously abundant, potentially with significant differences in the strength of their signals and conflicting, e.g. drawn from multiple classes. It does not provide direct insights into the relative importance of each individual feature for correct classification. For instance, it is entirely possible that texture features exhibit only higher signal strength but are less predictive than shape features. As demonstrated by Hermann and Lampinen [26], simpler but less predictive features can suppress more complex yet highly predictive features in a model's decision-making process. This suggests that a reported bias toward texture may not necessarily indicate its greater importance for classification performance, but rather its dominance due to stronger signal strength. This led to further possible misinterpretations, such as that consequently, shape features were deemed less informative, as they were evaluated only in a binary, either-or framework relative to texture. However, this interpretation neglects the complexity of feature interactions and their varying contributions to classification performance.

We conclude that the reported findings of the cue-conflict experiment not only exhibit potential methodological issues, as discussed in Section 3.3, but were also generously interpreted as evidence of a general texture bias in CNNs. This interpretation was made despite the fact that the bias neutrality of the experiment itself was not explicitly verified. Moreover, both the results of Geirhos et al.'s cue-conflict experiment and their evaluation framework, which described the bias towards texture or shape on a singular scale, contributed significantly to this misunderstanding. This binary perspective of feature bias, treating models as being biased toward one of two conflicting feature preferences, disregards the potential relevance of other image features, such as spectral content. It also overlooks the possibility that models may rely on different combinations of image features depending on the specific circumstances, sometimes emphasizing singular features and other times relying on combinations of features for accurate classification. Nonetheless, this feature bias perspective was broadly interpreted as a

direct indicator of the relevance of image features for model classification performance, leaving these broader questions unanswered.

This thesis seeks to address these gaps by investigating how reliant models trained on datasets of the RS and CV domains are on different image features for classification tasks. In addition, it treats spectral features as independent and significant, providing a more comprehensive evaluation of feature importance in classification tasks. In the next chapter, we describe the proposed novel feature reliance evaluation protocol, designed to improve upon the described limitations of the cue-conflict evaluation protocol by Geirhos et al. through the targeted suppression of individual image feature types, enabling a more precise understanding of feature importance.

4 Feature-Reliance Evaluation Protocol

This chapter introduces the proposed feature reliance evaluation protocol, designed to quantify the relative importance of individual image features types across different datasets.

4.1 Introduction and Underlying Assumptions

The cue-conflict evaluation method proposed by Geirhos et al.[1] focused on evaluating conflicting image features at test time to determine which feature models preferred for their classification decisions. This approach was intended to predict the relevance of different image features by evaluating bias toward one of two options: texture or shape. However, as discussed in Chapter3, this binary perspective is not without its flaws. A preference for one feature might simply indicate that this feature exhibits a stronger signal strength when present, rather than confirming it as the most predictive or relied-upon feature for correct predictions. As demonstrated by Hermann et al. [26], models can learn features that are easier to detect, even when these features are less predictive for the task. In the cue-conflict evaluation setup, as both features are simultaneously present, the evaluation measures only the relative signal strength of the features, not their true predictive relevance. This distinction highlights a key limitation in interpreting the results of such experiments.

To evaluate the reliance of models on different image features, we take a different approach than that of the experiments conducted by Geirhos et al. Specifically, we clearly distinguish local shape and spectral content features from the definition of texture, and do not focus on how models predict between two conflicting classes with features for both present. Instead, our approach evaluates how well models classify objects when one or more image feature categories are intentionally disturbed, rendering them unusable for the classification. This approach addresses the theoretical gap between signal strength and predictive power. By eliminating conflicting classes and focusing solely on the absence or presence of specific features types, varying signal strength of different feature types is no longer an issue. All usable features signal contributes directly to the correct classification. The importance of missing features can therefore be measured directly by their effect on classification performance, removing the confounding effects of varying signal strength. The importance of individual image features can thus be evaluated in two ways: first, by measuring performance reduction when a specific feature type is missing, and second, by measuring remaining performance when only that specific feature type is present, while other features types are obstructed. This method allows us to analyze the reliance of models on combinations of more than one image feature by observing how well single missing feature types are handled across all feature types, providing a more comprehensive understanding of feature interactions and their contributions to classification performance.

To explicitly target image feature reliance depending on different datasets, we need to understand the origins of feature reliance. We hypothesize that the reliance of a deep learning

model on a specific image feature is influenced by a combination of three major contributing factors:

1. The image domain, which determines the usefulness of individual image features. Certain image domains may contain highly predictive information through specific features, whereas in others, the same features might hold less relevance [36, 37]. For instance, spectral content may be crucial in the RS domain for identifying land cover classes [38], but its importance might diminish in natural image datasets dominated by shape or texture information. This variability underscores the dependence of feature relevance on the characteristics of the domain.
2. The training strategy, which significantly influences the predictive capabilities of individual image features. As demonstrated by Hermann et al. [26], the way models are trained can emphasize or suppress the reliance on specific features, depending on factors such as data augmentation, loss functions, or optimization techniques. These strategies can play a role in how a model prioritizes and uses available feature information during classification tasks.
3. The model's architecture, which inherently encodes inductive biases toward specific image features. For example, CNNs, with their convolutional layers, exhibit an inductive bias toward localized pattern information [39, 40], which can them particularly reliant on small-scale localized features.

To evaluate differences of feature reliance in datasets of RS domain, we employ an identical set of model architectures throughout the entire evaluation of our feature reliance protocol, where models were trained using a straightforward training strategy. This ensures that any measured differences in model reliance on image features are directly attributable to differences in the data characteristics of each dataset.

4.2 Feature-Suppressing Transformations

To target specific image feature types, we apply image transformations designed to suppress these features by reducing their usability. These transformations are applied at test time and are categorized into three types, corresponding to the image feature definitions described in Section 3.1:

1. **Shape Suppressing Transformations**: Transformations preserving spectral content and local patterns while splitting continuous edges, disrupting global and local shape features.
2. **Texture Suppressing Transformations**: Transformations smoothing local surface patterns, leaving spectral content and edges largely intact, thereby targeting texture features.
3. **Spectral Suppressing Transformations**: Transformations retaining edges and patterns while altering channel information, disrupting spectral features.

Each transformation is controlled by a specific intensity parameter, a non-negative value allowing for a variable intensity of the transformation. For instance, the *patch shuffle* transformation divides images into patches and shuffles them, with the intensity parameter grid size determining the number of patches in width and length into which the image is divided. A parameter value of 0 indicates that the transformation is not applied. All transformations are

applied to normalized image tensors, with a mean of 0 and a standard deviation of 1, ensuring consistency across all datasets and allowing for a controlled suppression of individual feature types. In the following, individual transformations are described.

4.2.1 Channel Shuffle

The first spectral feature suppressing transformation is *channel shuffle*, where for each image, a specified portion of randomly chosen image channels is shuffled. Let $\mathbf{I}, \widetilde{\mathbf{I}} \in \mathbb{R}^{C\times H\times W}$ represent the original image tensor and transformed image tensor respectively with C channels, height H, and width W. The transformation intensity parameter $p \in [0,1]$ denotes the fraction of channels to shuffle, with $\alpha = \lfloor p \cdot C \rfloor$. Alternatively, the number of channels to shuffle p_c can be defined with $\alpha = min(p_c, c)$. Randomly select α channels $S \subseteq \{1,\dots,C\}$, and let $\pi : S \to S$ be a derangement of selected channels (i.e., $\pi(c) \neq c$ for all $c \in S$). Such a selection ensures that every selected image channel is displaced, not allowing for random shuffling to keep channels in place by chance. The transformed output tensor $\mathbf{I}'$ is defined as:

$$\mathbf{I}'_{c,h,w} = \begin{cases} \mathbf{I}_{\pi(c),h,w}, & c \in S, \\ \mathbf{I}_{c,h,w}, & c \notin S \end{cases} \tag{4.1}$$

The *channel shuffle* transformation disrupts the model's understanding of spectral content by breaking the consistent mapping of certain classes to specific channel values. By altering the correspondence between channels and their original spectral information, the transformation renders the spectral content unreliable, thus impeding the model's ability to rely on predefined spectral cues for classification.

4.2.2 Channel Inversion

The second spectral feature suppressing transformation is *channel inversion*, where for each image, a share of randomly selected image channels is inverted. Let $\mathbf{I}, \widetilde{\mathbf{I}} \in \mathbb{R}^{C\times H\times W}$ represent the original image tensor and transformed image tensor respectively with C channels, height H, and width W. Let the transformation intensity parameter $p \in [0,1]$ denote the fraction of channels to invert, with $\alpha = \lfloor p \cdot C \rfloor$. Alternatively, the number of channels to shuffle p_c can be defined with $\alpha = min(p_c, c)$. Randomly select α channels $S \subseteq \{1,\dots,C\}$. The transformed tensor $\mathbf{I}'$ is given by:

$$\mathbf{I}'_{c,h,w} = \begin{cases} -\mathbf{I}_{c,h,w}, & c \in S, \\ \mathbf{I}_{c,h,w}, & c \notin S \end{cases} \tag{4.2}$$

This transformation primarily disrupts the model's interpretation of spectral content by rendering spectral values unreliable. It is noteworthy that by altering only a subset of the image channels, the relative contrasts between channels are modified, which could inadvertently affect model's understanding of edge information and patterns. This occurs because edge and pattern recognition often rely on consistent relationships between channels, and partial inversion introduces inconsistencies that may interfere with these processes. Consequently, while this transformation targets spectral content, its additional effect on shape and texture features when inverting only some of the image channels must be considered.

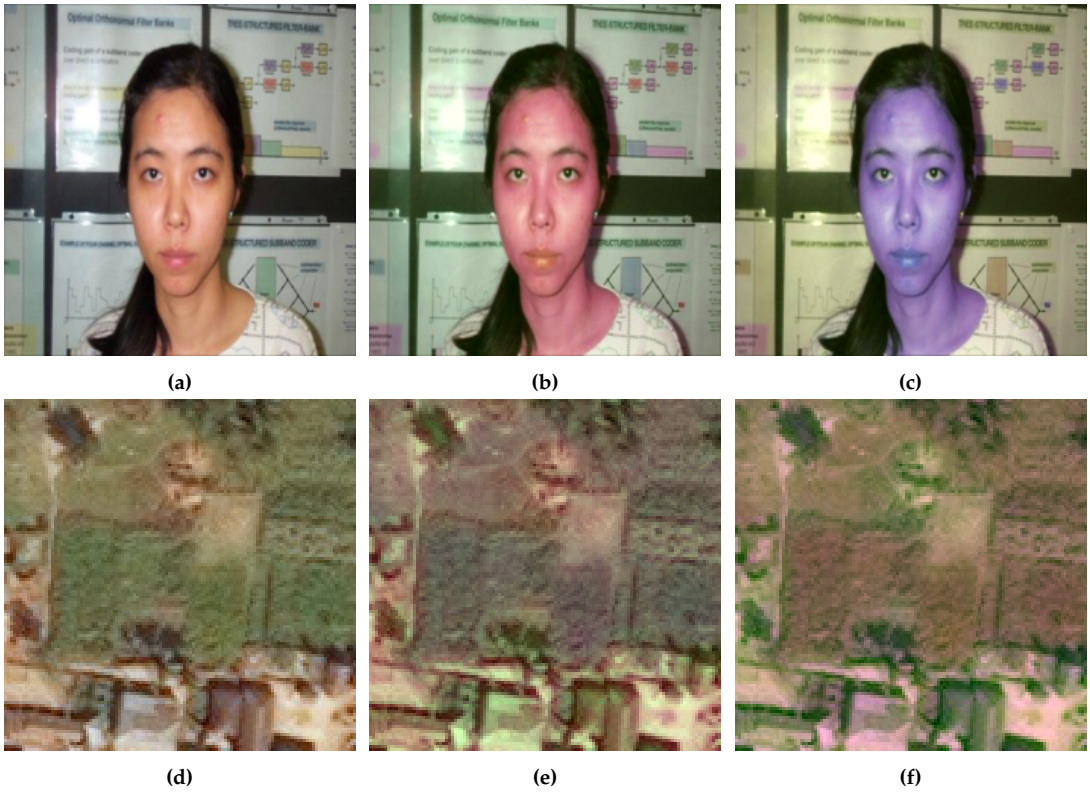


Figure 4.1: Examples of *channel shuffle* transformation applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row, suppressing spectral features. Effect of transformation is shown for different shares of channels shuffled, with **(a,d)** no channels shuffled, **(b,e)** 2/3 channels shuffled and **(c,f)** all channels shuffled .

4.2.3 Channel Mean

The third spectral feature suppressing transformation is *channel mean*, where for each image, all image channels are averaged toward a global mean value. Let $\mathbf{I}, \widetilde{\mathbf{I}} \in \mathbb{R}^{C\times H\times W}$ represent the original image tensor and transformed image tensor respectively with C channels, height H, and width W. Let $\mathbf{M} \in \mathbb{R}^{C\times H\times W}$ be the uniform channel mean defined as:

$$\mathbf{M}_{c,h,w} = \frac{1}{C}\sum_{k=1}^{C} \mathbf{I}_{k,h,w} \tag{4.3}$$

Let the transformation intensity parameter $p \in [0,1]$ denote the fraction by which each channel is replaced by the uniform channel mean. Alternatively, the number of channels to shuffle p_c can be defined with $\alpha = min(p_c, c)$. The transformed output tensor $\mathbf{I}'$ is defined as:

$$\mathbf{I}' = p\,\mathbf{M} + (1-p)\,\mathbf{I} \tag{4.4}$$

As p increases, the spectral information from the original tensor $\mathbf{I}$ is progressively replaced by the uniform average $\mathbf{M}$, reducing the distinctiveness of individual channels and making spectral features less reliable for classification. A value of $p = 1$ indicates that channels are

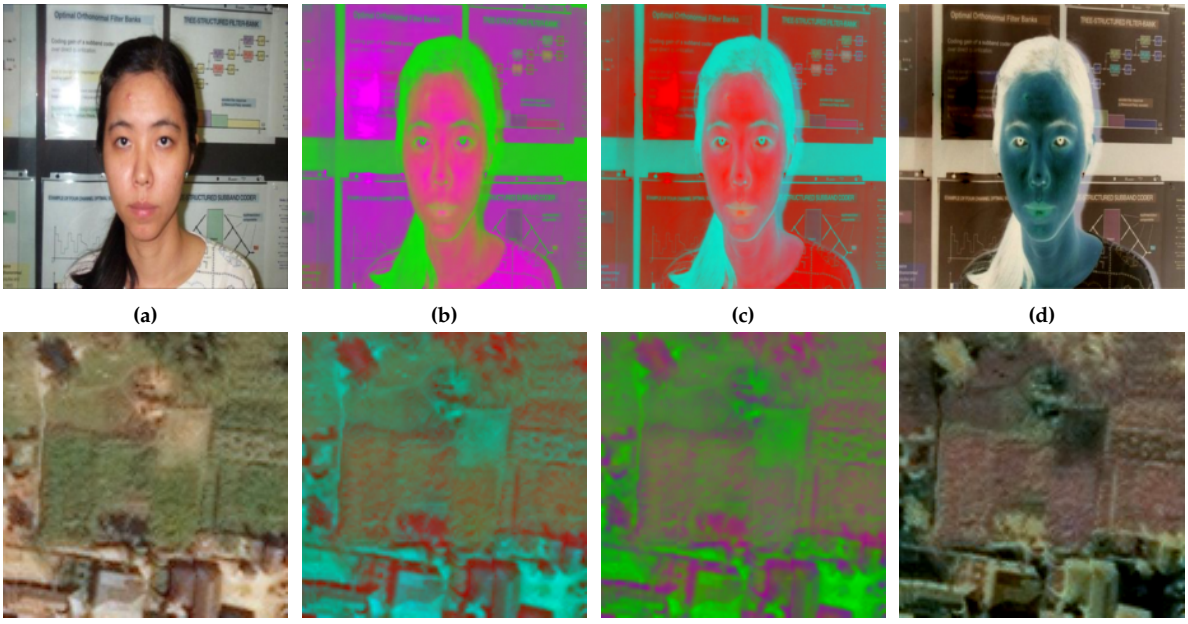


Figure 4.2: Examples of *channel inversion* transformation applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row, suppressing spectral features. Effect of transformation is shown for different shares of channels inverted, with **(a,e)** no channels inverted, **(b,f)** 1/3 channels inverted, **(c,g)** 2/3 channels inverted and **(d,h)** all channels inverted.

entirely replaced by their mean. This transformation disrupts spectral features in a way so that original spectral distinctions are progressively lost, making their information content less reliable for classification.

4.2.4 Median Filter

The first texture feature suppressing transformation is *median filter*, where for each image, a median filtering kernel is applied. Let $\mathbf{I}, \widetilde{\mathbf{I}} \in \mathbb{R}^{C \times H \times W}$ represent the original image tensor and transformed image tensor respectively and transformed with C channels, height H, and width W. Let $I(c,h,w)$ denote the pixel value in channel c and spatial coordinates (h,y) in the input image. Let k be defined as $k = 2n+1$ where $n \in \mathbb{N}$, representing the kernel size and specifying a $k \times k$ neighborhood centered at (h,w). Values of the transformed output tensor $\widetilde{\mathbf{I}}$ are defined as:

$$\widetilde{\mathbf{I}}(c,h,w) = \text{median}\left\{ I\left(c, h+\Delta\widetilde{h}, w+\Delta\widetilde{w}\right) \mid -\left\lfloor \frac{k}{2} \right\rfloor \leq \Delta h, \Delta w \leq \left\lfloor \frac{k}{2} \right\rfloor \right\} \tag{4.5}$$

Where relative offset terms $\Delta\widetilde{h}$ and $\Delta\widetilde{w}$ are defined according to the mirror padding strategy, to ensure that no static spectral values are introduced during the filtering process:

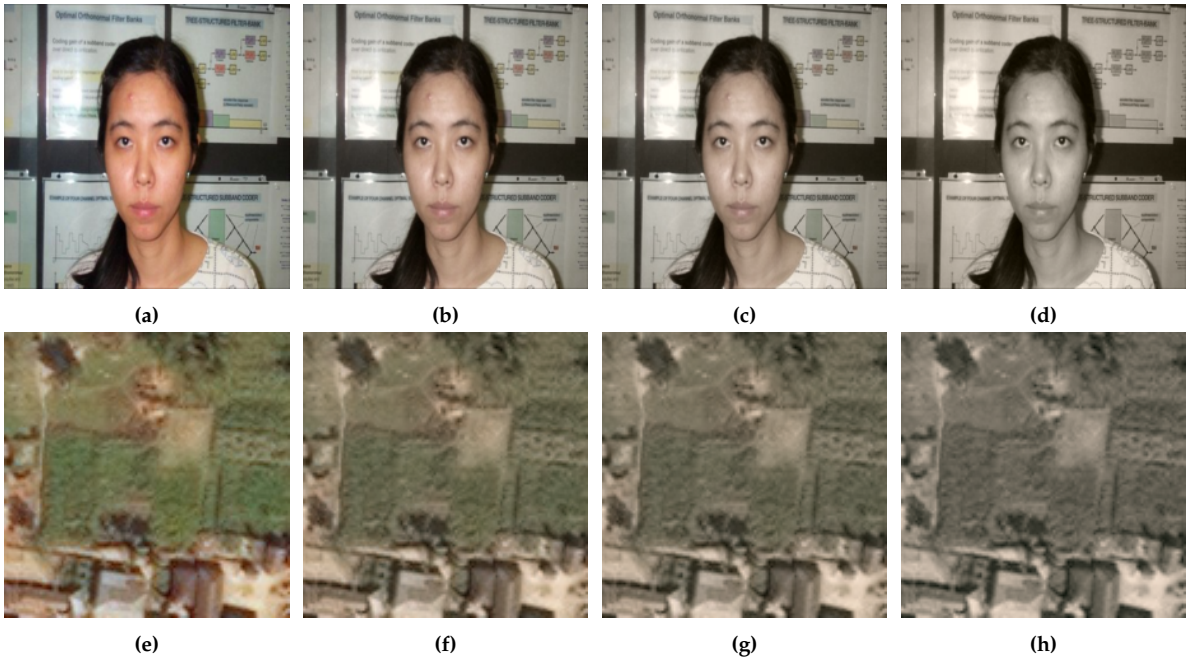


Figure 4.3: Examples of *channel mean* transformation applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row, suppressing spectral features. Effect of transformation is shown for different values of intensity parameter p, with **(a,e)** no channels averaging, **(b,f)** channels mean replacing channels by 50%, **(c,g)** channels mean replacing channels by 70% and **(d,h)** all channels replaced by their mean.

$$\Delta\widetilde{h} = \begin{cases} -\Delta h - 1, & h + \Delta h \leq 0 \\ \Delta h, & 0 < h + \Delta h \leq H \\ 1 - \Delta h, & h + \Delta h > H \end{cases} \tag{4.6}$$

$$\Delta\widetilde{w} = \begin{cases} -\Delta w - 1, & w + \Delta w \leq 0 \\ \Delta w, & 0 < w + \Delta w \leq W \\ 1 - \Delta w, & w + \Delta w > W \end{cases} \tag{4.7}$$

The *median filter* transformation smooths the image by replacing each pixel value with the median value of the surrounding pixels within the kernel. This effectively removes small-scale repetitive patterns, such as noise or fine texture details, while preserving the sharpness of edges, similar to the behavior of the *bilateral filter*. The transformation intensity parameter is the *kernel size* which defines the dimensions of the filter window. Larger kernel sizes result in greater smoothing intensity, as the filter considers a larger area for computing the median. Notably, unique small-scale features, such as local edges, may also be affected by the filter. This can potentially alter some shape information, particularly for smaller or finer structures in the image.

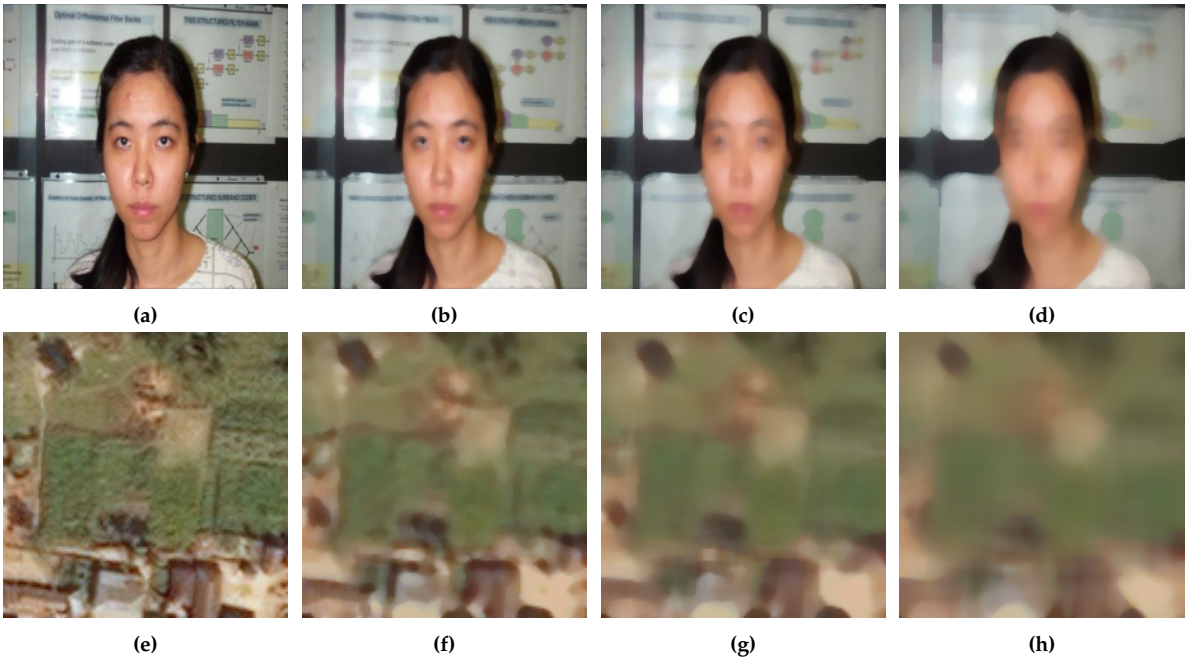


Figure 4.4: Examples of *median filter* transformation applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row, suppressing texture features. Effect of transformation is shown for different kernel sizes, with **(a,e)** no median filtering applied, **(b,f)** median filtering with a kernel size of 5, **(c,g)** median filtering with a kernel size of 9 and **(d,h)** median filtering with a kernel size of 15.

4.2.5 Gaussian Filter

The second texture feature suppressing transformation is *gaussian filter*. A Gaussian Filter is applied to all images, smoothing the image by reducing repetitive patterns and softening edges. This transformation operates by convolving the image with a Gaussian kernel, which applies a weighted average to the values of the surrounding pixels.This can be expressed as follows: Let $\mathbf{I}, \widetilde{\mathbf{I}} \in \mathbb{R}^{C\times H\times W}$ represent the original image tensor and transformed image tensor respectively with C channels, height H, and width W. Let $I(c,h,w)$ denote the pixel value at channel c and spatial coordinates (h,w) in the input image and σ the standard deviation of the Gaussian distribution. Values of the transformed output tensor $\widetilde{\mathbf{I}}$ are defined as:

$$\widetilde{\mathbf{I}}(c,h,w) = \sum_{\Delta h=-r}^{r} \sum_{\Delta w=-r}^{r} G(\Delta h, \Delta w) \cdot I\left(c, h+\Delta\widetilde{h}, w+\Delta\widetilde{w}\right) \tag{4.8}$$

where $r = \lceil 3\sigma \rceil$ defines the kernel radius, and $G(\Delta h, \Delta w)$ denotes the Gaussian kernel weight defined as:

$$G(\Delta h, \Delta w) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\Delta h^2 + \Delta w^2}{2\sigma^2}\right) \tag{4.9}$$

Relative offset terms $\Delta\widetilde{h}$ and $\Delta\widetilde{w}$ are defined according to the mirror padding strategy to ensure that no static spectral values are introduced during the filtering process:

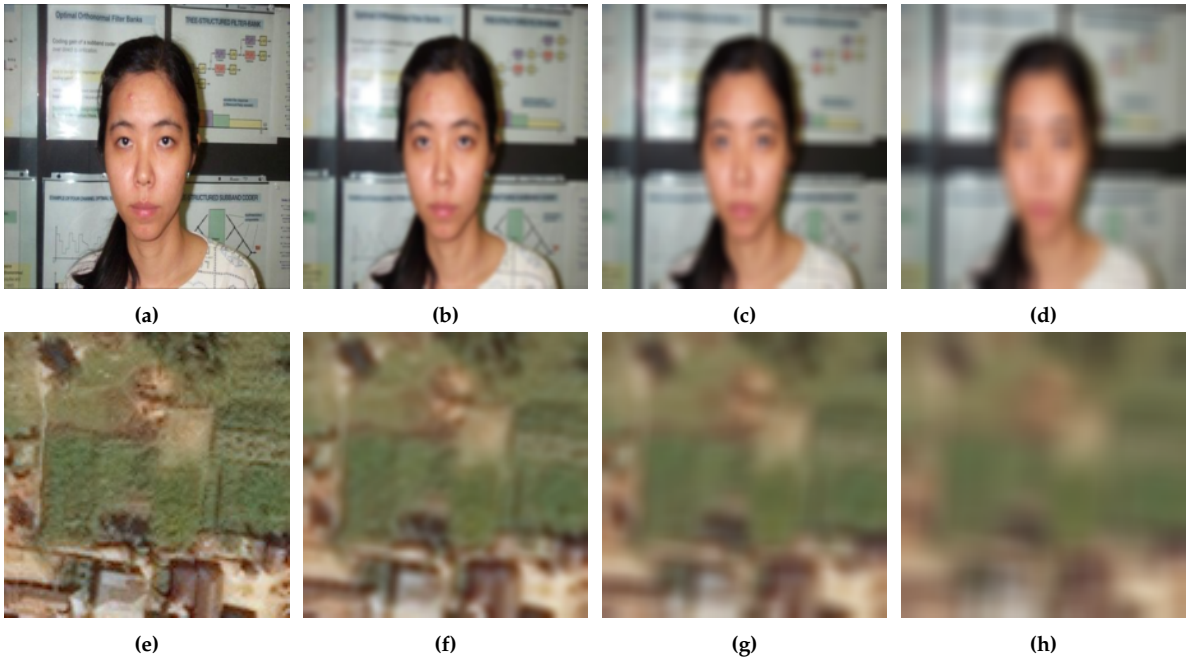


Figure 4.5: Examples of *gaussian filter* transformation applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row, suppressing texture features. Effect of transformation is shown for different σ values, with **(a,e)** no *gaussian filter* applied, **(b,f)** *gaussian filter* with a σ value of 2, **(c,g)** *gaussian filter* with a σ value of 4 and **(d,h)** *gaussian filter* with a σ value of 7.

$$\Delta\widetilde{h} = \begin{cases} -\Delta h - 1, & h + \Delta h \leq 0 \\ \Delta h, & 0 < h + \Delta h \leq H \\ 1 - \Delta h, & h + \Delta h > H \end{cases} \tag{4.10}$$

$$\Delta\widetilde{w} = \begin{cases} -\Delta w - 1, & w + \Delta w \leq 0 \\ \Delta w, & 0 < w + \Delta w \leq W \\ 1 - \Delta w, & w + \Delta w > W \end{cases} \tag{4.11}$$

The transformation intensity parameter is the standard deviation (σ) of the Gaussian distribution which controls the extent of smoothing. As σ increases, the filter applies broader smoothing, affecting a larger area of the image and resulting in more pronounced blurring. Notably, this transformation, while primarily targeting texture features by smoothing patterns, also has a parallel effect on edge information. Edges are inherently defined by high-frequency content in the image, which is attenuated by the Gaussian Filter. As σ increases, the edges become progressively blurred, reducing the sharpness of edge transitions and potentially affecting the representation of shape features in the image.

4.2.6 Bilateral Filter

The third texture feature suppressing transformation is *bilateral filter*, where to each image a bilateral filtering kernel [41] is applied. This filter smooths regions of similar spectral content

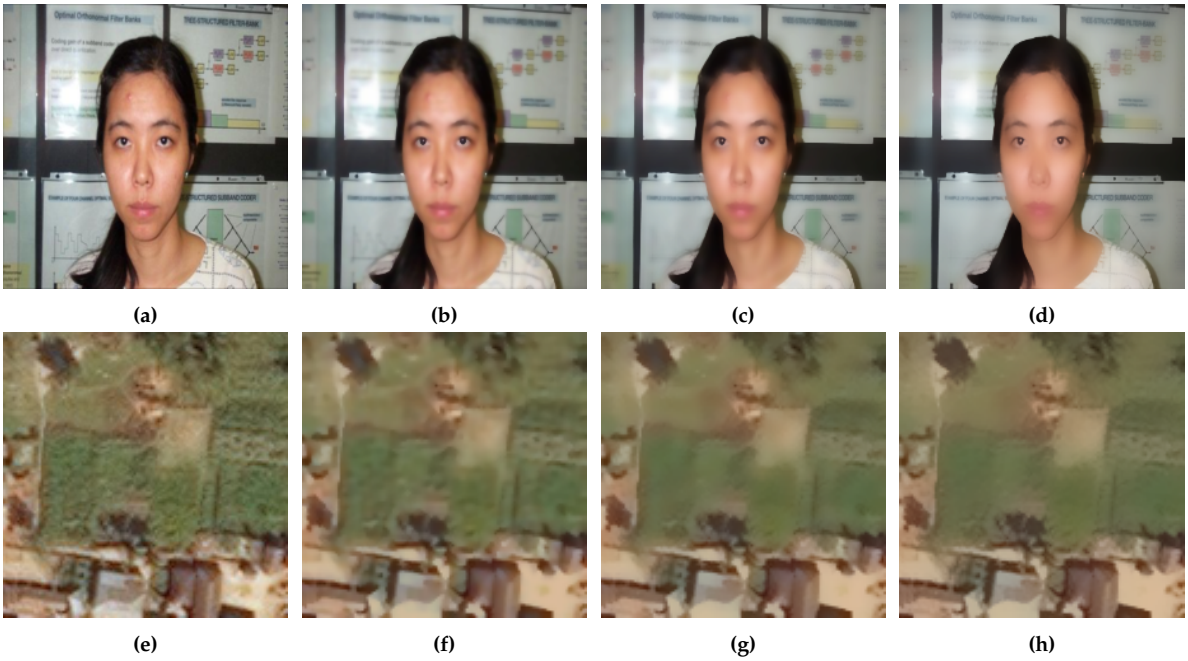


Figure 4.6: Examples of *bilateral filter* transformation applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row, suppressing texture features. Effect of transformation is shown for different kernel diameters, with **(a,e)** no transformation applied, **(b,f)** *bilateral filter* with a kernel diameter of 5, **(c,g)** *bilateral filter* with a kernel diameter of 9, and **(d,h)** *bilateral filter* with a kernel diameter of 15.

by making them uniform, effectively removing small-scale repetitive patterns in the image. In contrast, edges are preserved due to the filter's edge-aware design. Let $\mathbf{I}, \widetilde{\mathbf{I}} \in \mathbb{R}^{C\times H\times W}$ represent the original image tensor and transformed image tensor respectively with C channels, height H, and width W. Let $I(c,h,w)$ denote the pixel value at channel c and spatial coordinates (h,w) in the input image. Define σ_s and σ_r as the spatial and range standard deviations, respectively, controlling the extent of the filter in the spatial and intensity domains. Values of the transformed output tensor $\widetilde{\mathbf{I}}$ are defined as:

$$\begin{aligned}\widetilde{\mathbf{I}}(c,h,w) = \frac{1}{W(h,w)} \sum_{\Delta h=-r}^{r} \sum_{\Delta w=-r}^{r} & G_s(\Delta h, \Delta w) \\ & \cdot G_r\left(I(c,h,w), I\left(c, h+\Delta\widetilde{h}, w+\Delta\widetilde{w}\right)\right) \\ & \cdot I\left(c, h+\Delta\widetilde{h}, w+\Delta\widetilde{w}\right)\end{aligned} \tag{4.12}$$

where d denotes the kernel diameter, $r = \frac{d-1}{2}$ the kernel radius, and W the normalization factor with:

$$W(h,w) = \sum_{\Delta h=-r}^{r} \sum_{\Delta w=-r}^{r} G_s(\Delta h, \Delta w) \cdot G_r\left(I(c,h,w), I\left(c, h+\Delta\widetilde{h}, w+\Delta\widetilde{w}\right)\right) \tag{4.13}$$

The spatial Gaussian kernel G_s and range Gaussian kernel G_r are defined as:

$$G_s(\Delta h, \Delta w) = \exp\left(-\frac{\Delta h^2 + \Delta w^2}{2\sigma_s^2}\right) \tag{4.14}$$

$$G_r\left(I(c,h,w), I\left(c, h+\Delta\widetilde{h}, w+\Delta\widetilde{w}\right)\right) = \exp\left(-\frac{\left(I(c,h,w) - I\left(c, h+\Delta\widetilde{h}, w+\Delta\widetilde{w}\right)\right)^2}{2\sigma_r^2}\right) \tag{4.15}$$

Relative offset terms $\Delta\widetilde{h}$ and $\Delta\widetilde{w}$ are defined according to the mirror padding strategy to ensure that no static spectral values are introduced during the filtering process:

$$\Delta\widetilde{h} = \begin{cases} -\Delta h - 1, & h + \Delta h \leq 0 \\ \Delta h, & 0 < h + \Delta h \leq H \\ 1 - \Delta h, & h + \Delta h > H \end{cases} \tag{4.16}$$

$$\Delta\widetilde{w} = \begin{cases} -\Delta w - 1, & w + \Delta w \leq 0 \\ \Delta w, & 0 < w + \Delta w \leq W \\ 1 - \Delta w, & w + \Delta w > W \end{cases} \tag{4.17}$$

The transformation intensity parameter is the kernel diameter parameter d which controls the size of the neighborhood considered for smoothing. As the diameter parameter increases, the filter takes into account a larger surrounding area for averaging, resulting in broader smoothing effects. In contrast, smaller values restrict the smoothing to more localized regions, maintaining finer details. Among the proposed texture-suppressing smoothing transformations, *bilateral filter* preserves shape information conveyed through edges most effectively. As a result, it appears to be best qualified for targeting texture image features without introducing secondary effects on shape image features.

4.2.7 Patch Shuffle

The first shape feature suppressing transformation is *patch shuffle*, where every image is divided into a grid of patches of equal width and length and each patch an individual grid position is assigned. Let $\mathbf{I}, \widetilde{\mathbf{I}} \in \mathbb{R}^{C\times H\times W}$ represent the original image tensor and transformed image tensor, respectively, with C channels, height H and width W. Let g denote the grid size parameter, by which the image is divided into $g \times g$ patches, and $G : \{1, \ldots, g\}$ the collection of possible patch indexes. Let $p_h = \frac{H}{G}$ and $p_w = \frac{W}{G}$ be the patch height and patch width. Let $g_i : \{1, \ldots, H\} \to G$ and $g_j : \{1, \ldots, W\} \to G$ be the grid row and grid column index mapping functions and $h_{local} : \{1, \ldots, H\} \to \{1, \ldots, p_h\}$, $w_{local} : \{1, \ldots, W\} \to \{1, \ldots, p_w\}$ be the patch-specific height and width mappings for any given h, w with:

$$g_i(h) = \left\lfloor \frac{h}{p_h} \right\rfloor, \quad g_j(w) = \left\lfloor \frac{w}{p_w} \right\rfloor \tag{4.18}$$

$$h_{local}(h) = h - g_i(h) \times p_h \tag{4.19}$$

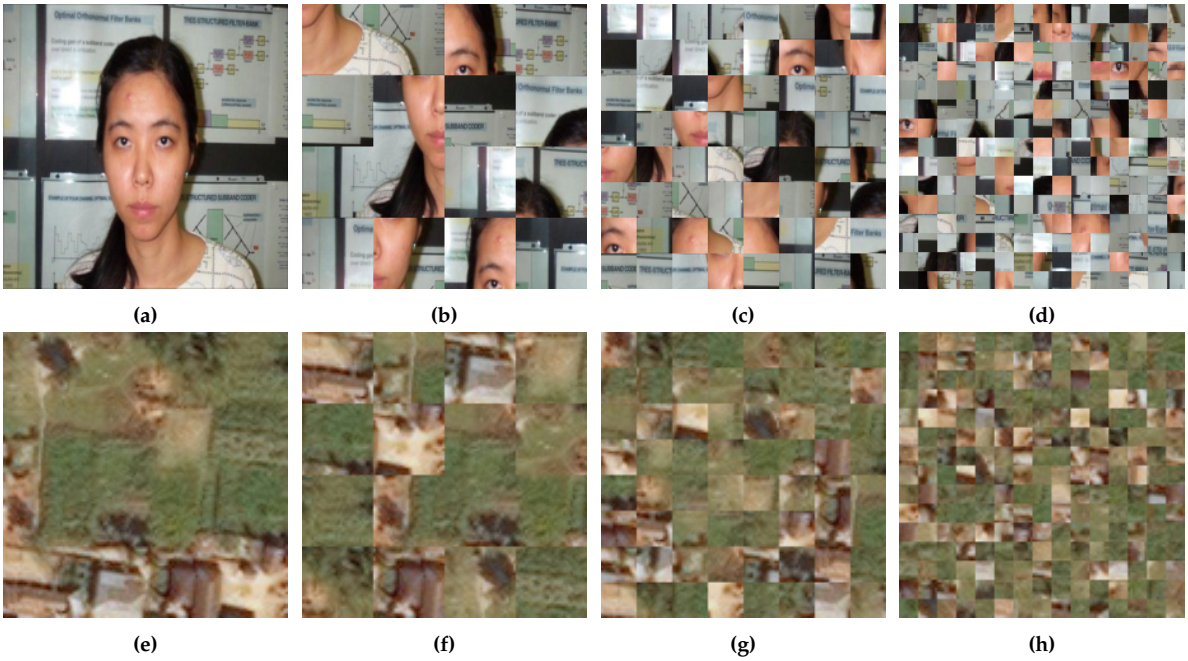


Figure 4.7: Examples of *patch shuffle* transformation applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row, suppressing shape features. Effect of transformation is shown for different grid sizes, with **(a,e)** no *patch shuffle* applied, **(b,f)** *patch shuffle* with a grid sizes of 5, **(c,g)** *patch shuffle* with a grid sizes of 9, **(d,h)** and *patch shuffle* with a grid sizes of 15.

$$w_{local}(w) = w - g_j(w) \times p_w \tag{4.20}$$

Let $\Theta_i : G \times G \to G$ and $\Theta_j : G \times G \to G$ be a derangement mappings of patch coordinates, taking as input a patch coordinate $(i, j) \in G \times G$ and output the randomly selected new patch coordinate:

$$\Theta_i(i, j) = i', \quad \Theta_j(i, j) = j', \tag{4.21}$$

where $i', j' \in G$ and $(i', j') \sim \mathcal{U}(G)$. Let $(\widetilde{h}, \widetilde{w})$ be the corresponding spatial coordinates for each pair (h, w) after transformation with

$$\widetilde{h} = \Theta_i(g_i(h), g_j(w)) \times p_h + h_{local}(h) \tag{4.22}$$

$$\widetilde{w} = \Theta_j(g_i(h), g_j(w)) \times p_w + w_{local}(w) \tag{4.23}$$

Finally, values of the transformed output tensor $\widetilde{\mathbf{I}}$ are defined as:

$$\widetilde{\mathbf{I}}(c, h, w) = \mathbf{I}(c, \widetilde{h}, \widetilde{w}) \tag{4.24}$$

This transformation disrupts edge information that spans across patch borders, reducing its contribution to the shape image feature. Thereby, even edges that span smaller sizes than that of one patch have a chance of being affected due to the random positioning of the grid. As a result, the continuity and coherence of shape-related information are significantly diminished. At the same time, smaller scale patterns representing texture features still remain present in

the image, even when repositioned. The transformation intensity parameter is the grid size which specifies the number of rows and columns into which the image is divided. Images not fully divisible in size by the grid size are upscaled before the transformation and rescaled to original size after. A higher grid size parameter results in more patches of smaller size, thereby affecting a greater number of edges. This not only suppresses global shape features, as in previous applications of this methods [20, 2], but importantly also disrupts local shape features, reducing their usability for classification. Example images of the *patch shuffle* transformation applied can be seen in Figure 4.7

4.2.8 Patch Rotation

The second shape feature suppressing transformation is *patch rotation*, dividing images into grids of patches of equal width and length and rotating each patch. Let $\mathbf{I}, \widetilde{\mathbf{I}} \in \mathbb{R}^{C \times H \times W}$ represent the original image tensor and transformed image tensor respectively with C channels, height H, and width W, g denote the grid size parameter, indicating that the image is divided into $g \times g$ patches, and $G : \{1, \ldots, g\}$ the collection of possible patch indexes. Let $p_h = \frac{H}{G}$ and $p_w = \frac{W}{G}$ be the patch height and patch width. Let $g_i : \{1, \ldots, H\} \to G$ and $g_j : \{1, \ldots, W\} \to G$ be the grid row and grid column index mapping functions and $h_{local} : \{1, \ldots, H\} \to \{1, \ldots, p_h\}$, $w_{local} : \{1, \ldots, W\} \to \{1, \ldots, p_w\}$ be the patch-specific height and width mappings for any

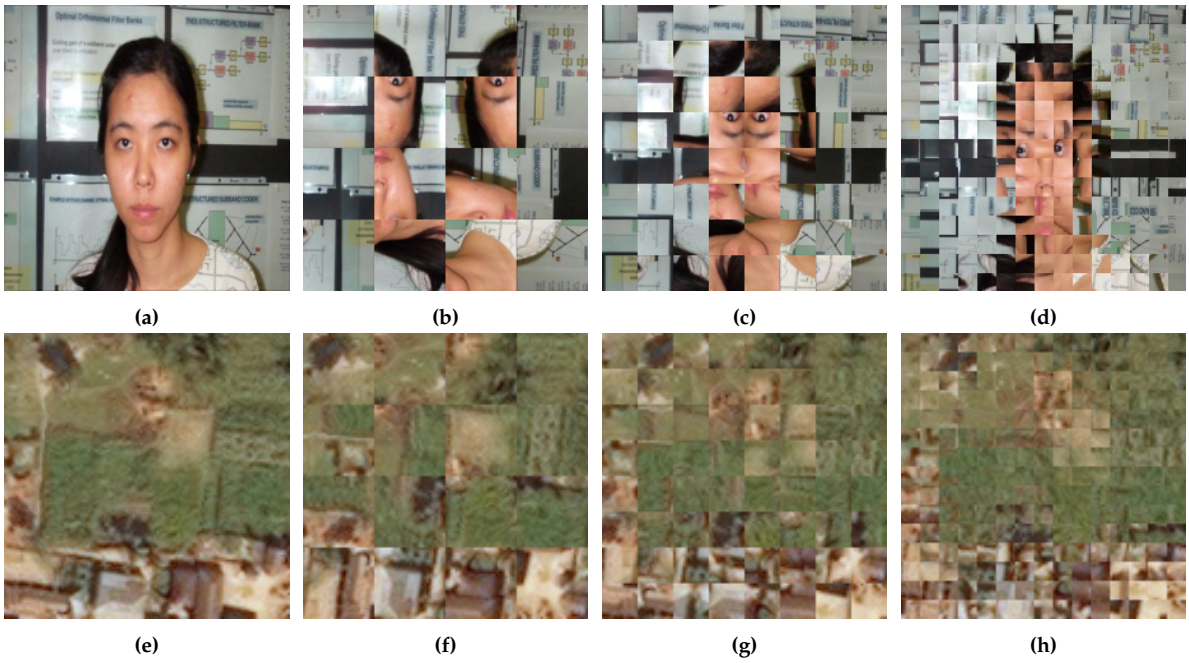


Figure 4.8: Examples of applied *patch rotation* transformation applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row, suppressing shape features. Effect of transformation is shown for different grid sizes, with **(a,e)** no *patch rotation* applied, **(b,f)** *patch rotation* with a grid sizes of 5, **(c,g)** *patch rotation* with a grid sizes of 9, and **(d,h)** *patch rotation* with a grid sizes of 15.

given h, w with:

$$g_i(h) = \left\lfloor \frac{h}{p_h} \right\rfloor, \quad g_j(w) = \left\lfloor \frac{w}{p_w} \right\rfloor \tag{4.25}$$

$$h_{local}(h) = h - g_i(h) \times p_h \tag{4.26}$$

$$w_{local}(w) = w - g_j(w) \times p_w \tag{4.27}$$

Let $R : \{90°, 180°, 270°\}$ be the collection of possible rotation values ensuring a rotation and $\Theta : G \times G \to R$ be the random selection of a rotation value for each patch, with:

$$\Theta(i, j) = r) \tag{4.28}$$

where $r \sim \mathcal{U}(R)$.

Let $\widetilde{h}_{local}(h, w)$ and $\widetilde{w}_{local}(h, w)$ be the mapping functions to local height and width within a patch after it's rotation, with:

$$\widetilde{h}_{local}(h, w) = \begin{cases} w_{local}(w), & \Theta(g_i(h), g_j(w)) = 90° \\ p_h + 1 - h_{local}(h), & \Theta(g_i(h), g_j(w)) = 180° \\ p_w + 1 - w_{local}(w), & \Theta(g_i(h), g_j(w)) = 270° \end{cases} \tag{4.29}$$

$$\widetilde{w}_{local}(h, w) = \begin{cases} p_h - 1 - h_{local}(h), & \Theta(g_i(h), g_j(w)) = 90° \\ p_w - 1 - w_{local}(w), & \Theta(g_i(h), g_j(w)) = 180° \\ h_{local}, & \Theta(g_i(h), g_j(w)) = 270° \end{cases} \tag{4.30}$$

Let $\widetilde{h}(h, w)$ and $\widetilde{w}(h, w)$ be the mapping functions mapping the rotated coordinates with

$$\widetilde{h} = g_i(h) \times p_h + \widetilde{h}_{local}(h, w) \tag{4.31}$$

$$\widetilde{w} = g_j(h) \times p_w + \widetilde{w}_{local}(h, w) \tag{4.32}$$

Finally, values of the transformed output tensor $\widetilde{\mathbf{I}}$ are defined as:

$$\widetilde{\mathbf{I}}(c, h, w) = \mathbf{I}(c, \widetilde{h}, \widetilde{w}) \tag{4.33}$$

By that, similar to *patch shuffle*, edge information stretching over the size of more than a patch is getting divided, suppressing their usability as shape features. The transformation intensity parameter is the grid size specifying the number of rows and columns into which the image is divided. Images not fully divisible in size by the grid size are upscaled before the transformation and rescaled to original size after. Example images of the *patch rotation* transformation applied can be seen in Figure 4.8

However, in contrast to the *patch shuffle* transformation, which distinctly destroyed object silhouettes by changing the patches position inside the image, *patch rotation* does not. The general content of each patch, even though rotated, remains in relative location to each other, whereby the global silhouette shape information of the object is not entirely destroyed. This effect is more pronounced at higher grid sizes, representing a smaller pixel-wise displacement than when larger patches are rotated for smaller grid sizes.

4.2.9 Examples for transformation pairs

As all techniques are image transformations, they can be applied consecutively. This allows for the suppression of not just one but two of the three image feature types as defined in 3.1, thus leaving a single image feature type unaffected. Spectral features remaining can be observed in Figure 4.9, where the *bilateral filter* and *patch shuffle* transformation suppress texture and shape features. Texture features that remain unaffected are shown in Figure 4.10, where the *patch shuffle* and *channel shuffle* transformation impair spectral and shape features. Shape features remaining after the transformations can be seen in Figure 4.11, where the *bilateral filter* and *channel shuffle* transformation reduce usability of spectral and texture features.

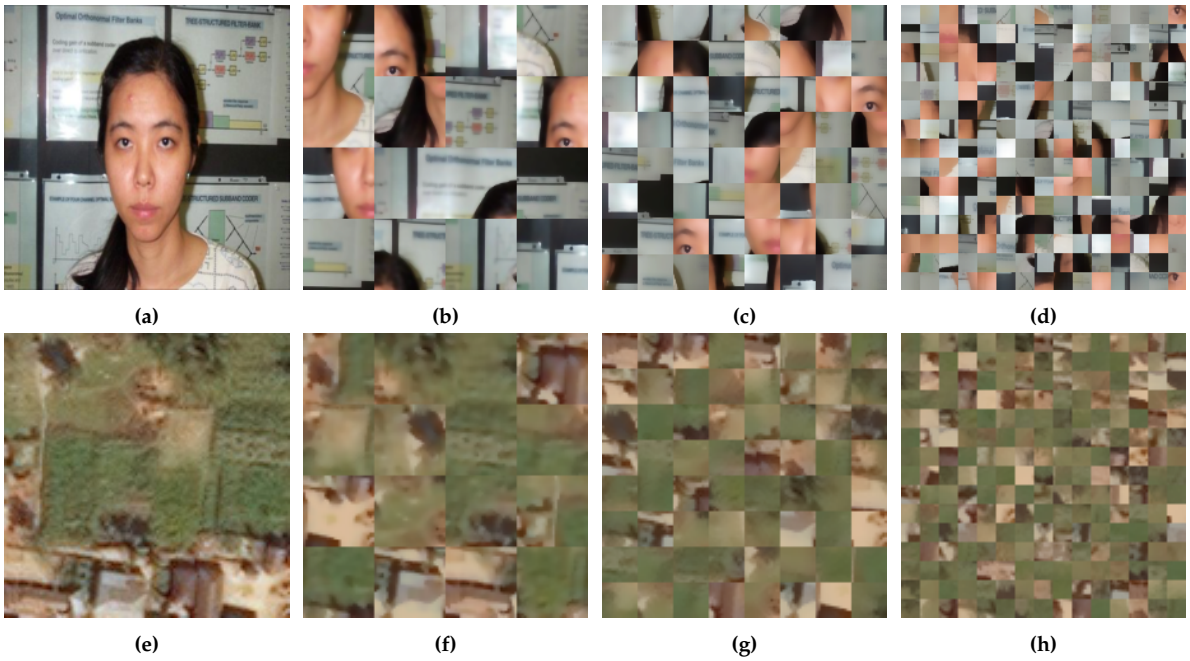


Figure 4.9: Examples of spectral features remaining with *bilateral filter* and *patch rotation* transformations applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row. Effect of transformation is shown for different kernel diameters and grid sizes, with **(a,e)** no transformations applied, **(b,f)** *bilateral filter* with kernel diameter of 5 and *patch shuffle* with a grid size of 4, **(c,g)** *bilateral filter* with kernel diameter of 9 and *patch shuffle* with a grid size of 8, and **(d,h)** *bilateral filter* with kernel diameter of 15 and *patch shuffle* with a grid size of 15.

4.3 Proposed Evaluation Protocol

This section describes the detailed evaluation process of the proposed feature reliance evaluation protocol. Concretely, the evaluation protocol operates by testing a combination of previously trained models, each trained on a specific dataset. This approach ensures that the protocol is independently applicable, regardless of the models or datasets selected. Initially, the test performance of the models is evaluated on the original test set of the dataset they were trained upon. Following this, the trained models are tested on the same test set multiple times, with a different feature-suppressing transformation, or pairs of such transformations applied

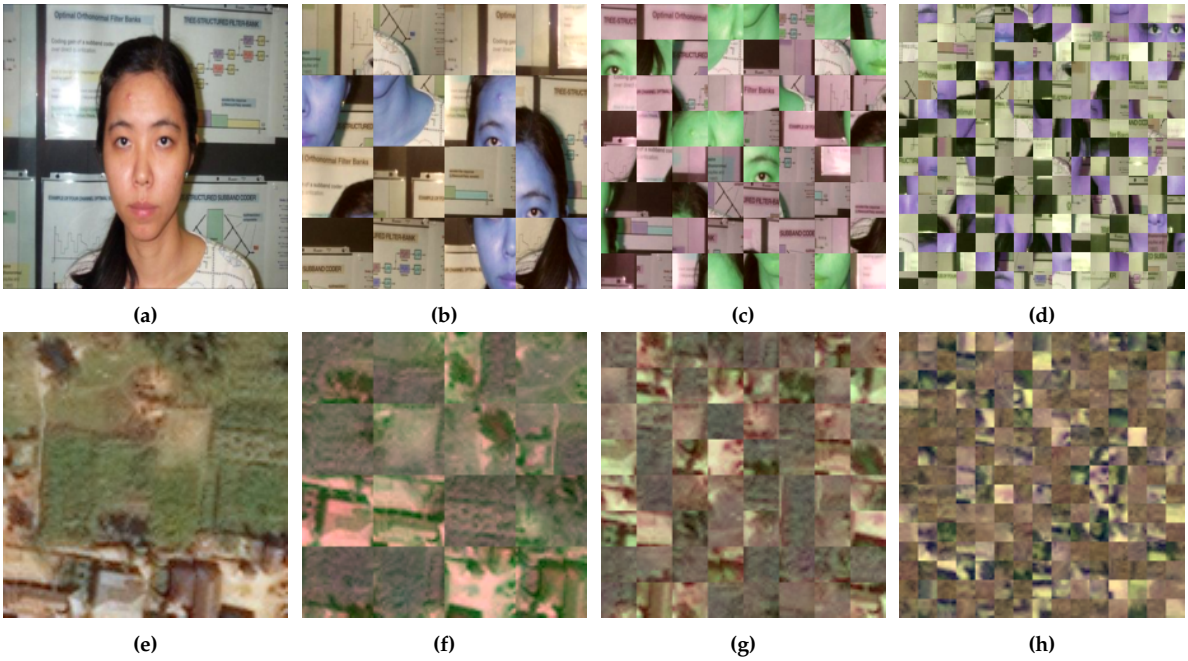


Figure 4.10: Examples of texture features remaining with *patch shuffle* and *patch rotation* transformations applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row. Effect of transformation is shown for different grid sizes, with **(a,e)** no transformations applied, **(b,f)** *patch shuffle* with grid size of 4 and *channel shuffle* with a share of 2/3, **(c,g)** *patch shuffle* with grid size of 8 and *channel shuffle* with a share of 1, and **(d,h)** *patch shuffle* with grid size of 15 and *channel shuffle* with a share of 1.

at each iteration, as described in Section 4.2. These transformations, only applied at test time, reduce the usable information conveyed through specific image features by selectively disturbing them. Thereby, the evaluation of model reliance on both individual image features or combinations of image features is realized on real-world datasets.

To evaluate the reliance on a certain feature type of models trained on various datasets, a single transformation is applied that suppresses the corresponding feature category. To evaluate the singular predictive strength of a feature type for models on a dataset, a pair of transformations is applied, where each transformation suppresses one of the other feature categories. For example, to evaluate the singular predictive strength of shape features, a texture-suppressing transformation and a spectral content-suppressing transformation are applied in conjunction. This ensures that both other competing feature categories are suppressed, leaving only the shape features intact for evaluation.

Model performances are measured under both scenarios: applying each of the eight different single transformations individually and applying pairs of transformations, across varying transformation intensities. To evaluate general trends of feature reliance for a specific dataset, model performances can be averaged. This approach provides insights into how the overall majority of models behave and rely on image features within the dataset. By averaging performances, model-specific biases toward certain feature types are excluded from the evaluation, allowing for a more generalized understanding of feature reliance patterns for datasets as a whole.

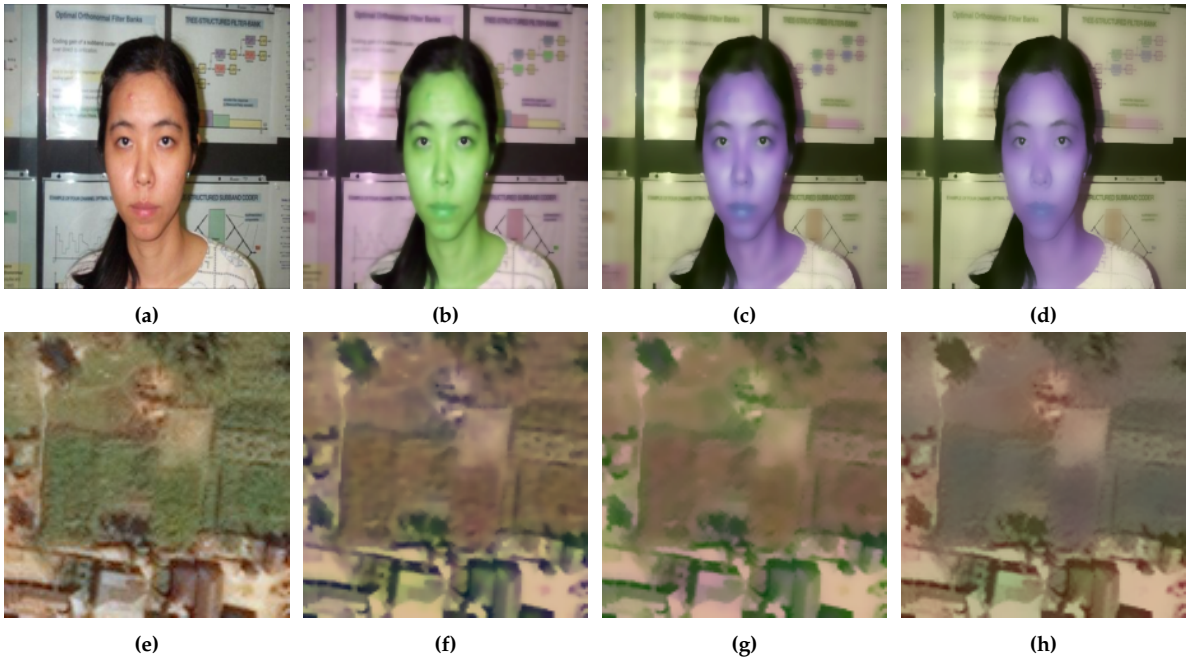


Figure 4.11: Examples of shape features remaining with *bilateral filter* and *channel shuffle* transformations applied to images of the Caltech shown in first row, and DeepGlobe dataset shown in second row. Effect of transformation is shown for different grid sizes, with **(a,e)** no transformations applied, **(b,f)** *bilateral filter* with kernel diameter of 5 and *channel shuffle* with a share of 2/3, **(c,g)** *bilateral filter* with kernel diameter of 9 and *channel shuffle* with a share of 1, and **(d,h)** *bilateral filter* with kernel diameter of 15 and *channel shuffle* with a share of 1.

The test performance measurements with certain feature types suppressed are compared to the original test performance of the model, obtained when no transformations were applied. In this way, a performance score is determined, which is calculated relative to the optimal baseline of test performance with all image features present. This provides a direct metric to assess the impact of a given transformation, and consequently the missing feature information, on the overall classification performance. By normalizing the performance in this manner, the metric becomes independent of differences in absolute classification performances across different datasets, allowing for consistent evaluation across diverse data sets.

Additionally, as classification performance scores can vary due to differences in dataset properties, random guessing can achieve different baseline scores depending on the dataset. This is particularly evident for multilabel classification datasets, where the likelihood of correct predictions by chance differs based on class distributions and label characteristics. To address this, we determined an equally measured lower bound for all datasets by testing model performance on completely noisy images while keeping the original labels and classes intact. In this scenario, no usable image features are available at all, allowing us to establish the baseline performance achievable purely by random guessing. This lower bound is defined as 0 in the performance score, ensuring that the score provides a consistent reflection of relative performances across all datasets. With this normalization, a score of 1 represents "optimal" performance, while a score of 0 corresponds to "random guessing," offering a standardized and interpretable range for evaluating the impact of missing or impeded image features. The so

calculated performance metric will be called "relative performance" throughout the thesis.

Now, the interpretation of relative performance differs based on whether one or two image features were impaired by the applied transformations. If a single transformation suppressing image features of one category is applied, the resulting loss in relative performance can be interpreted as the reliance of models trained on that dataset on the suppressed feature. If two transformations are applied suppressing two categories of image features, the remaining relative performance can be interpreted as the contribution of the independent, unaffected image feature to the overall classification performance. However, this does not imply that image features are entirely independent of each other in terms of their impact on classification performance. Conversely, by analyzing the relative performance when only one image feature remains intact, we can gain insights into how interconnected the image features are for a specific model and dataset, providing a deeper understanding of the interaction between image features in contributing to classification accuracy.

5 Datasets and Experimental Setup

This chapter presents the datasets utilized in this study and outlines the experimental setup, including selected architectures and training methodology.

5.1 Datasets

For our experiments, we utilized four datasets, including two from the RS domain and two from the CV domain. From the RS domain, we selected BigEarthNet-S2 and DeepGlobe, which represent classical RS datasets with diverse characteristics. From the CV domain, we chose ImageNet-1K and Caltech, which are well-established CV datasets. The detailed characteristics of these datasets are presented in Table 5.1, providing an overview of their properties relevant to our experiments.

Table 5.1: Summary of key attributes of various scene classification datasets, including the number of images ($||D||$), unique labels (L), average labels per image, number of channels (C), image dimensions and spatial resolution.

Datasets	$\|D\|$	L	**Avg.** L **per Image**	C	**Image Size (Spatial Resolution)**
BigEarthNet-S2 [11]	250,249	19	2.95	12	120×120 (10 m), 60×60 (20 m)
DeepGlobe [12]	30,443	6	1.71	3	120×120 (0.5 m)
ImageNet-1K [9]	1,281,167	1000	1	3	224×224 (-)
Caltech [13]	4,230	20	1	3	224×224 (-)

Figure 5.1: Example images from the BigEarthNet-S2 dataset.

The BigEarthNet-S2 dataset [11] is a multi-label dataset derived from Sentinel-2 multispectral

images captured over ten European countries, with example images visualized in Figure 5.1. It contains 590,326 image patches, each annotated with a subset of 19 Land Use and Land Cover (LULC) classes, including various forest types, water bodies, and complex urban or agricultural areas. On average, each patch contains 2.95 annotated classes.

Figure 5.2: Example images from the DeepGlobe dataset.

The DeepGlobe Land Cover Classification Challenge (DeepGlobe-LCCC) dataset [42] consists of 1,949 RGB tiles, each measuring 2448 × 2448 pixels with a spatial resolution of 0.5 m, collected from regions in Thailand, Indonesia, and India. The dataset includes the classes urban, agriculture, rangeland, forest, water, and barren, along with an additional "unknown" class.

For this thesis, the DeepGlobe dataset is used, as derived by Burgert et al. [12]. Smaller patches of 120 × 120 pixels were extracted from the original tiles, excluding those containing the "unknown" class. Examples images of the dataset can be seen in Figure 5.2. The dataset includes 20% of patches with a single present class and all patches with multiple present classes, resulting in an average of 1.71 classes per patch, comprising a total of 30,443 patches.

Figure 5.3: Example images from the ImageNet dataset.

The ImageNet-1K dataset [9] is a large-scale, single-label classification dataset derived from the broader ImageNet database. It consists of 1,281,167 images, spanning 1,000 diverse object classes. Each image is labeled with one primary class, ensuring a balanced distribution of a minimum of 1,000 images per class in the training set. The classes are organized according to the WordNet hierarchy, encompassing a wide range of categories such as animals, vehicles, plants, and everyday objects, whcih can be seen in Figure 5.3 ImageNet-1K has served as a benchmark for various CV tasks and contributed significantly to advancements in deep learning, particularly through its role in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

Figure 5.4: Example images from the Caltech101 dataset.

The Caltech101 [13] dataset is a collection of images designed to facilitate CV research in the areas of object detection and image classification. It contains 9,144 images with 101 distinct categories of objects. The categories are diverse, including a variety of human faces, animals, household objects, and other common items, each represented by 40 to 800 images. Examples images of the dataset can be seen in Figure 5.4 To gain access to a dataset with a comparably small number of label classes, as common in the RS domain, only the twenty classes containing the highest sample count were included, additionally increasing the smallest sampled class count to 80 improving the overall class distribution. The so generated dataset will be called Caltech. Hereby, the *car side* class, containing grayscale images unlike the rest of the dataset, are ignored due to their lack of full spectral content and therefore a hindrance to a fair comparison of image features.

Additionally, three dataset and training variations are employed during experiments. Caltech-120 is a variation of the Caltech dataset, with images downsized to a image resolution of 120x120 px. RGB-BigEarthNet is a variation of the BigEarthNet dataset, with only RGB channels included for spectral content. Finally, Caltech-FT described a variation in pretraining. The same data as Caltech is included, however ImageNet pretrained models were used, and subsequently finetuned on the data. For an easier comparison, results measured by this approach will be compared to the results of models across other datasets.

5.2 Design of Experiments

This section gives an overview about the model architectures proposed to give an assessment of image feature reliances for the different ImageNet, Caltech, BigEarthNet and DeepGlobe datasets, as well as the employed training strategy. In contrast to previous works, we chose model architectures not only from one architectural archetype, CNN or vision transformer, but picked architectures of both to gain a more generalized overview independent from specific properties each might contain. All specific model architectures were chosen for comparable parameter sizes, with less than one order of magnitude between the architecture, with the lowest amount of parameters being 4.9 million and highest amount of parameters with 30 million respectively. The Python *PyTorch* library is used for implementation of all experiments, from model training to evaluation. For all models, the *timm* library is used to handle specific implementations of model architectures. All names of specific model architectures, as presented by 5.2 and 5.3 relate to the implementation by the library.

For CNN, we chose ten different model architectures, as presented in 5.2. These models

provide a broad overview over the current state-of-the-art regarding CNN architectures, with them collectively representing key advancements in CNN architecture, illustrating diverse design principles that have shaped modern CV. Innovations such as residual connections (ResNet [43]), dense connectivity (DenseNet [44]), and grouped convolutions (ResNeXt [45]) address the challenges of training deep networks. Multi-branch designs (Inception [46]) and separable convolutions (Xception [47]) enhance feature extraction, while systematic scaling approaches (RegNet [48]) and hybrid designs (ConvNeXt [49]) demonstrate the adaptability of CNNs. Together, these models provide a comprehensive overview of foundational and cutting-edge CNN design strategies.

Table 5.2: Overview of selected CNN Architectures.

Model Family	Specific Architecture	Parameter Count
ResNet [43]	ResNet50	25 M
EfficientNet [50]	EfficientNet-B5	30 M
ConvNeXt [49]	ConvNeXt-Tiny	28 M
RegNetX [48]	RegNetX-016	16 M
DenseNet [44]	DenseNet161	28 M
ResNeXt [45]	ResNeXt50_32x4d	25 M
MobileNetV3 [51]	MobileNetV3-Large	5.5 M
Xception [47]	Xception	23 M
Inception [46]	Inception V3	27 M
RegNetY [48]	RegNetY-004	20 M

Regarding vision transformers, we chose six different architectures as listed in 5.3. The chosen architectures comparably represent the current state-of-the-art of vision transformers, including improvements upon the original vision transformer (ViT) as proposed by [52]. Architectures optimize training for vision transformers with fewer data requirements (DeiT [53]), improves performance through hierarchical designs and attention refinements (CaiT [54]), incorporate multi-scale features for better adaptability to downstream tasks (PVT [55]), or introduces pooling mechanisms to enhance computational efficiency (PiT [56]). Hybrid designs such as ConvMixer [57] blend convolutional and transformer principles, showcasing the versatility of transformer architectures.

Table 5.3: Overview of selected Transformer Architectures.

Model Family	Specific Architecture	Parameter Count
ViT [10]	Vit-Tiny-Patch16-224	5.7 M
DeiT [53]	DeiT-Tiny-Patch16	5.7 M
CaiT [54]	CaiT-S24-224	24 M
PVT [55]	PVT-V2-B2	25 M
PiT [56]	PiT-Ti-224	4.9 M
ConvMixer [57]	ConvMixer-768-32	21 M

The models were trained on all the different datasets, with instances of each model being trained separately on each dataset, as previously described in Section 5.1.

The training strategy employed is not highly optimized for maximizing model performance. The primary goal is not to achieve high classification accuracy but to understand feature reliance on the datasets. To this end, we use a training strategy designed to provide a basic starting point for analyzing feature reliance within the RS domain. This approach ensures that the focus remains on evaluating feature reliance rather than on fine-tuning model performance. The models were trained using the Adam optimizer [58] with a learning rate (LR) of 1×10^{-4}. A cosine annealing learning rate scheduler [59] is applied, with the learning rate gradually reduced to a minimum LR of 1×10^{-5} over the course of training.

The cross entropy (CE) loss function is used for multi-class datasets, while the binary cross entropy (BCE) loss function is employed for multi-label datasets. The cross entropy (CE) loss function is a standard choice for multi-class classification problems, as it effectively penalizes incorrect predictions by comparing the predicted probability distribution over classes to the true class labels. This ensures that the model learns to output high probabilities for the correct class while minimizing uncertainty. For multi-label datasets, the binary cross entropy (BCE) loss function is more appropriate, as it treats each class prediction as an independent binary classification task. This allows the model to handle scenarios where multiple labels can be simultaneously assigned to a single sample, ensuring proper optimization across all relevant classes.

Models were trained for a total of 30 epochs, with a batch size of 32 and model checkpointing enabled. Early stopping is used with a patience of 5 epochs, terminating training when the performance are observed to converge, with all models converging within the training duration. For the ImageNet dataset, due to the immense computational cost of training, pretrained models were utilized. Model checkpoints with the best validation score were selected for downstream evaluation. Test performance experiments under single transformation intensities were approximated by a random subset of 16,000 images, due to computational costs. All experiments were conducted on a system using a Tesla V100-SXM2 graphics card with 32 GB of memory and an Intel(R) Xeon(R) Gold 6152 CPU.

6 Experimental Results

In this chapter, the experimental results and discussions are organized into four sections. The first examines the effect of suppressing single image feature categories, followed by an analysis of feature reliance when only single image feature categories are unaffected. The third section provides a class-wise analysis, highlighting variations in feature reliance across individual classes. Finally, the influence of model architectures on feature reliance is explored, offering insights into how design and pretraining affect feature utilization.

6.1 Suppressing Single Image Features

This section provides a dataset-wise analysis of feature reliance for the three image feature types: spectral, texture, and shape. The datasets analyzed are BigEarthNet, DeepGlobe, ImageNet, and Caltech. For each feature category, the analysis is conducted comparatively across all datasets. Results are presented separately for each domain, with datasets from the RS domain visualized in shades of blue and datasets from the CV domain in shades of red. The performance scores used are relative performance scores, as described in Section 4.3, ensuring a consistent basis for comparison.

6.1.1 Spectral Features Suppressed

In Figure 6.1, dataset-wide reliance on spectral features is visualized by performance reductions through the *channel shuffle* transformations. Channel shuffling has a significant effect on the classification performance of models across all three datasets of the RS domain they were trained on. Performance consistently decreases as the proportion of shuffled channels increases. BigEarthNet experiences the highest effect, with classification performance decreasing by 93% when all channels are shuffled. RGB-BigEarthNet exhibits a strong effect, with performance decreasing by 79% under the same condition. DeepGlobe trained models see the lowest effect among the RS datasets, losing 65% relative performance when all channels are shuffled. Notably, BigEarthNet with 12 channels overall suffers a similar performance loss of 56% with only two (i.e. 1/6 of) channels shuffled. The smallest effect overall is observed for DeepGlobe when two channels are switched, losing 59% performance.

For datasets in the CV domain, *channel shuffle* has a moderate effect on classification performances of models. All datasets consist of three image channels, meaning a share of 0.67 shuffled channels corresponds to two channels being switched. For Caltech, performance decreases by 39% with two channels switched and 45% when all three channels are shuffled. ImageNet experiences a performance decrease by 22% when all channels are shuffled. In contrast, Caltech-FT shows virtually no performance decrease with 1% relative performance lost with all channels shuffled, representing the lowest impact observed overall.

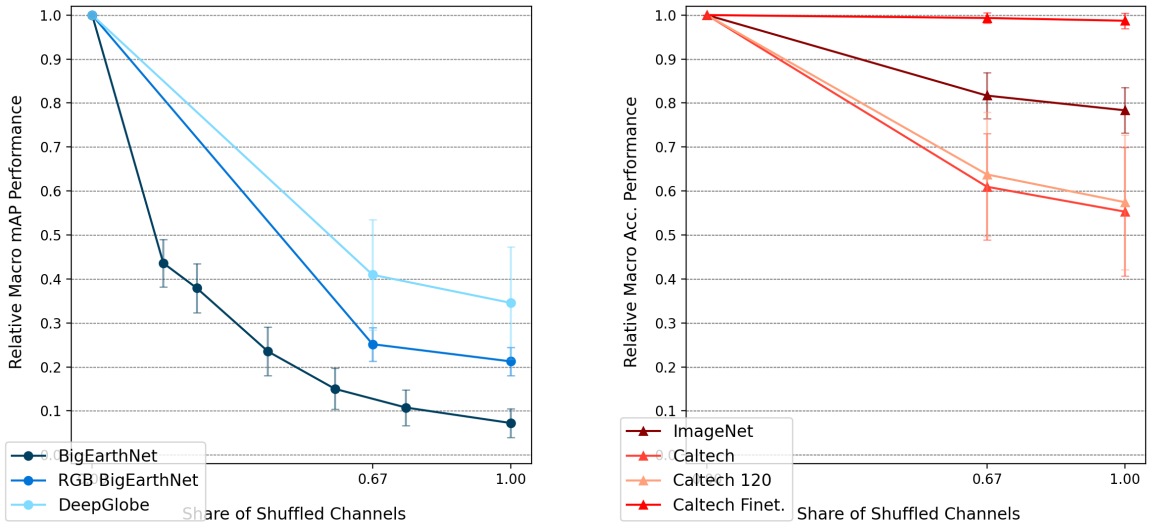


Figure 6.1: Visualization of the effect of *channel shuffle* transformation suppressing spectral features on classification performance. Plots show averaged relative performance under increasing transformation intensity, with the relative performance metric on the y-axis, the shape of shuffled channels on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

In Figure 6.2, dataset-wide reliance on spectral features is visualized by performance reductions through the *channel inversion* transformations.

Channel inversion has an even greater effect on the classification performance of all RS datasets, indicating a high reliance on spectral features. BigEarthNet trained models experience the greatest impact, with performances decreasing by full 100% when all channels are inverted. RGB-BigEarthNet sees a similarly strong impact, with a relative performance decrease of 96% when two of its three channels are inverted. Interestingly, with all three channels inverted, the effect on RGB-BigEarthNet models performances slightly decreases to 92%. Here, unlike other spectral feature-suppressing transformations, *channel inversion* does not exhibit a consistent decrease in classification performance with an increasing number of inverted channels. DeepGlobe demonstrates a similar impact, of 98% with all channels inverted. The lowest measured effect overall is the inversion of a single channel for BigEarthNet, which still reduces performance by 71%.

For CV datasets, *channel inversion* demonstrates the highest effect on classification performances among all transformations affecting spectral content. Caltech exhibits a very high impact regardless of the number of channels inverted, with two channels inverted generating the most significant performance drop of 97% and all three channels inverted resulting in a relative performance loss of 89%. Models on ImageNet see a high effect with one and two channels inverted, with performance decreases of 56% and 61%, respectively. However, when all three

channels are inverted, the performance impact decreases dramatically to 18%. A similar trend is observed in Caltech-FT, where performances are reduced by 44% and 47% for one and two channels inverted, respectively, but improve significantly to only 3% reduction when all three channels are inverted. This phenomenon, also noted in RGB-BigEarthNet, is likely due to the inversion of the entire image preserving relative contrasts critical for edge features, whereas partial inversion disrupts these contrasts. This suggests a potential reliance on shape features in the affected datasets. Additionally, the standard deviation of performance for both ImageNet and Caltech-FT is notably high, with values reaching up to 28%, indicating high variability in performance depending on model architectures.

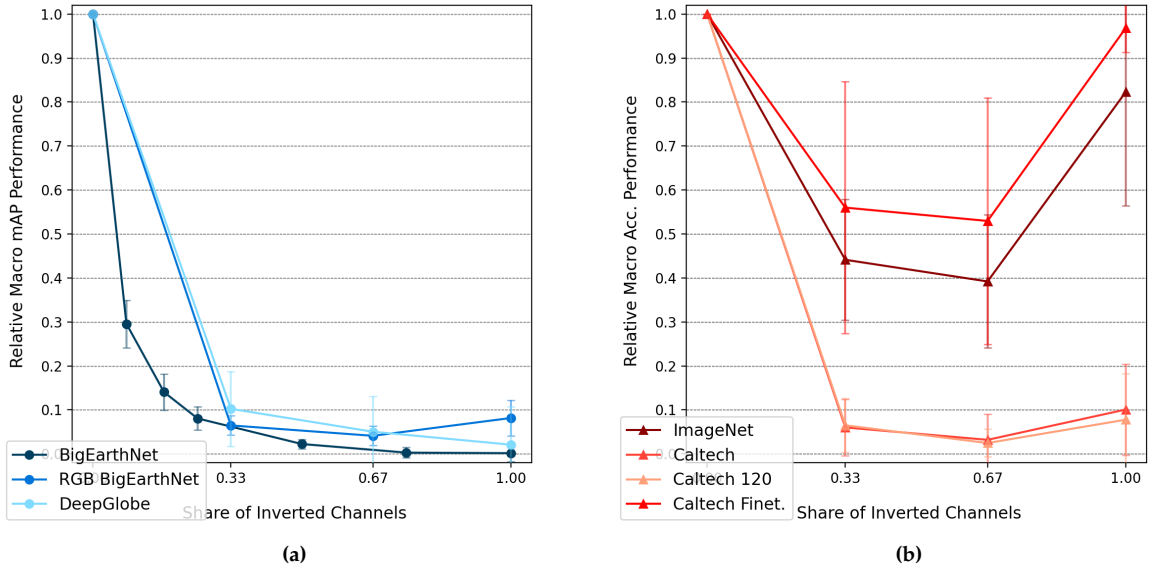


Figure 6.2: Visualization of the effect of *channel inversion* transformation suppressing spectral features on classification performance. Plots show relative performance under increasing transformation intensity, with the relative performance metric on the y-axis, the share of inverted channels on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

In Figure 6.3, dataset-wide reliance on spectral content is visualized by performance reductions through the *channel mean* transformations.

The *channel mean* transformation has a moderately high effect on the classification performance on models of the RS datasets. BigEarthNet and RGB-BigEarthNet see similar performance decreases, with relative performance losses of 85% and 88%, respectively, when channels are completely averaged. At lower averaging factors, BigEarthNet models appear slightly more affected than RGB-BigEarthNet models, with relative performances losses of 21% and 14%, respectively, for an averaging factor of 0.3. High standard deviation values, particularly for BigEarthNet, suggest that models handle partially averaged channels with varying effectiveness. DeepGlobe models experiences a comparatively lesser performance loss, losing only 47% relative performance when image channels are entirely averaged, representing the low-

est impact at full transformation intensity of all rs datasets. For CV datasets, the *channel mean* transformation also has a relatively low effect on classification performances. Averaged relative performance losses for all datasets remain below 6% with an averaging factor of 0.5. Notable performance decreases are observed only when the factor reaches 0.7 or higher. For entirely average channels, Caltech-FT demonstrates virtually no performance decrease, retaining 99% relative performance. ImageNet shows a modest performance decrease by 18%, while Caltech experiences a more pronounced decrease by 30% for an averaging factor of 1.

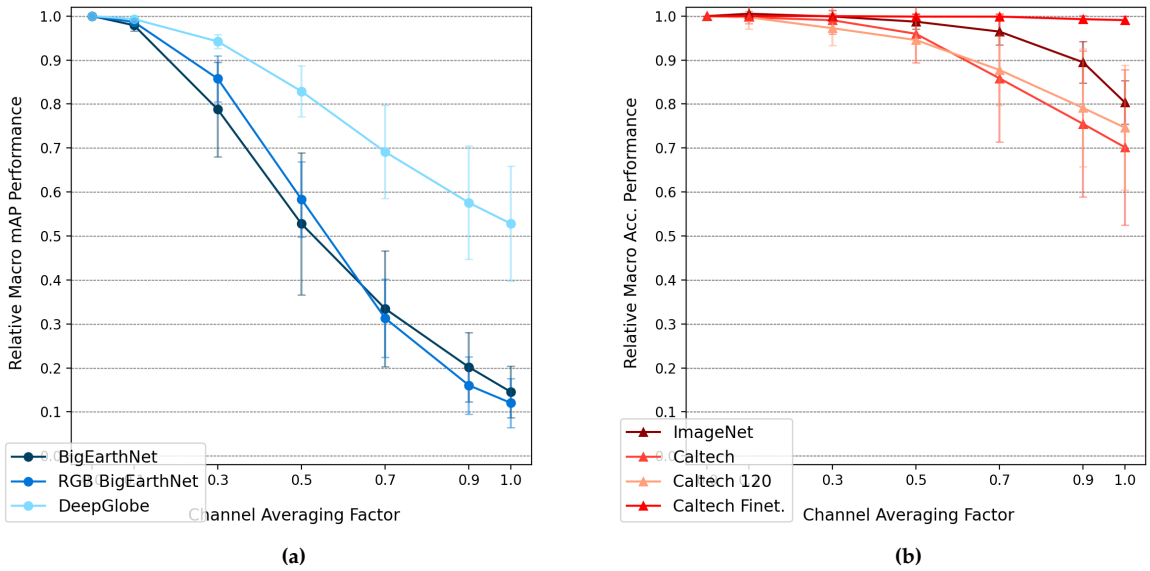


Figure 6.3: Visualization of the impact of *channel mean* transformation suppressing spectral features on classification performance. Plots show relative performance under increasing transformation intensity, with the relative performance metric on the y-axis, the channel averaging factor on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

In Table 6.1 model-averaged relative performance scores of each dataset are presented for all three spectral content suppressing transformations. Over all three chosen transformations affecting spectral content: *channel shuffle*, *channel inversion* and *channel mean*, models trained on the BigEarthNet dataset see the strongest reduction of classification ability, closely followed by RGB-BEN. Caltech models exhibit the highest feature-reliances for spectral content of researched CV datasets, with an extremely high effect observed for the *channel inversion* transformation. A possible reason for that might be the comparably low number of training samples, possibly leading to bad generalization capabilities of models, overfitting predictions on specific spectral values. The lowest feature-reliance for spectral content is observed for Caltech-FT models, followed by ImageNet models. Pretraining on ImageNet apparently lead to a better generalization over multiple image feature types, as Caltech-FT models appear to have no difficulties classifying between 20 classes without reliable spectral content. ImageNet trained models see similar exhibited performance losses between thoses of Caltech and Caltech-FT.

Table 6.1: Averaged relative performances of models for each dataset, with spectral content affected by either *channel shuffle*, *channel inversion* or *channel mean* transformation. Transformations were applied with highest intensity parameters, with all channels shuffled, inverted and channels entirely averaged respectively. Best relative performances across transformations for each dataset in bold.

Dataset	Channel Shuffle	Channel Inversion	Channel Mean	**Average**
BigEarthNet	0.07	0.00	**0.15**	0.07
DeepGlobe	0.35	0.02	**0.53**	0.30
RGB-BigEarthNet	**0.21**	0.08	0.12	0.14
ImageNet	0.78	**0.82**	0.80	0.80
Caltech	0.55	0.11	**0.70**	0.45
Caltech-FT	**0.99**	0.97	**0.99**	0.98
Caltech-120	0.58	0.09	**0.75**	0.47

This way, even though models on Caltech-FT were finetuned on a small dataset with a strong possibility of overfitting to the data, models appear to have not done so, showing a reliance on spectral content closer to ImageNet trained models than to Caltech trained ones. Interestingly, models on DeepGlobe exhibits a lower feature-reliance on spectral content than BigEarthNet. While DeepGlobe possesses only 6 label classes instead of BigEarthNets 19, making a spectral differentiation between classes easier as there are only fewer to decide between, BigEarthNet possesses 12 channels utilized in our experiment, offering more spectral content to differentiate classes. Although the models on both datasets seem highly reliant on spectral content, BigEarthNet trained models appear to be even more dominantly so, with relative performance losses over 90% across all three transformations. This could be explained two factors, the predictive strength other spectral bands besides RGB offer, as well as the strong difference in spectral resolution, making a classification based on spatial features like texture and shape more viable in the DeepGlobe dataset. For all transformations, *channel inversion* appears to be the most aggressive transformation overall, with highest performance reduction for 6 of the 7 compared datasets, only except ImageNet. Channel Mean, in contrast, seems to be the least aggressive transformation, with the highest relative performance for 5 of the 7 datasets.

When previously averaged scores over all models were examined to highlight a general feature reliance over most models, Table 6.2 summarizes the overall highest relative model performances under the three spectral content suppressing transformations. These performance values indicate to what extend the missing features are entirely necessary for prediction performances across all models. Here, we again observe a striking difference between our selection of RS and CV domain datasets. While DeepGlobe sees the highest performance values for *channel mean* of 72% and *channel shuffle* of 56%, BigEarthNet and RGB-BigEarthNet have the highest performances of 23% and 27%, respectively. In contrast, all CV datasets have relative performances for at least one transformation category of 96% or higher. Overall, without exception, measured spectral content reliance is higher on all rs datasets than on any cv dataset for all three transformations.

Table 6.2: Highest relative performance of all models for each dataset, with spectral content affected by either *channel shuffle*, *channel inversion* or *channel mean* transformation. Transformations were applied with highest intensity parameters, with all channels shuffled, inverted and channels entirely averaged respectively. Best relative performances across transformations for each dataset in bold.

Dataset	Channel Shuffle	Channel Inversion	Channel Mean	**Average**
BigEarthNet	0.13	0.05	**0.23**	0.14
DeepGlobe	0.56	0.15	**0.72**	0.48
RGB-BigEarthNet	**0.27**	0.16	0.22	0.22
ImageNet	0.86	**1.00**	0.88	0.92
Caltech	0.91	0.28	**0.97**	0.72
Caltech-FT	**1.00**	**1.00**	**1.00**	1.00
Caltech-120	**0.96**	0.28	**0.96**	0.73

6.1.2 Texture Features Suppressed

In Figure 6.4, we see the effect of an application of the *bilateral filter* transformation for models across all datasets. The *bilateral filter* transformation has a moderately high effect on models trained on the RS datasets BigEarthNet, RGB-BigEarthNet, and DeepGlobe. Performance decreases continuously with increasing kernel diameter values. Among the datasets, RGB-BigEarthNet models experience the strongest effect with performance reduced by 58% for a kernel diameter of 15. DeepGlobe models exhibit the weakest effect of 28% performance reduction for the same kernel diameter, while BigEarthNet models show an intermediate effect with a performance reduction of 30%. Performance variability is high across all three datasets, with standard deviations reaching up to 21% for RGB-BigEarthNet.

The *bilateral filter* transformation has a lesser effect on models trained on the selected CV datasets. Models trained on ImageNet experience the strongest effect, with relative performances loss of 27%. Models trained on Caltech and Caltech-FT see a small effect by maintaining average relative performances above 90%, with Caltech-FT loosing only 1% and Caltech 6% performance for the highest transformation intensity.

The *median filter* transformation results in Figure 6.5, show a moderate effect on models trained on the selected RS datasets. BigEarthNet shows the lowest effect, with an average relative performance of 29% for the largest kernel size of 15. DeepGlobe exhibits a higher performance decrease of 50%, while RGB-BigEarthNet models experience the highest effect, with a relative performance reduction of 63% for the same kernel size. As observed with the *bilateral filter*, high standard deviation values suggest that the effect varies significantly depending on the model architecture.

Models trained on the selected CV datasets exhibit a comparable performance decrease. Caltech and Caltech-FT are moderately affected, with relative performances of 79% and 77%, respectively, for the largest kernel size of 15. Models on Caltech-FT consistently outperform those on Caltech by approximately 5% for kernel sizes ranging from 5 to 11. ImageNet-trained models show the most significant performance decrease, retaining only 23% relative performance for the largest kernel size, representing the strongest effect across all datasets.

The *gaussian filter* transformation result shown in Figure 6.6, shows a slightly stronger but

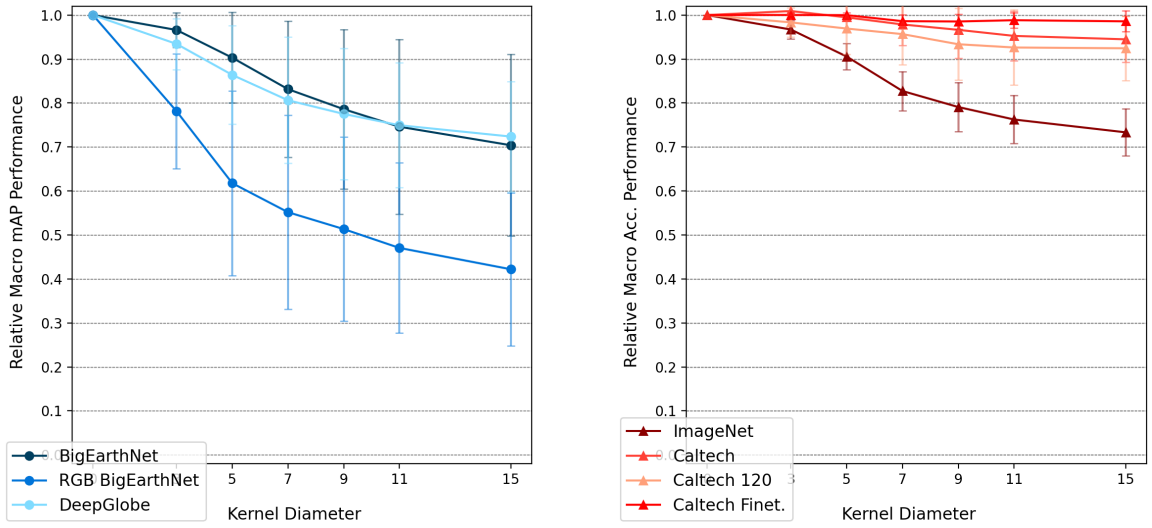


Figure 6.4: Visualization of the effect of *bilateral filter* transformation suppressing texture features on classification performance. Plots show relative performance under increasing transformation intensity, with the relative performance metric on the y-axis, the kernel diameter parameter on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

similar effect compared to the *median filter* transformation on relative performance scores for RS datasets. Models performed best on BigEarthNet, seeing a relative performance loss of 33% with the highest standard deviation parameter of 7. Performance is lower on DeepGlobe, with a relative performance loss of 52% for the same parameter, while models on RGB-BigEarthNet experienced the strongest effect, with a relative performance drop of 71%. High variability in results across models is observed, with the greatest variability noted for BigEarthNet. For CV datasets, models exhibited lower effects for smaller standard deviation parameter values compared to RS datasets but a larger effect at the highest value. As with RS datasets, results closely parallel those of the *median filter* transformation, with a slightly stronger performance reduction. ImageNet models see the largest effect among all datasets, with a relative performance loss of 78% at the highest transformation intensity. Caltech-FT models demonstrate the weakest effect, with a performance decrease on only 32% for the highest intensity. Caltech models display a intermediary effect, loosing 48% relative performance. Notably, Caltech and Caltech-FT exhibited high variability across models, whereas ImageNet did not, indicating that no models trained on ImageNet achieved significantly higher performances.

In Table 6.3 the model-averaged relative performance scores for each dataset are summarized across the spectral content suppressing transformations. Over all three chosen transformations affecting texture features: *bilateral filter*, *median filter* and *gaussian filter*, models trained on the RGB-BigEarthNet dataset see the strongest effect, followed by the ImageNet dataset. The weakest effect is seen on Caltech-FT, followed by Caltech and BigEarthNet. No clear trend for feature

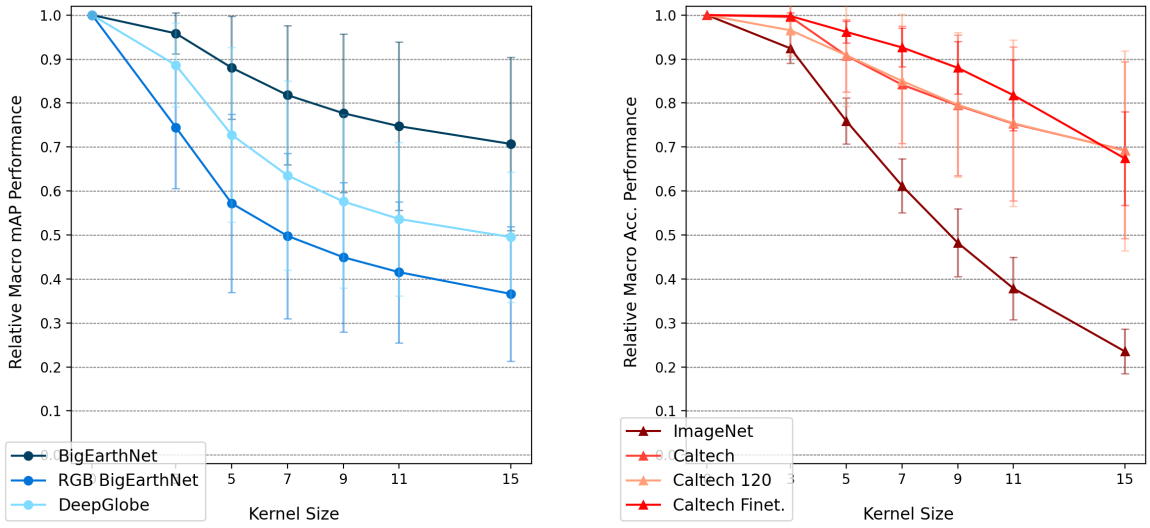


Figure 6.5: Visualization of the effect of *median filter* transformation suppressing texture features on classification performance. Plots show relative performance under increasing transformation intensity, with the relative performance metric on the y-axis, the kernel size parameter on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

Table 6.3: Averaged relative performances of models for each dataset, with texture features affected by either *bilateral filter*, *median filter* or *gaussian filter* transformation applied. Transformations were applied with highest intensity parameters respectively. Best relative performances across transformations for each dataset in bold.

Dataset	Bilateral Filter	Median Filter	Gaussian Filter	Average
BigEarthNet	0.70	**0.71**	0.67	0.69
DeepGlobe	**0.72**	0.50	0.48	0.57
RGB-BigEarthNet	**0.42**	0.37	0.29	0.36
Caltech-120	**0.92**	0.67	0.49	0.69
ImageNet	**0.73**	0.24	0.22	0.40
Caltech	**0.94**	0.67	0.52	0.71
Caltech-FT	**0.99**	0.67	0.68	0.78

reliance on texture features is discernible between our selection of datasets of the RS and CV domain. Bilateral appears to be the least aggressive texture-suppressing transformation with highest relative performance for 6 of 7 datasets, while *gaussian filter* appears to be the most aggressive with models on also 6 out of 7 datasets seeing their lowest relative performances when applied. This can be explained by the difference in edge-preserving capabilities of both

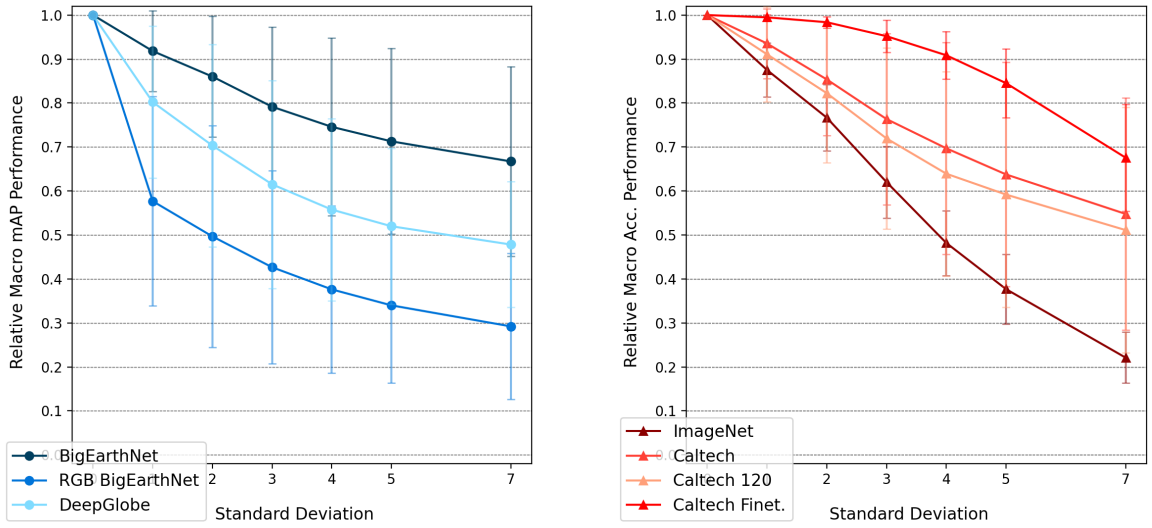


Figure 6.6: Visualization of the effect of *gaussian filter* transformation suppressing texture features on classification performance. Plots show relative performance under increasing transformation intensity, with the relative performance metric on the y-axis, the standard deviation parameter of the applied filter on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

transformations as described in Section 4.2, with the *bilateral filter* affecting other feature types less than the other two transformations, while the *gaussian filter* blurring edges may affect especially shape features. Following this intuition, the almost identical performance values for the BigEarthNet dataset across *bilateral filter* as well as *gaussian filter* transformation of 70% and 67% indicate a low reliance on shape features on the dataset. Comparing the performances of Caltech and Caltech-120, we observe no notable difference in reliance on texture features, indicating a relatively low effect of image resolution on texture reliance. Also, interestingly, when reducing the available spectral content as with the RGB-BigEarthNet dataset, we can observe a higher feature-reliance on texture instead. This observation follows the notion of Hermann and Lampinen [26], that simpler image features may suppress more complex ones.

Table 6.4 summarizes the overall highest relative model performances under the three texture suppressing transformations, giving perspective on definitive information loss under the transformations. Focusing on the *bilateral filter* transformation, we observe that most datasets see a highest relative performance of more than 90%, with only ImageNet losing 17% and RGB-BigEarthNet loosing 23% performance. Given the understanding that the *bilateral filter* is the transformation closest to pure texture feature loss, this can be seen as the most direct gauge of texture feature dependence. We observe that the most complex datasets with the highest number of classes and a lower available spectral content are the most affected, both for our selection of RS and CV datasets.

Table 6.4: Highest relative performance of all models for each dataset, with texture features affected by either *bilateral filter*, *median filter* or *gaussian filter* transformation applied. Transformations were applied with highest intensity parameters respectively. Best relative performances across transformations for each dataset in bold.

Dataset	Bilateral Filter	Median Filter	Gaussian Filter	Average
BigEarthNet	**0.98**	0.96	**0.98**	0.97
DeepGlobe	**0.91**	0.72	0.69	0.77
RGB-BigEarthNet	**0.77**	0.66	0.61	0.68
ImageNet	**0.83**	0.36	0.34	0.51
Caltech	**0.98**	0.94	0.93	0.95
Caltech-FT	**1.00**	0.84	0.87	0.91
Caltech-120	**1.00**	0.96	0.90	0.95

6.1.3 Shape Features Suppressed

In Figure 6.7 we observe the effect of the *patch shuffle* transformation on relative performances across all datasets. Grid size values range from 2 to 15. The performance impact of the *patch shuffle* transformation is relatively low on the selected RS datasets. Among the datasets, models on DeepGlobe and RGB-BigEarthNet experience the strongest effect, with a relative performance decrease of 36% and 33% for the highest grid size. BigEarthNet follows with a performance decrease of 18%. In contrast, the *patch shuffle* transformation has a very strong effect on the selected computer vision datasets. Performance decreases exceed 80% for all datasets at the highest grid size. Interestingly, models on Caltech exhibits the fastest performance decline, with a sharp decrease of 38% for a grid size of 2, followed by a slower decrease to 82% for a grid size of 15. ImageNet models show only a small effect of 13% for a grid size of 2, but performance drops significantly, with a 97% decrease for the highest grid size. Caltech-FT follows a similar performance reduction curve to ImageNet, with slightly lower effect of around 10% for small grid sizes and a decrease of 92% for the highest grid size.

These results demonstrate that *patch shuffle* with grid sizes of 2 or 4, which already disrupt the entire global shape of objects, does not significantly affect classification performances. This observation aligns with the findings of Brendel et al. [3] and Baker et al. [2], which indicate that global shape is not essential for accurate object classification. However, when local shape features are progressively destroyed by splitting the image into increasingly smaller patches, performance decreases dramatically. This highlights the important distinction between the importance of global shape and the general importance of shape features in classification.

Figure 6.8 illustrates the effect on classification performances when applying the *patch rotation* transformation. The *patch rotation* transformation has an even weaker effect than *patch shuffle* on models on the selected RS datasets. Grid sizes range from 2 to 15, with all performance reductions below 20%. Models on BigEarthNet see a minimal effect, with a maximum performance reduction of 4%. RGB-BigEarthNet experiences the highest relative effect among the datasets, with a 16% performance decrease for the largest grid size. DeepGlobe models exhibit a weak effect, with a 11% reduction for the highest grid size. Similar to *patch shuffle*, *patch rotation* has a strong effect on models trained on computer vision datasets. ImageNet

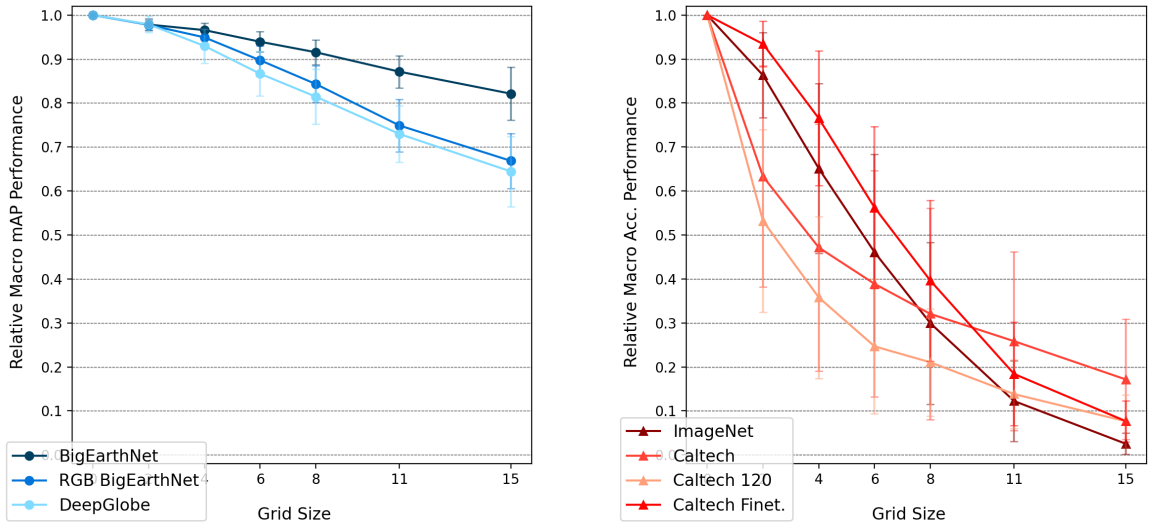


Figure 6.7: Visualization of the effect of *patch shuffle* transformation suppressing shape features on classification performance. Plots show relative performance under increasing transformation intensity, with the relative performance metric on the y-axis, the grid size parameter on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

shows a continuous performance decrease, reaching up to 95%. Caltech-FT models follow this trend with a slightly lesser effect, reaching a performance decrease of 85% for the largest grid size. Interestingly, models on Caltech exhibits the largest effect at the smallest active grid size of 2, with a performance reduction of 35%. For higher grid sizes, the effect decreases, with a reduction of only 34% for a grid size of 15.

One explanation for the observed differences between *patch shuffle* and *patch rotation*, as described in Section 4.2, lies in the nature of the transformations. *patch shuffle* disrupts object silhouettes by rearranging image parts, whereas *patch rotation* does not necessarily have the same effect. When applying *patch rotation* with a high number of patches, the patches remain in place, and the rotation primarily moves pixels over smaller distances. Consequently, a higher grid size may allow the object silhouette to remain largely intact. This observation suggests that models trained on Caltech might rely on rough object silhouettes for classification.

Table 6.5 summarizes the model-averaged relative performances across all datasets for shape feature suppressing transformations applied. When observing the performance decreases of models on all datasets, we see a strong difference between reliance on shape features exhibited on datasets of the RS domain and of the CV domain. Overall, *patch rotation* appears to be the less aggressive transformation with all averaged relative performances being higher, which can be explained by our intuition of global silhouettes being only partially destroyed at higher grid sizes. While averaged relative performances on all RS datasets exceed 60% for *patch shuffle*, the highest performance for CV datasets is Caltech with 18%, with all other datasets in the single

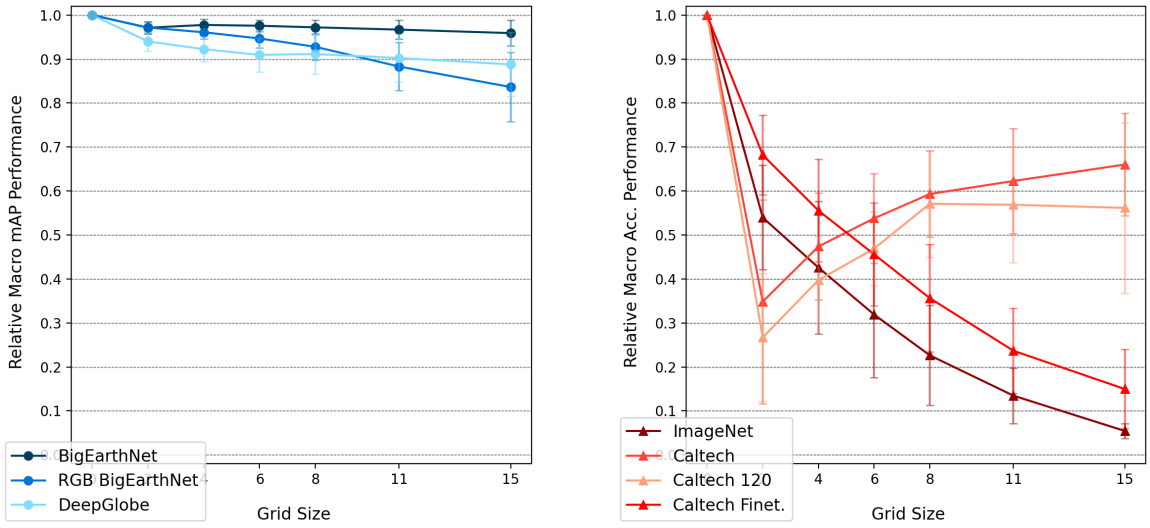


Figure 6.8: Visualization of the effect of *patch rotation* transformation suppressing shape features on classification performance. Plots show relative performance under increasing transformation intensity, with the relative performance metric on the y-axix, the grid size parameter on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

Table 6.5: Averaged relative performances of models for each dataset, with shape features affected by either *patch shuffle* or *patch rotation* transformation applied. Transformations were applied with highest intensity parameters respectively. Best relative performances across transformations for each dataset in bold.

Dataset	Patch Shuffle	Patch Rotation	Average
BigEarthNet	0.82	**0.96**	0.89
DeepGlobe	0.64	**0.89**	0.77
RGB-BigEarthNet	0.67	**0.84**	0.75
ImageNet	0.03	**0.05**	0.04
Caltech	0.18	**0.65**	0.42
Caltech-FT	0.08	**0.15**	0.11
Caltech-120	0.08	**0.55**	0.31

digits. Notably, comparing the performances of models on Caltech-120 to those of Caltech, we observe an increased reliance on shape features, indicating a possibly higher importance of simple shape features when texture features possibly become less distinct. Models on ImageNet and BigEarthNet see the largest difference, with ImageNet models scoring 3 and 5% for *patch shuffle* and *patch rotation*, while BigEarthNet models see performances of 82% and 96%. This highlights the strong difference between current go-to datasets for both domains, and thereby

the need for further analysis of image feature reliance when adopting an understanding of DNN from works of the CV domain.

When analyzing the highest relative performances across all datasets in Table 6.6, we can see a similar trend as before. Caltech sees a significant improvement, highlighting the possibility to partially solve datasets with smaller class amounts with less shape information. ImageNet however sees little improvement, with highest relative performances at or below 10 %. Meanwhile, all RS see performances of over 75% for *patch shuffle* and 95% for *patch rotation*, showing the low reliance on shape features for the datasets.

Table 6.6: Highest relative performance of all models for each dataset, with shape features affected by either *patch shuffle* or *patch rotation* transformation applied. Transformations were applied with highest intensity parameters respectively. Best relative performances across transformations for each dataset in bold.

Dataset	Patch Shuffle	Patch Rotation	Average
BigEarthNet	0.91	**0.99**	0.95
DeepGlobe	0.78	**0.98**	0.88
RGB-BigEarthNet	0.81	**0.96**	0.88
ImageNet	**0.10**	0.08	0.09
Caltech	0.40	**0.85**	0.63
Caltech-FT	0.15	**0.40**	0.28
Caltech-120	0.20	**0.87**	0.54

6.2 Suppressing Pairs of Image Features

To corroborate the findings presented in Section 6.1, we inverted the availability of image features by preserving the type of feature examined while disturbing the other two. This approach served as a sanity check to ensure the robustness of our overall conclusions on the importance of different image features, by avoiding reliance on single transformation implementations. In addition, it provided insights into which image features performed best when isolated, rather than in combination with others.

To investigate the relative importance of distinct image features for classification performance, three transformations were selected as representative methods for suppressing the spectral, texture, and shape feature categories:

- **Channel Shuffle**: Selected to suppress spectral content, as it linearly increases the difficulty of utilizing spectral information without significantly disturbing other features.
- **Bilateral Filter**: Chosen to suppress texture, as it minimizes the parallel suppression of edge information, as detailed in Section 4.2.
- **Patch Shuffle**: Used to suppress shape, as it destroys global shape silhouettes and is widely adopted in studies of shape versus texture bias [2, 20].

Two of these transformations were applied together for this experiment. Examples of such combinations are illustrated in Figures 4.9-4.11. The transformation intensity is simultaneously increased for both transformations.The lowest intensity corresponds to minimal suppression

of both features, whereas the highest intensity corresponds to the strongest suppression. This setup tests the models' ability to rely exclusively on the remaining, unaffected image feature for classification. Therefore, in the presented plots, a higher score indicates a higher predictive value of the remaining unaffected image feature.

6.2.1 Spectral Features Remaining

Figure 6.9 shows the remaining relative performance across datasets with both texture and shape features suppressed. When image features are impaired to have undisturbed spectral content only, by applying both *bilateral filter* and *patch shuffle*, models trained on the selected CV datasets generally exhibit very low single-feature usefulness of spectral content. All datasets show a remaining performance of less than 20%, with ImageNet-trained models scoring as low as 1%. Caltech models demonstrate the highest remaining relative performances at 18%, though they are more quickly affected than ImageNet at lower transformation intensities. Models trained on RS datasets demonstrate very high remaining relative performance when only spectral content remains unaffected. All datasets achieve scores exceeding 50%. RGB-BigEarthNet and DeepGlobe maintain robust performance, while BigEarthNet achieves nearly 80% performance relying solely on spectral content. These results validate common intuitions regarding the high predictive value of spectral content, particularly for the BigEarthNet dataset.

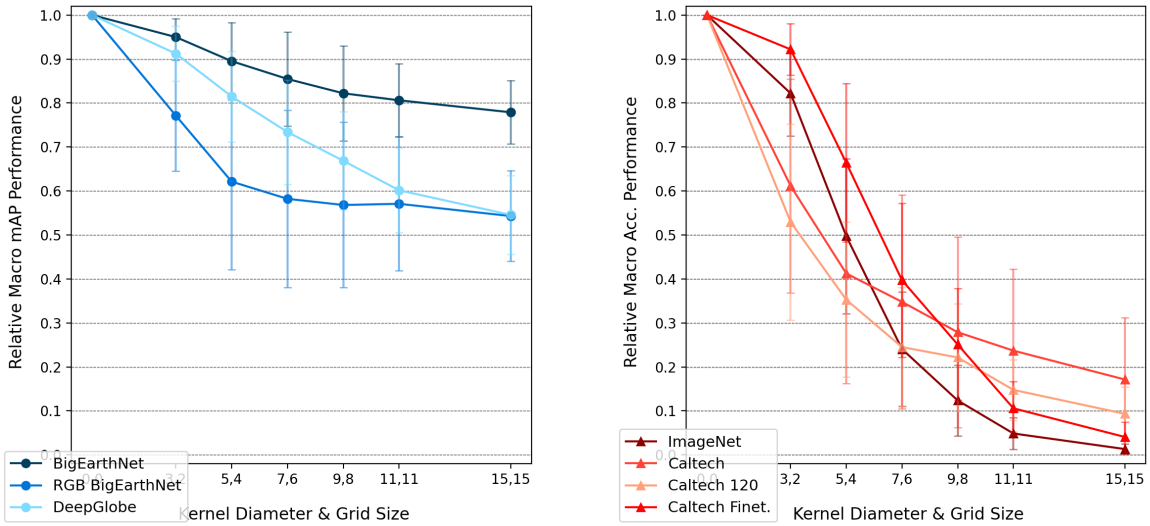


Figure 6.9: Visualization of the predictive value of spectral content remaining for classification performance. Plots show relative performance under increasing transformation intensity of both *bilateral filter* and *patch shuffle* transformation applied, with the relative performance metric on the y-axis, the transformation intensity parameters on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Higher scores indicate higher predictive value of feature type.

We can observe a strong overall difference between the RS and CV datasets in terms of re-

liance on spectral content. Classes of the RS datasets appear significantly better predictable by models utilizing spectral features alone, whereas models on CV datasets show only minimal remaining classification performances under the same conditions. Models trained on CV datasets appear to rely more heavily on other feature types, likely due to a greater availability of such features in CV data. These findings validate our observations presented in Section 6.1, which indicate that models trained on RS data are more dependent on spectral features, while those trained on CV data rely less on this feature type.

6.2.2 Texture Features Remaining

Figure 6.10 shows the remaining relative performance across datasets with both texture and shape features suppressed. Here we observe see that the effect of suppressing image features to only texture features undisturbed, by applying both *patch shuffle* as well as *channel shuffle*:

For models trained on the CV datasets, the remaining relative performance diminishes progressively with increasing transformation intensity. At the highest parameter values, all models achieve less than 5% performance. Caltech models experience the sharpest decline, dropping to 30% at the first transformation intensity, reflecting a high reliance on either shape features or spectral content being present. When both are disturbed, performance is severely impacted. Caltech-FT models demonstrate the highest robustness, maintaining superior relative performance at lower transformation intensities, though its performance also diminishes to 5% at the highest intensity. Models trained on the RS datasets also exhibit small remaining performance values, heavily impacted by the first transformation intensity. This is most likely due to the suppression of spectral content. Among the datasets, DeepGlobe retains a relatively high remaining performance of 24%, while BigEarthNet is more affected, with a remaining performance of 7% at the highest intensity. Noteworthily, RGB-BigEarthNet shows higher remaining performance for texture than BigEarthNet, suggesting that models trained on this dataset learn to rely on more complex feature types when spectral content is less available. This finding aligns with the observations of Hermann and Lampinen [26].

The RS datasets demonstrate greater usefulness of texture features alone compared to the CV datasets, where texture features contribute almost nothing to classification performance. This discrepancy could stem from textures being less unique than shapes in the context of CV datasets, where shape features play a dominant role in distinguishing classes. This difference aligns with our findings from the previous chapter, which showed a higher reliance on texture features in models trained on RS datasets than on CV datasets. Models on RS datasets, such as DeepGlobe, can retain up to 25% classification performance from texture features alone, a notable result given that the texture definition explicitly excludes spectral content. This may be partly explained by the relative simplicity of the DeepGlobe dataset, which involves classifying only six classes, allowing models to rely more less diverse features such as texture. Despite these observations, all datasets exhibit low remaining performances when relying solely on texture features. This indicates a generally low predictive value for texture features across domains, when separating spectral content from texture features. A finding which is highly counterintuitive in light of the results reported by Geirhos et al., which identified texture as the most influential feature for classification.

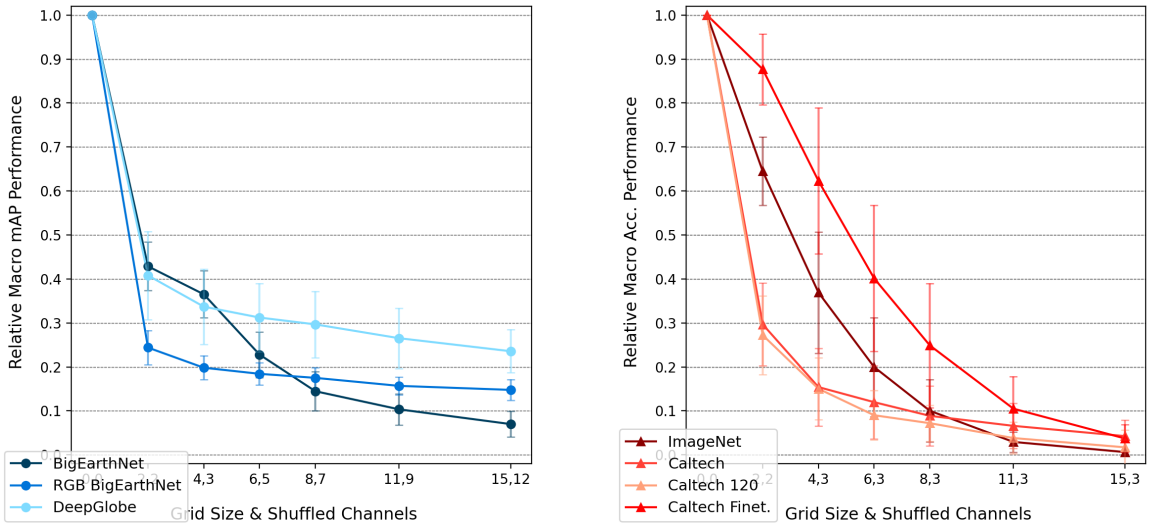


Figure 6.10: Visualization of the predictive value of texture features remaining for classification performance. Plots show relative performance under increasing transformation intensity of both *patch shuffle* and *channel shuffle* transformation applied, with the relative performance metric on the y-axis, the transformation intensity parameters on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Higher scores indicate higher predictive value of feature type.

6.2.3 Shape Features Remaining

Figure 6.11 shows the remaining relative performance across datasets with both spectral and texture features suppressed. We can see that when reducing the image features to only have shape features undisturbed, by applying both *channel shuffle* as well as a *bilateral filter*:

Models trained on the CV datasets exhibit very high remaining relative performances when shape features remain intact. No dataset shows a performance below 45%. Caltech is the most affected, likely due to the suppression of spectral features, with performance quickly dropping to 60%. In contrast, Caltech-FT is nearly unaffected, retaining 94% remaining performance overall. ImageNet demonstrates a continuous decline in performance as spectral content and texture features are increasingly disturbed. These results highlight the strong predictive value of shape features for the selected CV datasets. In contrast, models trained on the RS datasets exhibit comparatively low remaining performances when only shape features remain intact. All three datasets immediately drop to below 45% when spectral content is affected, continuing to final remaining performances of 25% or less. DeepGlobe is the least affected, maintaining a remaining relative performance of 25%, while models on RGB-BigEarthNet and BigEarthNet retain only 12% and 6%, respectively. Notably, RGB-BigEarthNet continues to demonstrate a higher predictive value for non-spectral features compared to BigEarthNet.

The selected CV datasets demonstrate drastically higher usefulness of shape features alone

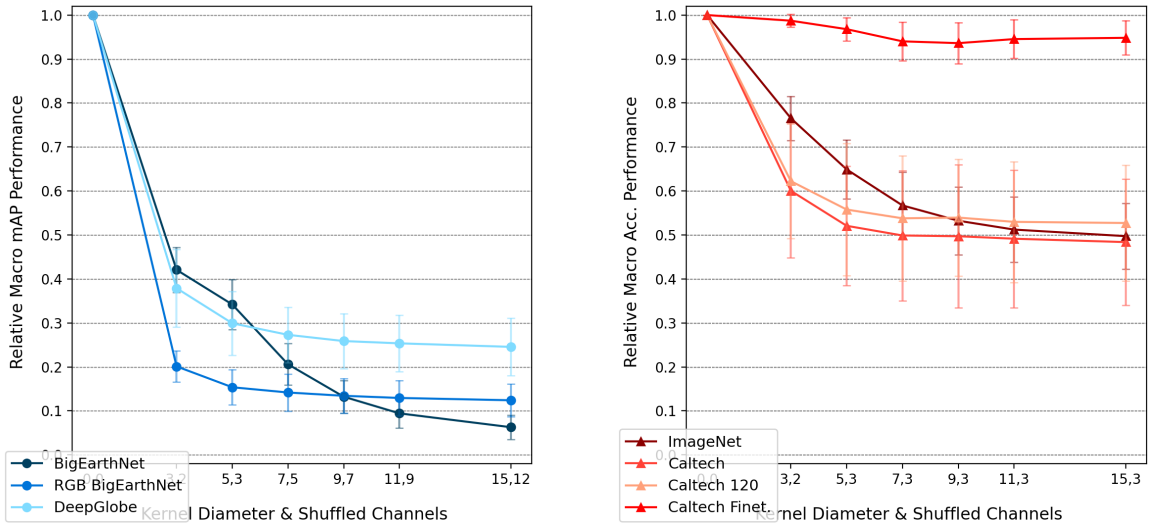


Figure 6.11: Visualization of the predictive value of shape features remaining for classification performance. Plots show relative performance under increasing transformation intensity of both *bilateral filter* and *channel shuffle* transformation applied, with the relative performance metric on the y-axis, the transformation intensity parameters on the x-axis and standard deviation displayed by error caps. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Higher scores indicate higher predictive value of feature type.

compared to the RS datasets. Still, the RS datasets exhibit higher remaining performance for shape features than the CV datasets for texture features. This suggests that while shape features are less prevalent in RS datasets, they do exist and contribute meaningfully to classification performance. The high single-feature predictive strength of shape features in the CV datasets is a particularly interesting finding, as it directly challenges the interpretation of results presented by Geirhos et al. This discrepancy can be explained by their lack of consideration for local shape features, which were explicitly accounted for in our experiments. When both global and local shape features are available, models achieve impressive classification performances on ImageNet, Caltech, and especially Caltech-FT with a remaining classification performance of over 95%. Thereby, these results also highlight the strong shape feature understanding that pretraining on ImageNet provides.

Table 6.7 shows averaged predictive strengths of individual features for models trained across all datasets. The findings indicate that the selected RS datasets exhibit higher predictive power of spectral content compared to their CV counterparts. This result was anticipated for BigEarthNet due to the abundance of spectral content available. However, this trend is similarly pronounced for datasets with only RGB channels, such as RGB-BigEarthNet and DeepGlobe. In contrast, texture and shape features show almost equally low importance when they are the only features present, suggesting that both feature categories alone are insufficient for robust classification in the RS datasets.

Table 6.7: Average remaining relative performances of models for each dataset, with either only spectral, texture or shape features remaining. *Bilateral filter*, *patch shuffle* and *channel shuffle* transformations suppressing both other feature categories were applied with highest intensity parameters respectively. Best score across feature categories for each dataset in bold.

Dataset	Spectral	Texture	Shape	Sum
BigEarthNet	**0.78**	0.07	0.06	0.91
DeepGlobe	**0.55**	0.24	0.25	1.03
RGB-BigEarthNet	**0.54**	0.15	0.12	0.81
ImageNet	0.01	0.01	**0.50**	0.52
Caltech	0.18	0.04	**0.48**	0.70
Caltech-FT	0.04	0.04	**0.95**	1.03
Caltech-120	0.10	0.02	**0.53**	0.64

The findings for the CV datasets differ significantly from those of the RS datasets. Results show that shape features alone exhibit the highest predictive power, maintaining performance levels above 50%, with Caltech-FT achieving nearly 100%. Spectral content alone demonstrates the second-highest predictive power, though its importance is substantially lower than that of shape. Texture features, in contrast, provide the lowest performance, with results approaching zero across all CV datasets. These results challenge the intuition derived from the findings of Geirhos et al. While the low importance of global shape aligns with observations by Baker et al. [2], our findings indicate that shape features overall, not just global shape, are the most critical for classification.

The increased reliance on shape features in models pretrained on ImageNet further underscores their importance. ImageNet-pretrained models show significantly higher performance compared to Caltech trained models, suggesting that pretraining on ImageNet improves the model's ability to interpret and leverage shape features. This improvement is not observed for texture features, likely because texture features alone have limited predictive utility. The relatively low importance of texture features alone, contrary to expectations based on Geirhos et al., is likely explained by the interaction between texture and spectral content, which may function as a combined feature in some contexts. These findings highlight the complexity of feature reliance, which requires careful consideration in interpreting studies directly comparing the importance of image features.

Therefore, for understanding relative feature importance in the RS domain, we observe a dramatically higher importance of spectral features compared to other image features. Texture features exhibit slightly higher importance, while shape features show dramatically lower importance in comparison to their role in typical classification applications in the CV domain. These distinctions provide valuable insights for selecting appropriate training strategies, designing pretraining methods, and optimizing models for RS domain-specific tasks, specifically on the BigEarthNet and DeepGlobe dataset.

Additionally, some datasets appear to present more complex classification problems, requiring a combination of differing features for successful prediction. This is evident from the sum of single-feature remaining scores, such as 0.52 for ImageNet and 1.03 for DeepGlobe. These substantial differences indicate that more complex datasets like ImageNet rely on multiple fea-

tures being present simultaneously to accurately predict classes. In contrast, simpler datasets often require only one dominant feature, with the presence of additional features potentially being redundant.

Table 6.8 shows the highest predictive strengths of individual features for models trained across all datasets. The same trends seen in Table 6.7 can be observed, validating the low predictive strength of Texture, especially for the CV datasets, as well as the high predictive strength of shape features for the CV datasets and the high predictive strength of spectral features for the RS datasets.

Table 6.8: Highest remaining relative performance of all models for each dataset, with either only spectral, texture or shape features remaining. *bilateral filter*, *patch shuffle* and *channel shuffle* transformations suppressing both other feature categories were applied with highest intensity parameters respectively. Best score across feature categories for each dataset in bold.

Dataset	Spectral	Texture	Shape	Sum
BigEarthNet	**0.88**	0.12	0.11	1.11
DeepGlobe	**0.70**	0.38	0.38	1.46
RGB-BigEarthNet	**0.73**	0.19	0.19	1.11
ImageNet	0.04	0.02	**0.62**	0.69
Caltech	0.41	0.10	**0.73**	1.24
Caltech-FT	0.11	0.11	**1.00**	1.23
Caltech-120	0.23	0.11	**0.83**	1.17

6.3 Class-wise Analysis of Feature Reliance

Previous results primarily emphasized the general feature reliance of models across specific datasets, providing insights into overarching trends. However, when aiming to understand which image features are more or less relied upon for classification performance, it is equally valuable to analyze feature reliance on a per-class basis. A class-wise analysis can offer deeper insights into how different classes within a dataset vary in their reliance on specific features. This experiment is designed to quantify the reliance on image features at the class level, providing a clearer understanding of feature importance for individual classes. Such analysis can uncover variations in feature reliance within datasets and give an overview of their magnitude, offering a more nuanced perspective on the relationship between image features and classification tasks.

Both experiments from previous Section involving single transformations (Section 6.1) and pairs of transformations (Section 6.2), designed to suppress either one or two of the three dominant image features, are revisited with a new focus. This time, the analysis examines relative performances for each individual class within the RS datasets BigEarthNet and DeepGlobe. Following the same intuition as the earlier experiments, this analysis introduces an additional dimension by considering class-specific effects. For simplicity and clarity, we focus on the highest transformation intensity for each feature suppression. Bar plots illustrate the relative performance of each class under a specific transformation compared to the original performance, as described in Section 4.3.

6.3.1 Spectral Features Suppressed

In Figure 6.12, we observe the class-wise relative performances of models on the BigEarthNet dataset. All classes are significantly affected by transformations targeting spectral content, with no class achieving a remaining performance higher than 50%. *channel inversion* has the greatest effect, with only the mixed forest class reaching 18% performance. For *channel mean*, the "broad-leaved forest" and "natural grassland and sparsely vegetated areas" (grassland) classes achieve the highest relative performances, 48% and 46%, respectively, well above the macro average of 15%. This suggests these classes are less reliant on spectral content compared to others. The "grassland" and "broad-leaved forest" classes are also higher than the macro average for *channel shuffle* transformation, though by a smaller margin. However, for *channel inversion*, neither class ranks among the best classified. Conversely, the mixed forest class, with highest classification performances for the *channel inversion* transformation, is not well classified for *channel shuffle* or *channel mean* transformations compared to other classes. One possible explanation is that the *channel inversion* transformation may lead to inverse classification outcomes for datasets with a high reliance on spectral content. Notably, different classes see relatively higher classification performances under the three transformations, reinforcing that all classes are fundamentally reliant on spectral features to varying degrees. This highlights the central role of spectral features across the BigEarthNet dataset, not only for land cover classes previously shown to be predictable by spectral content, such as "coniferous forests" [60].

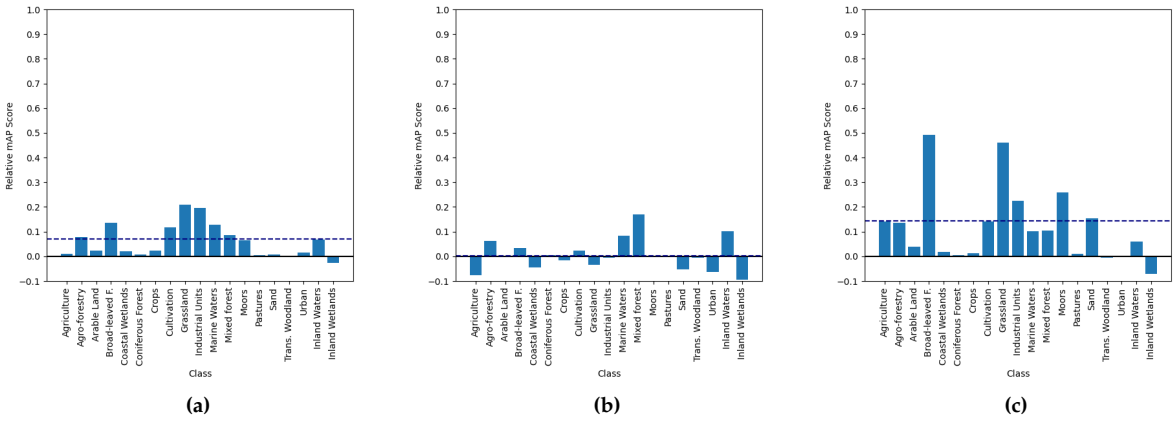


Figure 6.12: Averaged class-wise relative performances on BigEarthNet-S2 dataset for transformations affecting spectral features, **(a)** *channel shuffle*, **(b)** *channel inversion* and **(c)** *channel mean*. Transformations are applied with highest respective intensities. Relative score of 0 representing pure chance performance. Macro average shown as dottet line. Class-labels shortened. Lower values indicate higher reliance on feature type.

Analyzing the class-wise reliance on spectral features for models trained on the DeepGlobe dataset, as shown in Figure 6.13, reveals a similar trend. The "urban land" (urban) and "forest land" (forest) classes see higher classification performances by models than the macro average for both *channel shuffle* and *channel mean* transformations, but are not notably well classified under the *channel inversion* transformation. Conversely, "agricultural land" (agriculture), the best classified class under *channel inversion*, sees a lower than average classification performance for both *channel shuffle* and *channel mean* transformations applied. These results reinforce the obser-

vation that all classes in the DeepGlobe dataset are strongly reliant on spectral features, despite variations in their relative performances across different transformations. This highlights the critical importance of spectral content for classification tasks across the entire dataset.

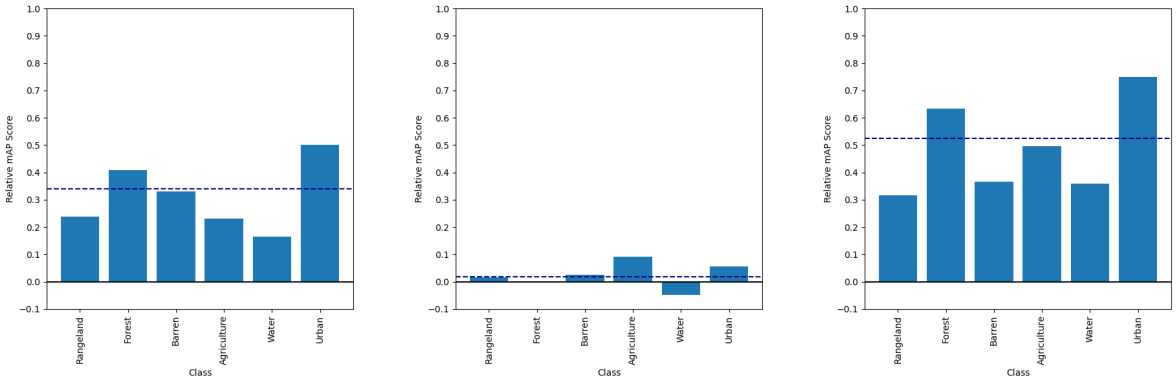


Figure 6.13: Averaged class-wise relative performances on DeepGlobe dataset for transformations affecting spectral features, **(a)** *channel shuffle*, **(b)** *channel inversion* and **(c)** *channel mean*. Transformations are applied with highest respective intensities. Relative score of 0 representing pure chance performance. Macro average shown as dottet line. Lower values indicate higher reliance on feature type.

6.3.2 Texture Features Suppressed

Figure 6.14 visualizes the class-wise relative performances of models on the BigEarthNet dataset, with texture feature suppressing transformations applied. Here, clear class-wise differences are observed, consistent across all three transformations. The "arable land" class appears severely dependent on texture features, with relative performances falling by more than 80%, showing a strong difference to other classes. Approximately half of the classes exhibit a significant reliance on texture features, with relative performance losses ranging between 40% and 60%. Conversely, several classes, including "grasslands", "Industrial or commercial units" (industrial units) and "Land principally occupied by agriculture, with significant areas of natural vegetation" (agriculture), show minimal reliance on texture features, with relative performance losses of less than 10%. These results highlight a highly varying importance of texture features across classes in the dataset.

For the the class-wise reliance on texture features for models trained on the DeepGlobe dataset, as shown in Figure 6.15, similarities across the transformations are again evident for all classes. The "urban" class is the least dependent on texture features with an performance reduction of the *bilateral filter* transformation of less than 10%, followed by "agriculture". In contrast, the "forest" class shows the highest reliance on texture features. All other classes exhibit a reliance close to the macro average, forming a middle group with moderate dependence on texture features. These results further emphasize the varying importance of texture features across classes within RS datasets.

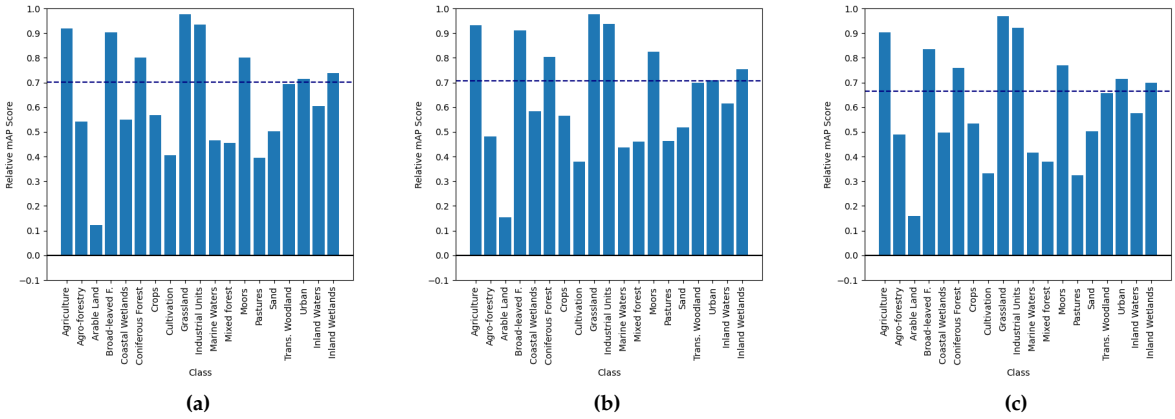


Figure 6.14: Averaged class-wise relative performances on BigEarthNet-S2 dataset for transformations affecting texture features, **(a)** *bilateral filter*, **(b)** *median filter* and **(c)** *gaussian filter*. Transformations are applied with highest respective intensities. Relative score of 0 representing pure chance performance. Macro average shown as dottet line. Class-labels shortened. Lower values indicate higher reliance on feature type.

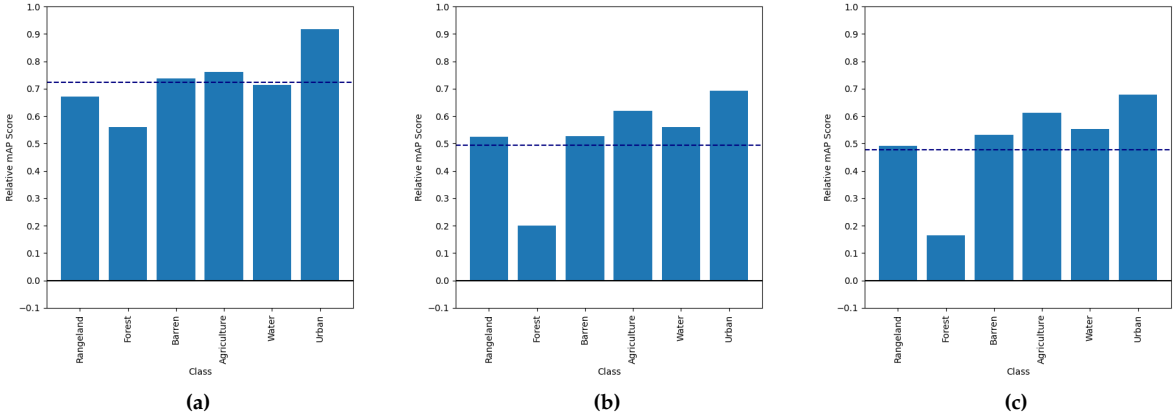


Figure 6.15: Averaged class-wise relative performances on DeepGlobe dataset for transformations affecting texture features, **(a)** *bilateral filter*, **(b)** *median filter* and **(c)** *gaussian filter*. Transformations are applied with highest respective intensities. Relative score of 0 representing pure chance performance. Macro average shown as dottet line. Lower values indicate higher reliance on feature type.

6.3.3 Shape Features Suppressed

In Figure 6.16, the class-wise relative performances of models on the BigEarthNet dataset are shown with shape features suppressed. The *patch shuffle* transformation significantly affects the "mixed forest" and "Transitional woodland, shrub" (transitional woodland) classes, resulting in a relative performance loss of approximately 50%, followed by "arable land" and "coniferous forest", each with a loss of 46%. Conversely, other classes such as "sand", "grassland", "urban", and "wetlands" experience a minimal effect, with performance losses below 10%. These results

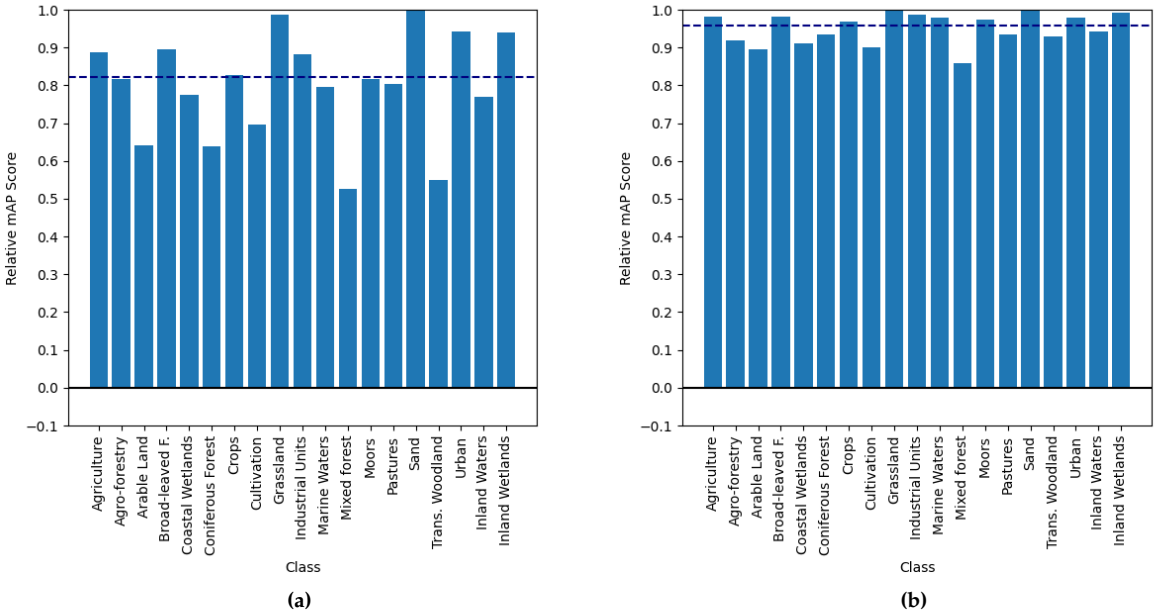


Figure 6.16: Averaged class-wise relative performances on BigEarthNet-S2 dataset for transformations affecting shape features, **(a)** *patch shuffle* and **(b)** *patch rotation*. Transformations are applied with highest respective intensities. Relative score of 0 representing pure chance performance. Macro average shown as dottet line. Class-labels shortened. Lower values indicate higher reliance on feature type.

highlight the varying reliance of different classes on shape features, for accurate classification. For the *patch rotation* transformation, the overall effect is much smaller but follows a similar pattern, with the same classes showing the highest sensitivity to the suppression of shape features.

Observing the class-wise reliance on shape features for the DeepGlobe dataset, seen in Figure 6.17, reveals a consistent trend. The *patch shuffle* transformation has a significantly stronger effect compared to other transformations, with the same classes, "rangeland" and "water", being the most affected. In contrast, models appear to be the least reliant on shape features for classification of the "agriculture" class, demonstrating low performance losses under shape-suppressing transformations.

6.3.4 Single Feature Types Remaining

When analyzing the class-wise predictive strength of single remaining feature types for the BigEarthNet dataset seen in Figure 6.18, the findings align with the results of the previous feature reliance analysis. Spectral features alone prove to be highly predictive for many classes in BigEarthNet. Notable exceptions include "arable land", with a relative performance of less than 40%, and mixed forest, with less than 50%. Conversely, several classes, such as "grassland", "agriculture", "inland wetlands", and "Beaches, dunes, sand" (sand), are almost entirely predictable by spectral features alone, achieving relative performances above 90%.

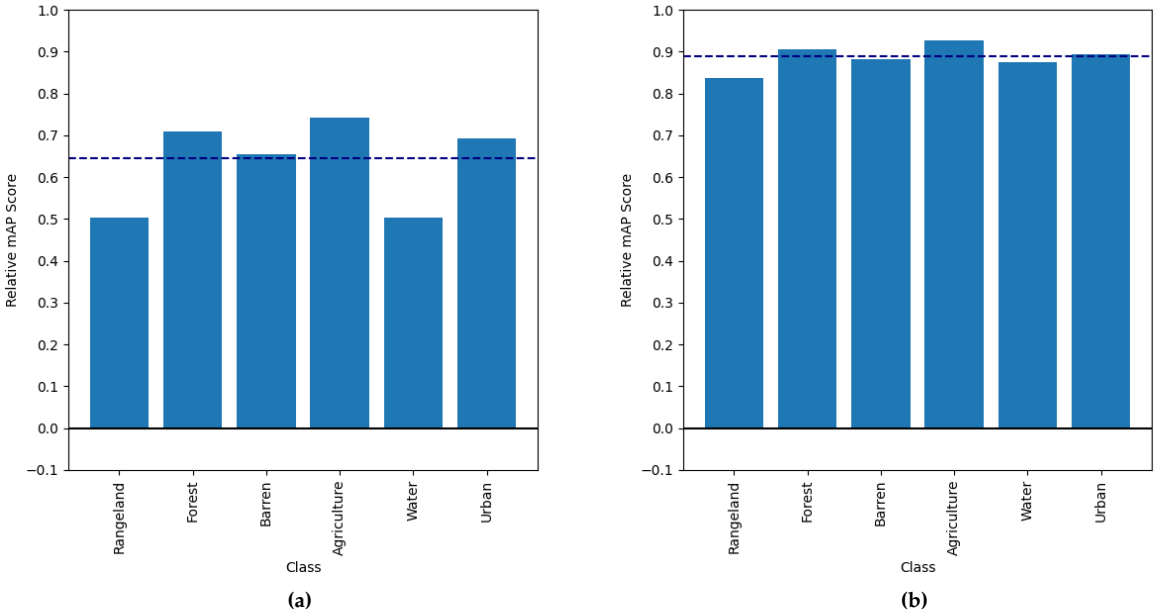


Figure 6.17: Averaged class-wise relative performances on DeepGlobe dataset for transformations affecting shape features, **(a)** *patch shuffle* and **(b)** *patch rotation*. Transformations are applied with highest respective intensities. Relative score of 0 representing pure chance performance. Macro average shown as dottet line. Lower values indicate higher reliance on feature type.

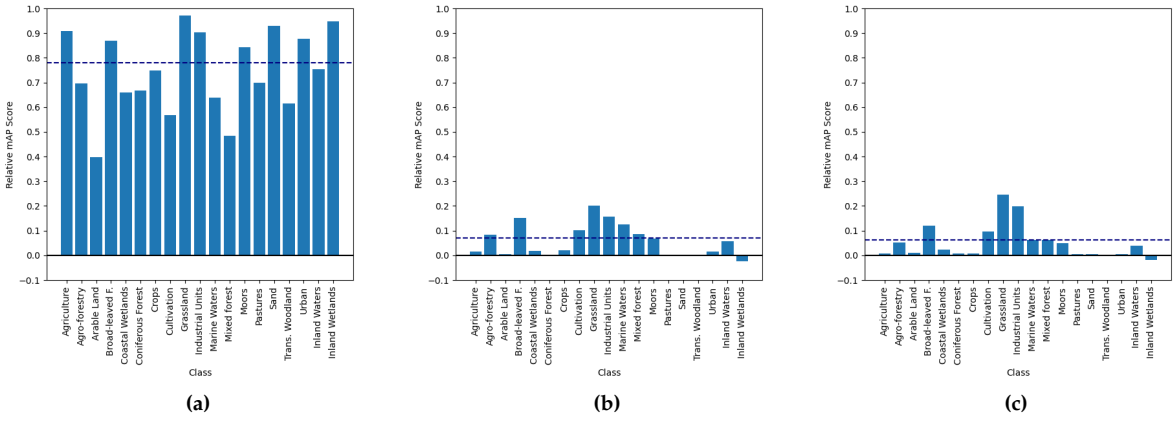


Figure 6.18: Averaged class-wise relative performances on BigEarthNet-S2 dataset for single remaining feature types **(a)** spectral features remaining, **(b)** texture features remaining and **(c)** shape features remaining. Pairwise transformations affecting both other feature categories for spectral *(channel shuffle)*, texture *(bilateral filter)* or shape features *(patch shuffle)* were used. Transformations are applied with highest respective intensities. Relative score of 0 representing pure chance performance. Macro average shown as dottet line. Class-labels shortened. Higher scores indicate higher predictive value of sole feature type.

Interestingly, "grassland" is also the best predicted class for the single remaining feature types of texture and shape. This suggests that different classes have varying requirements of feature combinations for their effective prediction. Such classes may be redundantly predictable by single feature types, highlighting a relatively low importance of feature integration for its prediction. In contrast, classes like "arable land" and "transitional woodland", which are less predictable by spectral features, also show low predictability from other feature categories individually. These classes achieve relative performance scores below the macro average for all feature categories, indicating a greater dependency on the use of multiple feature types in combination for accurate classification.

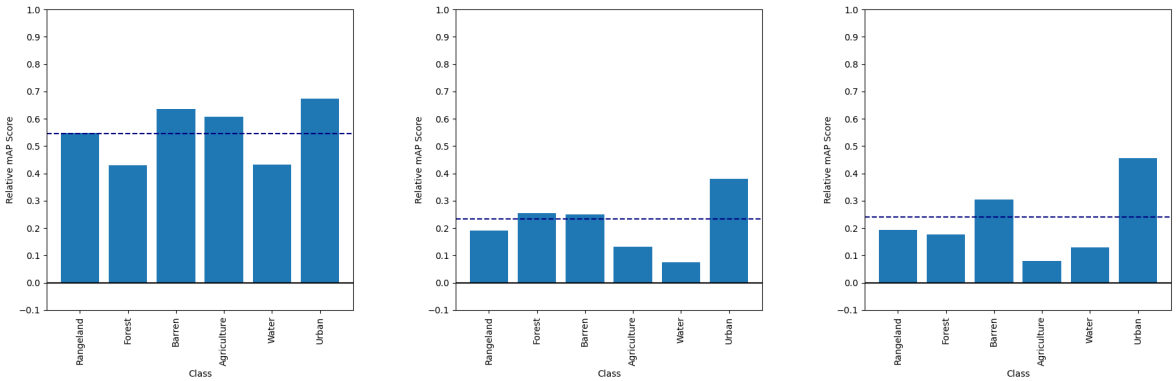


Figure 6.19: Averaged class-wise relative performances on DeepGlobe dataset for single remaining feature types **(a)** spectral features remaining, **(b)** texture features remaining and **(c)** shape features remaining. Pairwise transformations affecting both other feature categories for spectral *(channel shuffle)*, texture *(bilateral filter)* or shape features *(patch shuffle)* were used. Transformations are applied with highest respective intensities. Relative score of 0 representing pure chance performance. Macro average shown as dottet line. Class-labels shortened. Higher scores indicate higher predictive value of sole feature type.

In Figure 6.19, we analyze the predictive strength of single remaining feature categories for the DeepGlobe dataset, where differences in feature reliance on the presence of multiple features in combination are again observed. The "urban" class consistently sees a higher predictive ability of all three single feature types than the class average across, while the "water" class sees lower the average predictive ability scores for all feature types. As noted in the dataset-wide experiments, spectral features remain the strongest predictor for most classes. However, specific trends emerge among individual classes. The "barren land" (barren) class is better predicted by shape features compared to other classes, achieving a remaining relative performance of 31%. "Agriculture" and "barren" classes show stronger predictability by spectral features compared to other classes, while the "forest" class demonstrates relatively better predictability with texture features as the sole remaining feature.

Overall, we can observe moderate differences between individual classes for both datasets, when specific image features remain unaffected. The highest difference is seen for spectral features remaining, with some classes appearing highly predictable by spectral features alone while others less so. For texture and shape features, differences show individual classes being

partially predictable by the features, while other classes are entirely unpredictable by these feature types alone. However, no classes were entirely correctly predicted by texture or shape features alone. In general, our results highlight the high informative value our feature-reliance evaluation protocol offers for understanding datasets and their class-specific feature reliances in regards of what feature types define classes, and offer predictive capabilities for models to differentiate them.

6.4 Effect of Model Architectures on Feature Reliance

Finally, we examine the effect of model architecture differences on measured feature reliance for spectral, texture, and shape features. This analysis revisits the results of Section 6.1, this time separating relative performance outcomes for the two model categories across all datasets. In the following plots, averaged relative performances for transformer architectures are represented by a dotted line, while results for CNN architectures are shown as a continuous line. This comparison highlights how architectural differences influence feature reliance and provides insights into the strengths and limitations of each model type in utilizing specific image features.

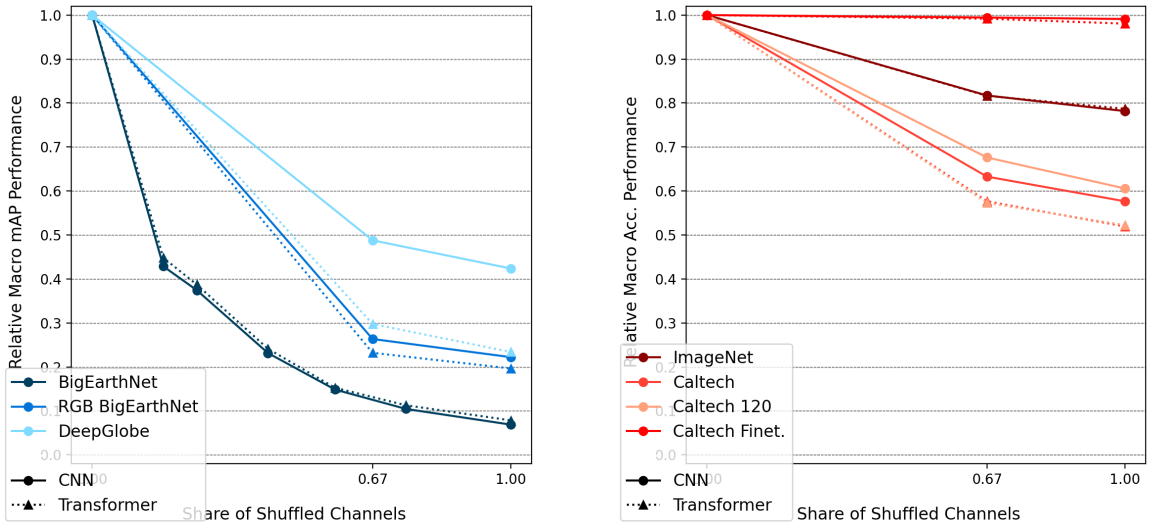


Figure 6.20: Visualization of the effect of suppressed spectral features on classification performance, for vision transformer and CNN architectures. Plots show relative performance under increasing transformation intensity of *channel shuffle* transformation, with the relative performance metric on the y-axis and the share of shuffled channels on the x-axis. Averaged relative performances shown for models of both cnn and transformer category for each dataset. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

For comparing feature reliance between CNNs and transformers with spectral features suppressed by transformation as shown in Figure 6.20, no strong differences in feature reliance are

observed across most datasets. A notable exception is found in the DeepGlobe dataset, where transformers exhibit a significantly higher reliance on spectral features. Transformers achieve a relative performance of 23%, which is nearly half of the averaged relative performance of their CNN counterparts at 42%. This shows that, for DeepGlobe, transformers depend more heavily on spectral content for accurate classification compared to CNNs.

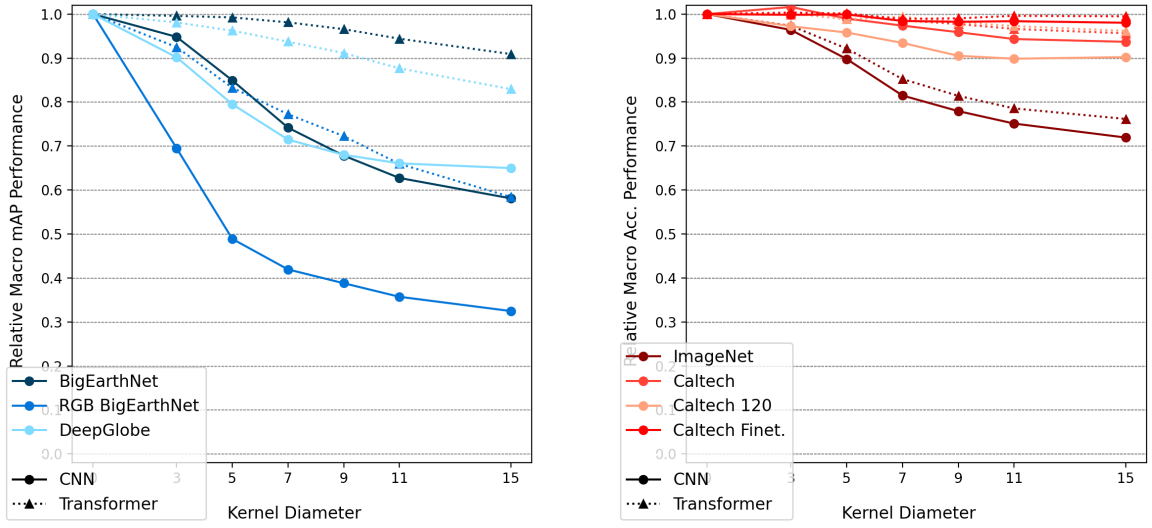


Figure 6.21: Visualization of the effect of suppressed texture features on classification performance, for vision transformer and CNN architectures. Plots show relative performance under increasing transformation intensity of *bilateral filter* transformation, with the relative performance metric on the y-axis and the kernel diameter on the x-axis. Averaged relative performances shown for models of both cnn and transformer category for each dataset. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

For comparing feature reliance between CNNs and transformers with texture features suppressed as seen in Figure 6.21, strong differences between the two architecture types are evident, particularly for datasets in the RS domain. CNN models are more strongly affected by the *bilateral filter* transformation, while transformer architectures exhibit a lesser performance reduction. This difference is most pronounced for the BigEarthNet dataset, where transformer models show an average relative performance decrease of only 9% at the highest transformation intensity, compared to CNN models, which experience an average performance loss of over 40%, four times as much. A similar trend, though less apparent, is observed for models on datasets in the CV domain, where transformers consistently outperform their CNN counterparts under texture suppression.

These findings underscore the significant difference in feature reliance on texture features between the two architecture categories, reflecting their distinct inductive biases. CNNs, due to their use of convolutional layers, exhibit a natural inductive bias toward localized texture information. In contrast, transformer architectures, which do not rely on convolutional layers, are

less dependent on texture features. This observation aligns with the findings of Mummadi et al. [20], who reported that transformer architectures are less texture-biased compared to CNNs. The reduced reliance on texture features by transformers highlights their broader flexibility in handling features beyond localized patterns.

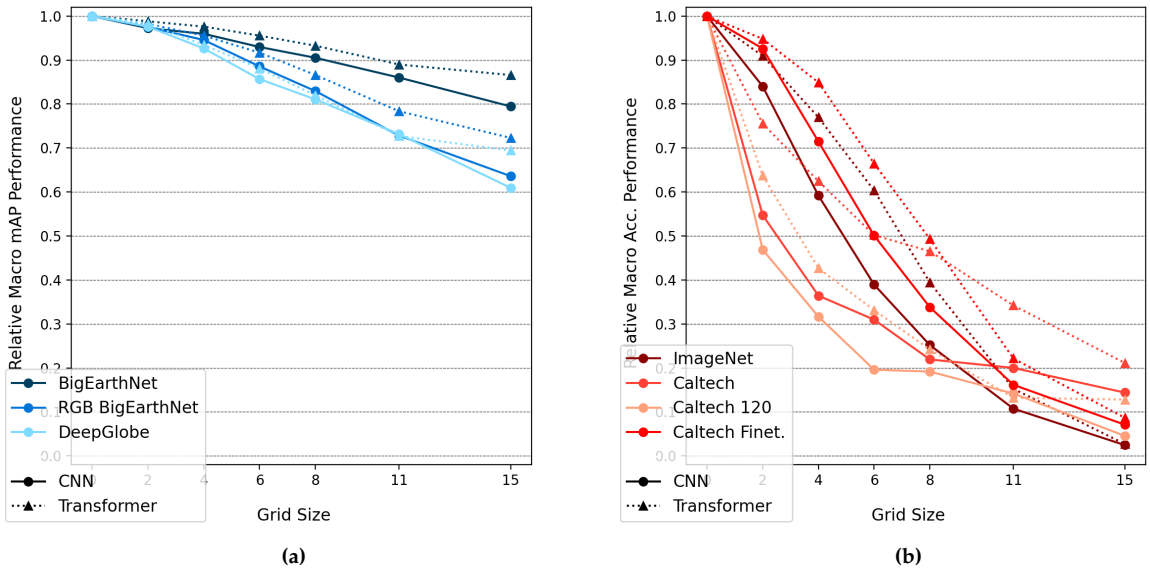


Figure 6.22: Visualization of the effect of suppressed shape features on classification performance, for vision transformer and CNN architectures. Plots show relative performance under increasing transformation intensity of *patch shuffle* transformation, with the relative performance metric on the y-axis and the share of shuffled channels on the x-axis. Averaged relative performances shown for models of both cnn and transformer category for each dataset. Datasets of **(a)** RS domain shown in in shades of blue, **(b)** CV domain shown in shades of red. Lower values indicate higher reliance on feature type.

Observing the feature reliance on shape features for both CNN and transformer models across all datasets, as seen in Figure 6.22, reveals that transformers are less reliant on the presence of such features compared to CNN architectures. Across all datasets, transformer models consistently retained higher classification performances as CNN models when shape features were suppressed. This difference may stem from the inductive bias of CNNs, which inherently account for spatial relationships between smaller features. As a result, CNNs are more affected by transformations that disrupt texture features, as both are closely tied to their architectural design. Additionally, the observed lower reliance on texture and shape features could indicate that CNN architectures rely more on a combination of features, whereas transformer architectures demonstrate a lower dependency on individual feature types, when others are still present. Transformers may benefit from their ability to model long-range dependencies between image features, enabled by their design, which places less emphasis on spatial proximity. This architectural characteristic allows transformers to maintain performance, possibly by identifying textures across the image, even when more complex spatial relationships, such as those found in shape features, are disrupted. These findings suggest that the broader contex-

tual understanding possibly inherent to transformers could provide an advantage in scenarios where spatial image features are disturbed.

7 Conclusion

This chapter summarizes the key findings from the thesis on feature bias and image feature reliance on datasets of the RS domain. The limitations of the study are discussed to highlight areas where this study may require further investigation. Finally, future research directions are proposed to guide subsequent studies in building upon the work presented here, addressing identified gaps, and exploring new opportunities to deepen the understanding of image feature importance and its implications for classification tasks in RS and beyond.

7.1 Summary of Key Findings

In this work, first, a comprehensive analysis of existing research on feature bias in DNN is conducted, with a focus on the work of Geirhos et al. [1] first reporting the phenomenon. Our investigation revealed several potential methodological and theoretical issues that could influence the reliability of their results. Methodologically, we found that the cue-conflict dataset used to evaluate feature bias may not fully separate image features as intended, possibly leading to an inherent texture-cue bias of the evaluation. Additionally, human experiments using the same cue-conflict experiment might have unintentionally biased participants toward shape-cue decisions by employing shapes as visual prompts for answers. Theoretically, we identified limitations in the research perspective of texture versus shape bias to understand the relative importance of image features. The binary evaluation approach of the cue-conflict experiment measures the relative signal strength of the feature types but does not assess their absolute predictive strength. Spectral features, were not included as relevant image features in these analyzes. Furthermore, the possibility that multiple image feature types may be required simultaneously for classification was not considered. Our findings on both methodological and theoretical issues raise important questions about the validity of the central finding of texture bias in ImageNet trained CNNs as reported by Geirhos et al. Given that the evaluation protocol employed in their work may itself be biased toward texture-cue-based decisions, the extent and nature of the reported bias in the evaluated models warrant further investigation. This highlights the need for a more robust framework to accurately assess feature reliance and bias in DNNs.

To address these shortcomings, we proposed a novel feature reliance evaluation protocol. This protocol improves upon existing methods by directly quantifying the relative reliance on image features for a given dataset. To this end, test-time transformations were employed that suppress single image feature categories. While previous methods as the cue-conflict evaluation protocol were only applicable for models trained on the CV domain, our method is entirely dataset-agnostic, allowing for a comparison of other domains to the domain of RS. Finally, our method is able to independently assess the reliance on different image feature categories and thus provides a quantitative evaluation of the relative importance of image features, without

binary comparison implicitly deducing one feature bias by measuring the opposite. This enables comparisons of relative feature importance in the RS domain against research focused predominantly on CV in DNN, making it a practical tool for analyzing individual feature importance in datasets or domains that differ from well-studied ones.

We conducted experiments using this protocol on four datasets: BigEarthNet, DeepGlobe, ImageNet, and Caltech, along with three additional experimental variation of datasets RGB-BigEarthNet, Caltech-120, and Caltech-FT to highlight the influence of specific dataset or pre-training properties. Models test performances are measured for unimpaired data as a performance baseline. Then, we evaluated feature reliance by applying test time transformations of three categories, suppressing either spectral, texture or shape features. Reliance on feature categories is evaluated by suppressing respective features in test data and measuring models performance relative to their baseline performance.

In answer to our research question about the relative importance of image features, our analysis reveals that the BigEarthNet and DeepGlobe datasets of the RS domain are highly reliant on spectral features for classification tasks. Both datasets demonstrate much lower reliance on texture and shape features, which exhibit similar levels of importance overall. This trend is particularly pronounced in models applied to the BigEarthNet dataset, where the importance of spectral features is dominant, with a relative performance loss as described in Section 4.3, of 93% averaged across 16 models when spectral features were suppressed by channel shuffling, and conversely, a remaining relative performance of 78% with only spectral features present. In contrast, models exhibited a significantly lower importance of shape and texture features, with no class prediction highly reliant on shape features, with all classes seeing remaining relative classification scores of less than 25% for only shape and less than 20% for only shape features present. Furthermore, while class-wise differences were observed to some extend, spectral features remained the most important feature category for classification of all classes of the DeepGlobe and BigEarthNet datasets. Texture features exhibited the greatest variability, with models for some classes such as "arable land" of BigEarthNet being highly reliant on texture features being present, while for others like "grassland" and "industrial units" being entirely unreliant, with relative reductions of classification performances ranging from 4% to 88%. Although a high importance of spectral features has already been established for the differentiation of certain classes, these findings highlight the overarching importance of spectral features across all classes in the RS domain, along with a more complex role of texture features and a generally lower importance of shape features for the RS domain. In addition, a clear difference in inductive bias is measured between the architectures of CNN and vision transformers, with vision transformers exhibiting a significantly lower reliance on texture features for classification performances.

Our analysis, in addition to the relative importance of image features for datasets of the RS domain, revealed clear trends in feature importance between the BigEarthNet and DeepGlobe datasets of the RS and ImageNet and Caltech dataset of the CV domain. Texture features, while exhibiting some minor utility in RS datasets, showed minimal individual importance for the CV datasets. Spectral features also were significantly less important to classification performances of models on the CV datasets than for RS. In contrast to this, models evaluated on ImageNet and Caltech datasets of the CV domain demonstrated a great reliance on shape features, the highest of all three feature categories. Differences were supported by additional experiments, measuring performance strength when features were solely present.

Our findings thereby offer multiple highly significant additional insights. Firstly, they highlight a possible substantial difference in the relative importance of image characteristics between data sets in the RS and CV domains, not only for spectral but also for shape features. Understanding this strong difference in importance of spectral and shape features between domains indicates potential issues with applying models pretrained on the CV domain to downstream tasks in the RS domain, as such models might have learned to rely on shape features significantly less present in such a downstream task. Combined with the understanding of Hermann and Lampinen [26] that image features may suppress others from being learned, pretraining on image domains showing similar feature importance characteristics could improve overall classification performances.

Secondly, they demonstrate a considerably different picture of feature importance than the presented results of Geirhos et al.'s reported texture bias [1]. Our results show that when separating spectral content as an individual feature category, models trained on the ImageNet dataset see a significantly lower utility of texture features than of shape features. Explicitly, with texture features suppressed by a *bilateral filter* transformation, models relative classification performance is reduced by only 30%, while the suppression of shape features in images by patch-shuffling reduced relative performance scores by 97%. This indicates that DNN are in fact, not texture biased, but as previously assumed, highly reliant on shape features for their interpretation of objects, especially for datasets with a high availability of shape features, such as ImageNet. These findings challenge the interpretation of Geirhos et al., who concluded that texture features are a more significant feature category than shape features. Our findings thus show that, while previous findings that global shape features may be of lower importance [3, 2] are corroborated, shape features as a whole remain highly important. Therefore, our results highlight the significant importance of local shape features, not specifically part of an objects outline, for classification performances.

7.2 Limitations

While our proposed evaluation protocol appears highly valuable to understand image feature reliance across data domains, and results show an important perspective on image feature reliance, there are limitations to our findings.

Although the chosen data sets, particularly ImageNet for CV and BigEarthNet for RS, are widely used and representative within their respective domains, they are not fully comprehensive. The analysis included only two datasets from each domain, limiting the generalizability of the findings. For the RS domain, BigEarthNet and DeepGlobe do not represent the full spectrum of potential shape features. Although DeepGlobe provides very high-resolution imagery, which could theoretically include more shape features, its six-class structure likely diminishes the need for shape information to distinguish between classes. Other RS datasets, such as those based on aerial photography with larger nomenclatures, may exhibit a greater reliance on shape features due to the need to differentiate a wider variety of closely related classes. Furthermore, while efforts were made to adapt additional dataset variations to ensure similarity, significant property differences remain. For instance, the RS datasets were multi-label, which could behave differently compared to the multi-class classification tasks predominantly used in the CV domain. These factors highlight the need for caution when generalizing findings across domains and the importance of further studies incorporating a broader range of datasets with

diverse properties.

Additionally, while we carefully selected transformations for each feature category to suppress, and in Section 6.2 focused on those identified through qualitative analysis as least affecting other feature categories, we cannot fully guarantee the absence of unintended overlap. For instance, texture features may be unintentionally degraded by shape-targeting transformations such as patch shuffle, which can disrupt texture patterns as a secondary effect. This overlap reduces the effectiveness of isolating individual features during evaluation. And, the feature suppression achieved by these transformations may not fully replicate the natural degradation of these features in real-world scenarios. This limitation underscores the challenges of simulating feature-specific transformations.

Finally, our findings on image feature reliance and predictive strength are limited to classification tasks and may not fully generalize to other image-related tasks, such as segmentation or fine-grained classification. Feature reliance and predictive strength could differ in these contexts, where the importance of specific features might vary depending on the task requirements. As of now, our analysis has exclusively focused on image classification, leaving the applicability of our conclusions to other tasks as an open area for future research.

7.3 Future Research Opportunities

As our proposed feature reliance evaluation protocol evaluates properties core to the understanding of the interaction of models trained on image data, we see the potential for the following directions for consequent studies:

Future research offers several avenues to expand and refine the understanding of feature reliance across datasets, tasks, and domains. One promising direction is the inclusion of a broader range of RS datasets with diverse characteristics, to further generalize our findings across the domain. For example, datasets with larger class nomenclatures and a high spectral resolution, such as those used in detailed land cover classification or aerial photography, could highlight the role of texture and shape features when finer distinctions between classes are needed. Incorporating hyperspectral imagery would provide insight into how extensive spectral content influences feature reliance on spectral features, particularly when compared to datasets limited to RGB channels.

The development of more precise feature suppression transformations is another important area for future work. Current transformations might unintentionally overlap in their influence on other feature categories, reducing the clarity of the results. Designing additional specifically feature targeting transformations could further enhance the accuracy of feature isolation and thereby assessments of feature importance.

The effects of pre-training on feature reliance also deserve a deeper exploration. Comparing models pre-trained on general-purpose datasets like ImageNet with those pre-trained on domain-specific datasets could reveal how pre-training shapes feature utilization. For the RS domain, where spectral content is critical, such studies could offer practical guidance on selecting or designing pre-training datasets and strategies to optimize performance. Here, cross-domain transferability represents another compelling research avenue. Understanding how feature reliance changes when models are transferred between domains, such as from CV to RS, could illuminate fundamental differences in feature importance. This includes examining

whether pretraining on CV datasets enhances or hinders the ability to adapt to RS tasks or whether RS-specific training provides advantages in other applications.

The interactions between different image features, such as texture and spectral content, also merit further study. Features rarely function independently and their combined influence may vary between datasets and tasks. For instance, texture and spectral content might act synergistically in some datasets while competing in others. Exploring these relationships would provide a more comprehensive understanding of feature reliance and its effect on classification performance.

Furthermore, future research can explore the role of data augmentation in shaping feature reliance. Systematic studies on the effects of augmentations that emphasize or suppress specific features could provide insights into how these techniques influence model training and generalization on specific datasets.

Expanding the feature-bias evaluation protocol to other domains, such as medical imaging or autonomous driving, would provide a broader perspective on feature importance. Different domains have unique challenges and feature distributions, and applying the protocol could help quantify domain-specific trends and needs. Such research would not only validate the protocol's versatility, but also inform the design of models tailored to specific applications.

Finally, extending the protocol beyond classification tasks could uncover new insights into feature reliance in tasks such as object segmentation. Such tasks often rely on features differently than classification tasks, and analyzing them could highlight new patterns of feature importance. By exploring these diverse opportunities, future research can refine understanding of the importance of different image features, vital to improving training design, and thereby enhancing performances of models in the RS domain and beyond.

Acronyms

BCE binary cross entropy. 41

BigEarthNet BigEarthNet-S2. xi, xii, 2, 3, 37, 39, 43–52, 54–66, 69, 74, 75, 87, 91–96

Caltech-FT Caltech-Finetuned. 39, 43, 45–50, 52–55, 57–61, 74, 90

Caltech101 Caltech101. xi, 39

CE cross entropy. 41

CNN convolutional neural network. xii, 1, 2, 5, 6, 12, 15, 17, 20, 39, 40, 68–70, 73, 74

CV computer vision. 1–3, 8, 9, 18, 37–40, 43–51, 53–61, 68–70, 73–77

DeepGlobe DeepGlobe. xi, xii, 2, 3, 9, 22–27, 29, 30, 32–34, 37–39, 43–45, 47–50, 52, 54–57, 59–67, 69, 74, 75, 87

DeepGlobe-LCCC DeepGlobe Land Cover Classification Challenge. 38

DNN deep neural networks. 1, 5, 7, 55, 73–75

ILSVRC ImageNet Large Scale Visual Recognition Challenge. 38

ImageNet ImageNet-1K. xi, 2, 37–39, 43–50, 52, 54–56, 58–61, 73–75, 88, 91–96

LR learning rate. 41

RGB-BigEarthNet RGB-BigEarthNet-S2. 39, 43–45, 47–52, 54–61, 74, 89

RS remote sensing. 1–3, 6–9, 11, 18, 20, 37, 39, 41, 43–61, 63, 68–70, 73–77

ViT vision transformer. 2, 40

Bibliography

[1] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," in *International Conference on Learning Representations*, 2018.

[2] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman, "Deep convolutional networks do not classify based on global object shape," *PLoS computational biology*, vol. 14, no. 12, p. e1006613, 2018.

[3] W. Brendel and M. Bethge, "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet," in *International Conference on Learning Representations*, 2018.

[4] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS journal of photogrammetry and remote sensing*, vol. 117, pp. 11–28, 2016.

[5] S. Illarionova, D. Shadrin, A. Trekin, V. Ignatiev, and I. Oseledets, "Generation of the nir spectral band for satellite images with convolutional neural networks," *Sensors*, vol. 21, no. 16, p. 5646, 2021.

[6] S. Gui, S. Song, R. Qin, and Y. Tang, "Remote sensing object detection in the deep learning era—a review," *Remote Sensing*, vol. 16, no. 2, p. 327, 2024.

[7] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[8] B. Landau, L. B. Smith, and S. S. Jones, "The importance of shape in early lexical learning," *Cognitive development*, vol. 3, no. 3, pp. 299–321, 1988.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[10] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[11] K. N. Clasen, L. W. Hackel, T. Burgert, G. Sumbul, B. Demir, and V. Markl, "reben: Refined bigearthnet dataset for remote sensing image analysis," *CoRR*, 2024.

[12] T. Burgert, T. Siebert, K. N. Clasen, and B. Demir, "A label propagation strategy for cutmix in multi-label remote sensing image classification," *arXiv preprint arXiv:2405.13451*, 2024.

[13] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *2004 conference on computer vision and pattern recognition workshop*. IEEE, 2004, pp. 178–178.

[14] N. Kriegeskorte, "Deep neural networks: a new framework for modeling biological vision and brain information processing," *Annual review of vision science*, vol. 1, no. 1, pp. 417–446, 2015.

[15] S. Ritter, D. G. Barrett, A. Santoro, and M. M. Botvinick, "Cognitive psychology for deep neural networks: A shape bias case study," in *International conference on machine learning*. PMLR, 2017, pp. 2940–2949.

[16] C. F. Cadieu, H. Hong, D. L. Yamins, N. Pinto, D. Ardila, E. A. Solomon, N. J. Majaj, and J. J. DiCarlo, "Deep neural networks rival the representation of primate it cortex for core visual object recognition," *PLoS computational biology*, vol. 10, no. 12, p. e1003963, 2014.

[17] D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo, "Performance-optimized hierarchical models predict neural responses in higher visual cortex," *Proceedings of the national academy of sciences*, vol. 111, no. 23, pp. 8619–8624, 2014.

[18] M. Zeiler, "Visualizing and understanding convolutional networks," in *European conference on computer vision/arXiv*, vol. 1311, 2014.

[19] K. Hermann, T. Chen, and S. Kornblith, "The origins and prevalence of texture bias in convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 000–19 015, 2020.

[20] C. K. Mummadi, R. Subramaniam, R. Hutmacher, J. Vitay, V. Fischer, and J. H. Metzen, "Does enhanced shape bias improve neural network robustness to common corruptions?" in *International Conference on Learning Representations*, 2021.

[21] H. Chung and K. H. Park, "Shape prior is not all you need: Discovering balance between texture and shape bias in cnn," in *Proceedings of the Asian Conference on Computer Vision*, 2022, pp. 4160–4175.

[22] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. Shahbaz Khan, and M.-H. Yang, "Intriguing properties of vision transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 296–23 308, 2021.

[23] J. Tang, Z. Zhang, L. Zhao, and P. Tang, "Increasing shape bias to improve the precision of center pivot irrigation system detection," *Remote Sensing*, vol. 13, no. 4, p. 612, 2021.

[24] N. Kalischek, R. C. Daudt, T. Peters, R. Furrer, J. D. Wegner, and K. Schindler, "Biasbed-rigorous texture bias evaluation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 221–22 230.

[25] Y. Ge, Y. Xiao, Z. Xu, X. Wang, and L. Itti, "Contributions of shape, texture, and color in visual recognition," in *European Conference on Computer Vision*. Springer, 2022, pp. 369–386.

[26] K. Hermann and A. Lampinen, "What shapes feature representations? exploring datasets, architectures, and training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9995–10 006, 2020.

[27] T. P. Wallace, O. R. Mitchell, and K. Fukunaga, "Three-dimensional shape analysis using local shape descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 3, pp. 310–323, 1981.

[28] L. Armi and S. Fekri-Ershad, "Texture image analysis and texture classification methods-a review," *arXiv preprint arXiv:1904.06554*, 2019.

[29] J. C. Russ, *The image processing handbook*. CRC press, 2006.

[30] R. Gens, "Spectral information content of remote sensing imagery," in *Geospatial technology for earth observation*. Springer, 2009, pp. 177–201.

[31] M. A. Islam, M. Kowal, P. Esser, S. Jia, B. Ommer, K. G. Derpanis, and N. Bruce, "Shape or texture: Understanding discriminative features in cnns," in *International Conference on Learning Representations*, 2021.

[32] S. Jain, D. Tsipras, and A. Madry, "Combining diverse feature priors," in *International Conference on Machine Learning*. PMLR, 2022, pp. 9802–9832.

[33] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[34] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[35] R. Geirhos, "rgeirhos/generalisation-humans-DNNs," Mar. 2024, original-date: 2018-05-23T09:02:52Z. [Online]. Available: https://github.com/rgeirhos/generalisation-humans-DNNs

[36] A. Varde, E. Rundensteiner, G. Javidi, E. Sheybani, and J. Liang, "Learning the relative importance of features in image data," in *2007 IEEE 23rd international conference on data engineering workshop*. IEEE, 2007, pp. 237–244.

[37] J. Wang, R. Du, D. Chang, K. Liang, and Z. Ma, "Domain generalization via frequency-domain-based feature disentanglement and interaction," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 4821–4829.

[38] X. Yu, D. Lu, X. Jiang, G. Li, Y. Chen, D. Li, and E. Chen, "Examining the roles of spectral, spatial, and topographic features in improving land-cover and forest classifications in a subtropical region," *Remote Sensing*, vol. 12, no. 18, p. 2907, 2020.

[39] N. Cohen and A. Shashua, "Inductive bias of deep convolutional networks through pooling geometry," in *International Conference on Learning Representations*, 2022.

[40] Z. Wang and L. Wu, "Theoretical analysis of the inductive biases in deep convolutional networks," *Advances in Neural Information Processing Systems*, vol. 36, pp. 74 289–74 338, 2023.

[41] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*. IEEE, 1998, pp. 839–846.

[42] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar, "Deepglobe 2018: A challenge to parse the earth through satellite images," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 172–181.

[43] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[44] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[45] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer*

vision and pattern recognition, 2015, pp. 1–9.

[47] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[48] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 428–10 436.

[49] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 976–11 986.

[50] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[51] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1314–1324.

[52] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[53] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*. PMLR, 2021, pp. 10 347–10 357.

[54] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 32–42.

[55] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 568–578.

[56] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 11 936–11 945.

[57] A. Trockman and J. Z. Kolter, "Patches are all you need?" *arXiv preprint arXiv:2201.09792*, 2022.

[58] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[59] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2022.

[60] S. M. Zraenko, "Combining channels to increase the differences between coniferous and hardwood vegetation in satellite images," in *2021 International Conference on Information Technology and Nanotechnology (ITNT)*. IEEE, 2021, pp. 1–4.

Appendix

1 Individual Model Test Performances

Original mAP macro test performance of models on the BigEarthNet-S2 dataset [11] with no transformations applied.

Model	mAP macro Test Performance
CaiT	0.631
ConvMixer	0.654
ConvNeXt	0.641
DeiT	0.625
DenseNet	0.659
EfficientNet	0.610
Inception	0.629
MobileNetV3	0.619
PiT	0.626
PVT	0.637
RegNetX	0.653
RegNetY	0.638
ResNet	0.664
ResNeXt	0.672
ViT	0.643
Xception	0.649

Original mAP macro test performance of models on the DeepGlobe dataset [12] with no transformations applied.

Model	mAP macro Test Performance
BeiT	0.713
CaiT	0.764
ConvMixer	0.774
ConvNeXt	0.687
DeiT	0.767
DenseNet	0.810
EfficientNet	0.624
Inception	0.758
MobileNetV3	0.612
PiT	0.771
PVT	0.736
RegNetX	0.756
RegNetY	0.761
ResNet	0.780
ResNeXt	0.774
ViT	0.793
Xception	0.748

Original Top-1 Acc. macro test performance of models on the ImageNet dataset [9] with no transformations applied.

Model	Top-1 Acc. macro Test Performance
CaiT	0.847
ConvMixer	0.828
ConvNeXt	0.859
DeiT	0.643
DenseNet	0.802
EfficientNet	0.812
Inception	0.695
MobileNetV3	0.826
PiT	0.664
PVT	0.855
RegNetX	0.823
RegNetY	0.736
ResNet	0.877
ResNeXt	0.891
Xception	0.632

Original Top-1 Acc. macro test performance of models on the Caltech dataset [13] with no transformations applied.

Model	Top-1 Acc. macro Test Performance
BeiT	0.531
CaiT	0.503
ConvMixer	0.666
DeiT	0.528
DenseNet	0.795
EfficientNet	0.468
Inception	0.658
MobileNetV3	0.455
PiT	0.631
PVT	0.628
RegNetX	0.577
RegNetY	0.521
ResNet	0.659
ResNeXt	0.657
ViT	0.528
Xception	0.769

Original mAP macro test performance of models on the RGB-BigEarthNet dataset with no transformations applied.

Model	mAP macro Test Performance
CaiT	0.566
ConvMixer	0.613
ConvNeXt	0.571
DeiT	0.572
DenseNet	0.609
EfficientNet	0.572
Inception	0.600
MobileNetV3	0.584
PiT	0.566
PVT	0.602
RegNetX	0.618
RegNetY	0.609
ResNet	0.615
ResNeXt	0.624
ViT	0.578
Xception	0.615

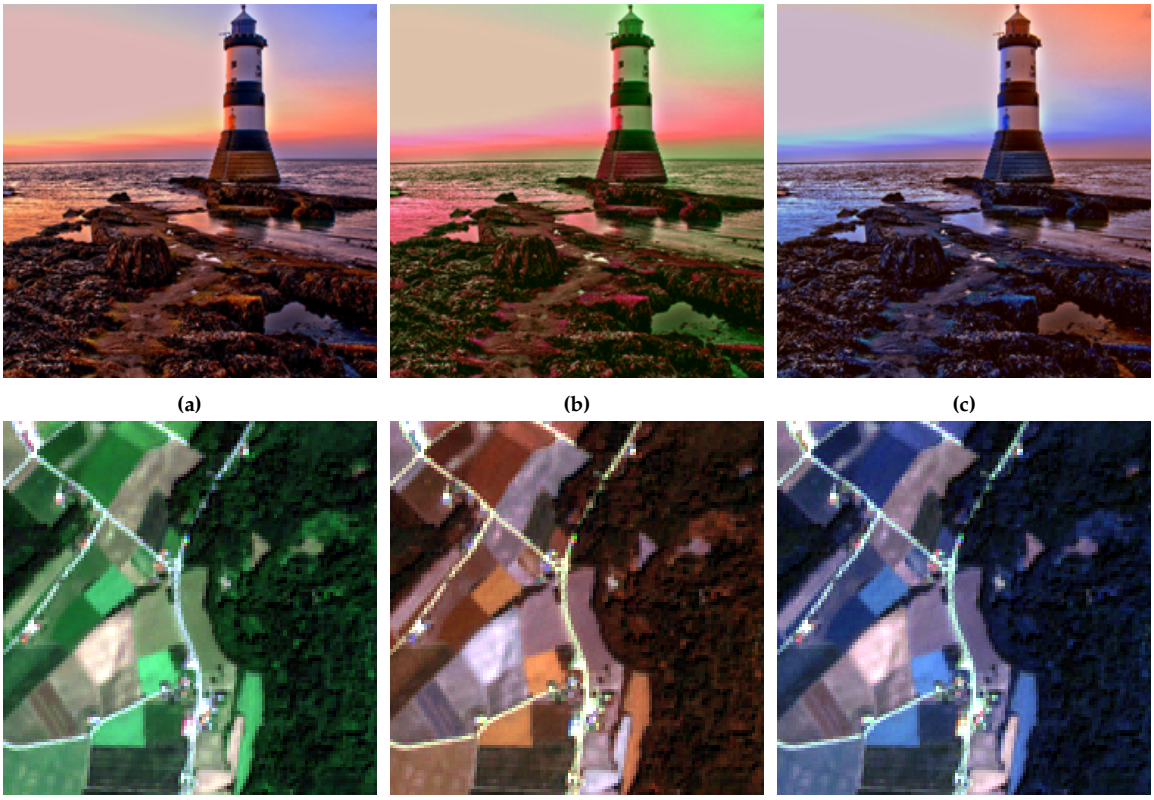
Original Top-1 Acc. macro test performance of models on the Caltech-120 dataset with no transformations applied.

Model	Top-1 Acc. macro Test Performance
CaiT	0.497
ConvMixer	0.455
DeiT	0.490
DenseNet	0.787
EfficientNet	0.301
Inception	0.572
MobileNetV3	0.406
PiT	0.595
PVT	0.667
RegNetX	0.535
RegNetY	0.492
ResNet	0.680
ResNeXt	0.673
ViT	0.539
Xception	0.719

Original Top-1 Acc. macro test performance of models on the Caltech-FT dataset with no transformations applied.

Model	Top-1 Acc. macro Test Performance
CaiT	0.986
ConvMixer	0.982
ConvNeXt	0.990
DeiT	0.990
DenseNet	0.990
EfficientNet	0.990
Inception	0.975
MobileNetV3	0.967
PiT	0.970
PVT	0.986
RegNetX	0.984
RegNetY	0.984
ResNet	0.980
ResNeXt	0.988
ViT	0.965
Xception	0.990

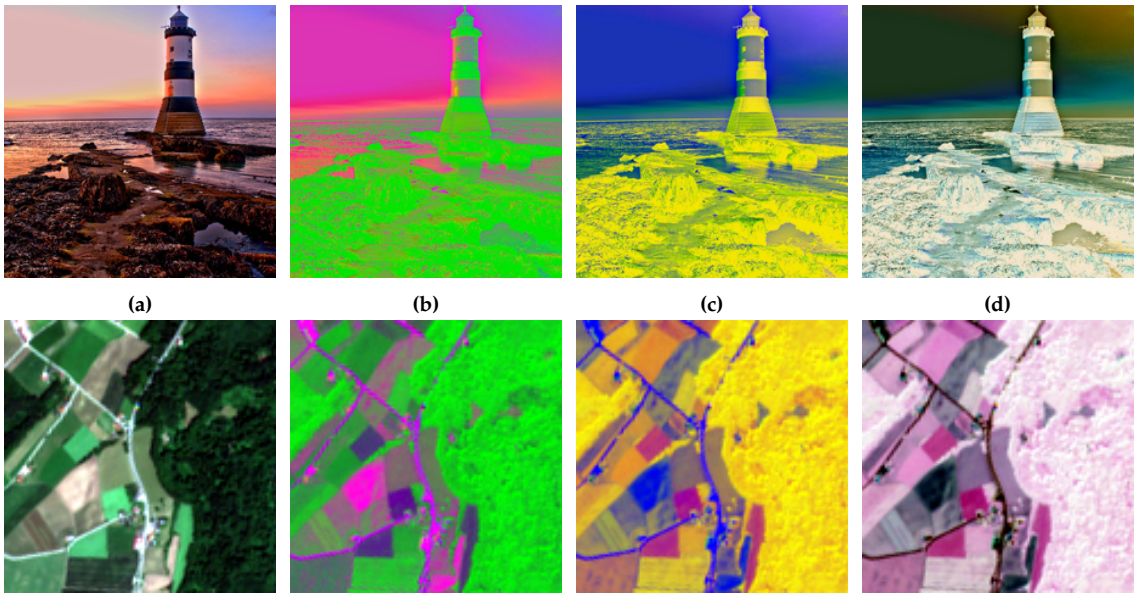
2 Additional Transformation Examples



(a) (b) (c)

(d) (e) (f)

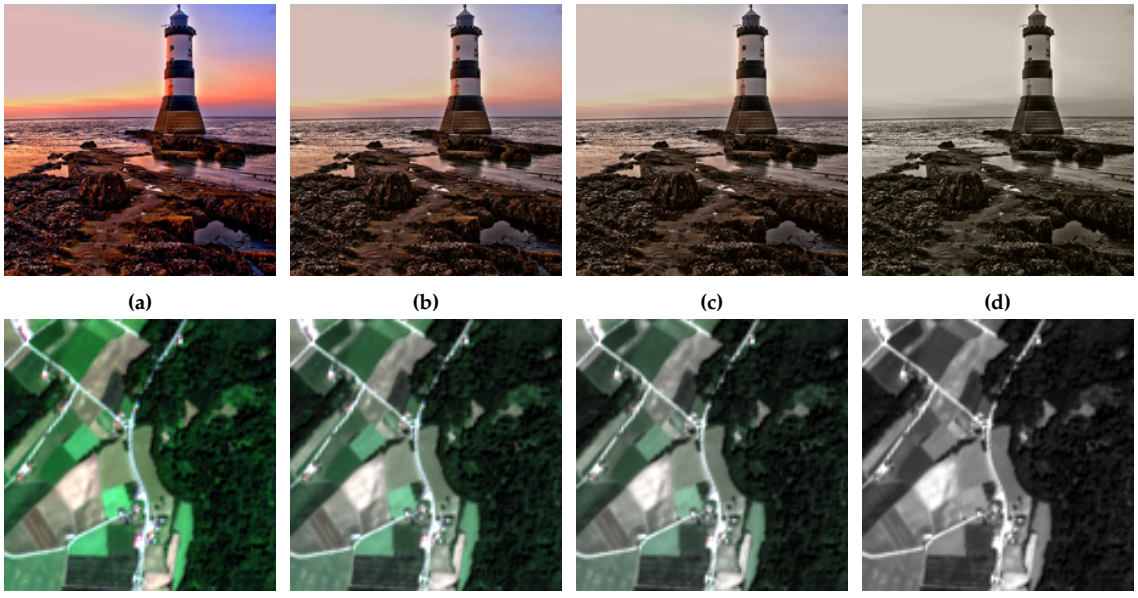
Examples of channel shuffle transformation applied to images of the ImageNet dataset [9] shown in first row, and BigEarthNet-S2 dataset [11] shown in second row. Effect of transformation is shown for different shares of channels shuffled, with **(a,d)** no channels shuffled, **(b,e)** 2/3 channels shuffled and **(c,f)** all channels shuffled . For the BigEarthNet dataset, RGB channels are visualized.



(a) (b) (c) (d)

(e) (f) (g) (h)

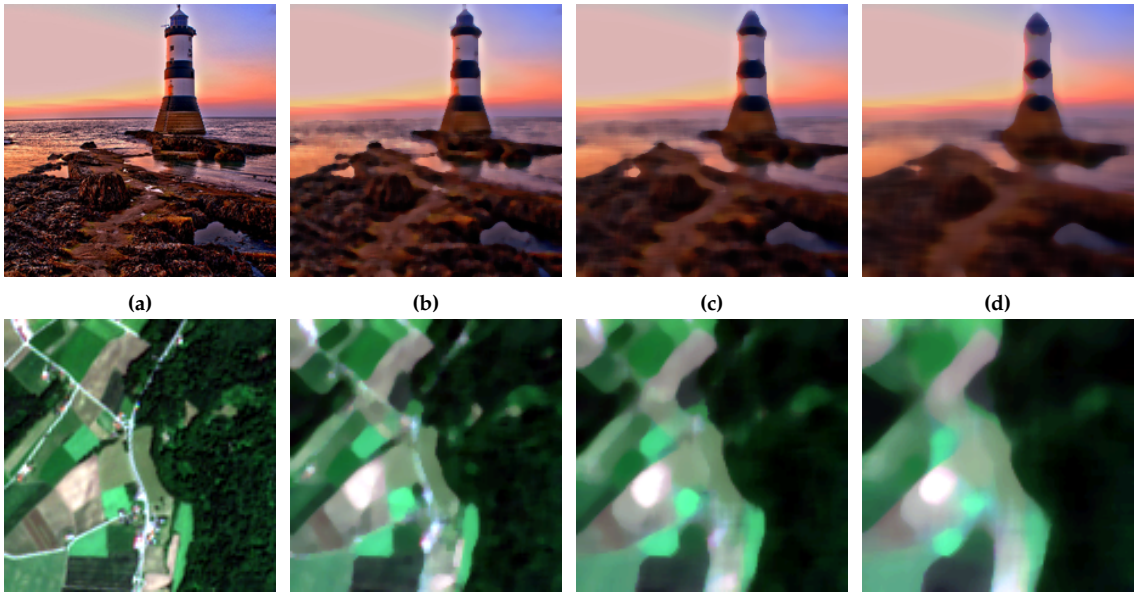
Examples of channel inversion transformation applied to images of the ImageNet dataset [9] shown in first row, and BigEarthNet-S2 dataset [11] shown in second row. Effect of transformation is shown for different shares of channels inverted, with **(a,e)** no channels inverted, **(b,f)** 1/3 channels inverted, **(c,g)** 2/3 channels inverted and **(d,h)** all channels inverted. For the BigEarthNet dataset, RGB channels are visualized.



(a) (b) (c) (d)

(e) (f) (g) (h)

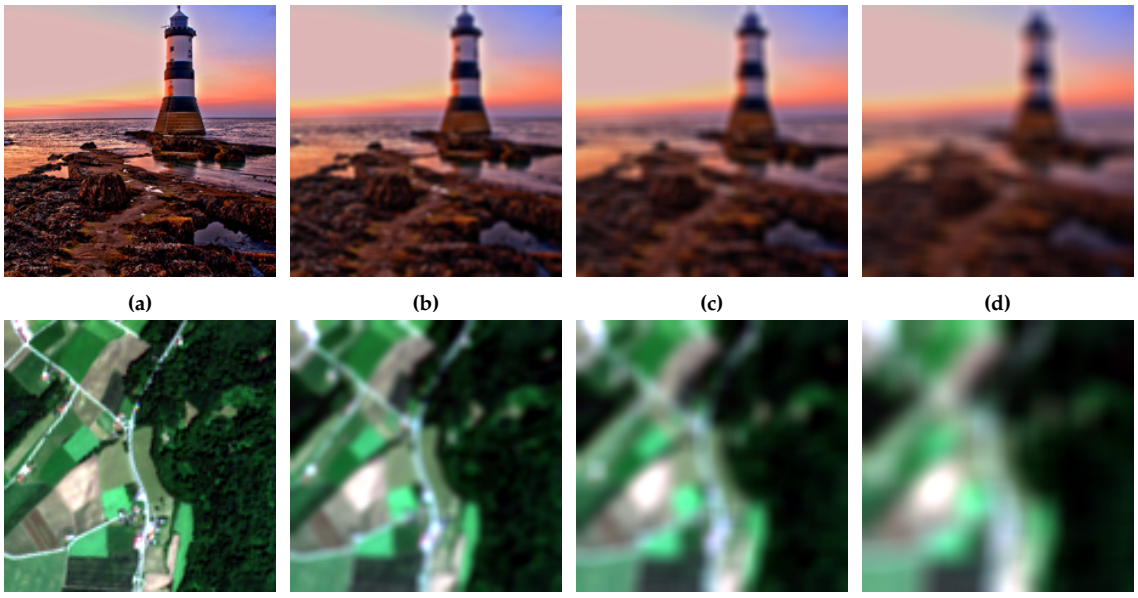
Examples of channel mean transformation applied to images of the ImageNet dataset [9] shown in first row, and BigEarthNet-S2 dataset [11] shown in second row. Effect of transformation is shown for different values of intensity parameter p, with **(a,e)** no channels averaging, **(b,f)** channels mean replacing channels by 50%, **(c,g)** channels mean replacing channels by 70% and **(d,h)** all channels replaced by their mean. For the BigEarthNet dataset, RGB channels are visualized.



(a) (b) (c) (d)

(e) (f) (g) (h)

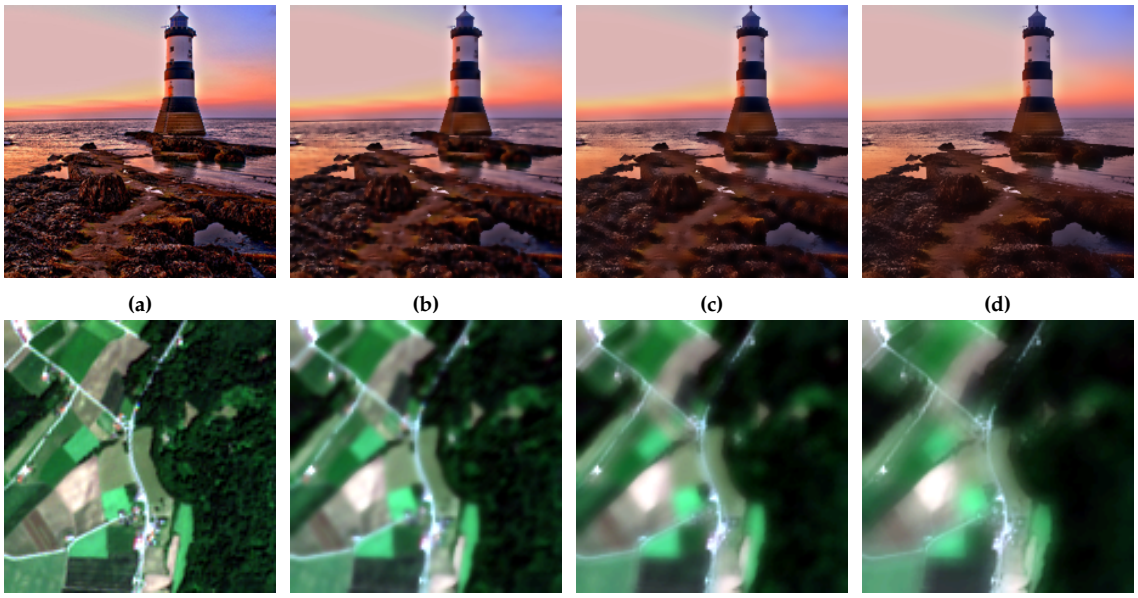
Examples of median filter transformation applied to images of the ImageNet dataset [9] shown in first row, and BigEarthNet-S2 dataset [11] shown in second row. Effect of transformation is shown for different kernel sizes, with **(a,e)** no median filtering applied, **(b,f)** median filtering with a kernel size of 5, **(c,g)** median filtering with a kernel size of 9 and **(d,h)** median filtering with a kernel size of 15. For the BigEarthNet dataset, RGB channels are visualized.



(a) (b) (c) (d)

(e) (f) (g) (h)

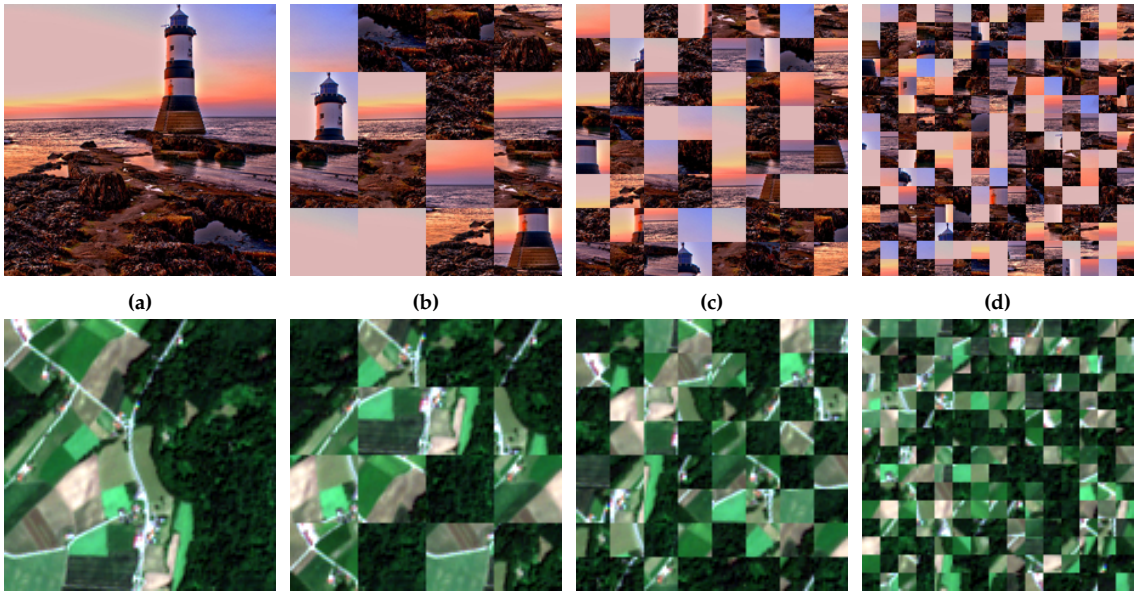
Examples of Gaussian Filter transformation applied to images of the ImageNet dataset [9] shown in first row, and BigEarthNet-S2 dataset [11] shown in second row. Effect of transformation is shown for different σ values, with **(a,e)** no Gaussian Filter applied, **(b,f)** Gaussian Filter with a σ value of 2, **(c,g)** Gaussian Filter with a σ value of 4 and **(d,h)** Gaussian Filter with a σ value of 7. For the BigEarthNet dataset, RGB channels are visualized.



(a) (b) (c) (d)

(e) (f) (g) (h)

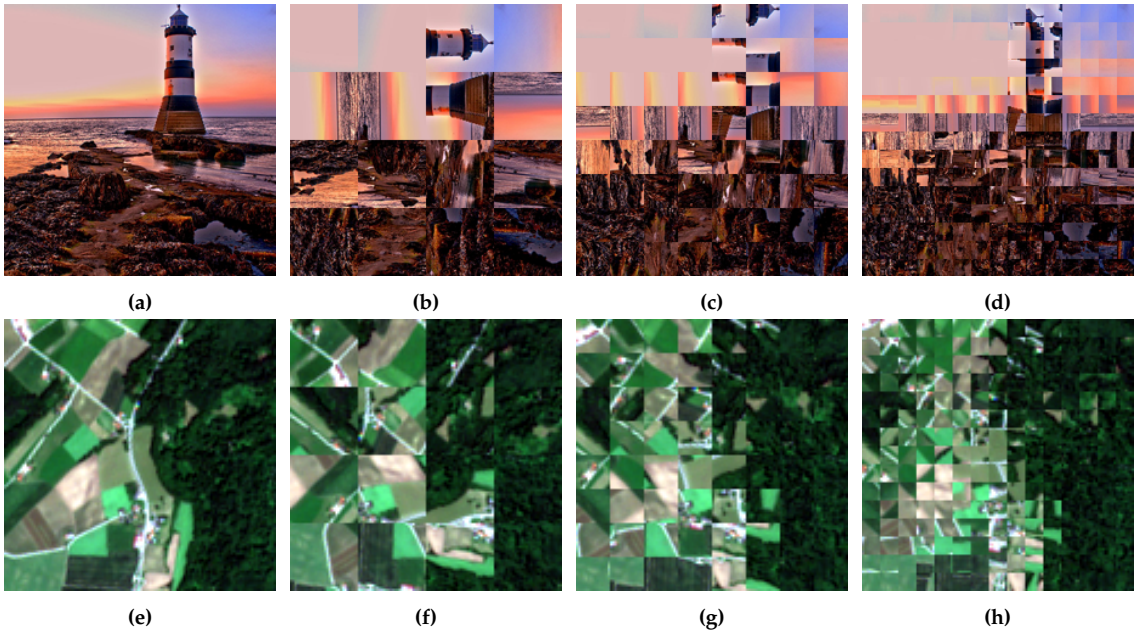
Examples of bilateral filter transformation applied to images of the ImageNet dataset [9] shown in first row, and BigEarthNet-S2 dataset [11] shown in second row. Effect of transformation is shown for different kernel diameters, with **(a,e)** no bilateral filter applied, **(b,f)** bilateral filter with a kernel diameter of 5, **(c,g)** bilateral filter with a kernel diameter of 9, and **(d,h)** bilateral filter with a kernel diameter of 15. For the BigEarthNet dataset, RGB channels are visualized.



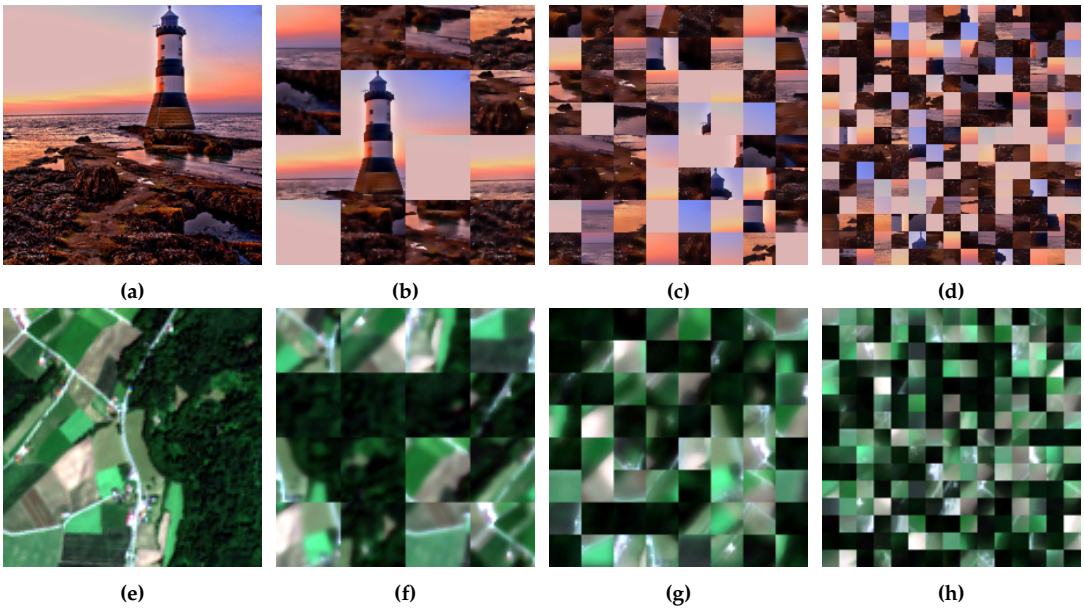
(a) (b) (c) (d)

(e) (f) (g) (h)

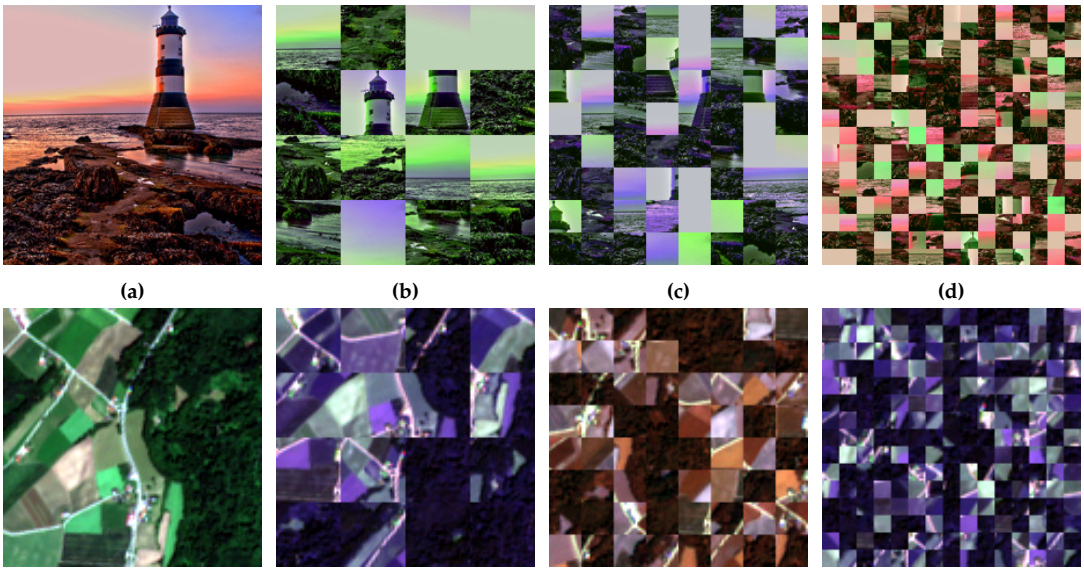
Examples of patch shuffle transformation applied to images of the ImageNet dataset [9] shown in first row, and BigEarthNet-S2 dataset [11] shown in second row. Effect of transformation is shown for different grid sizes, with **(a,e)** no patch shuffle applied, **(b,f)** patch shuffle with a grid sizes of 5, **(c,g)** patch shuffle with a grid sizes of 9, **(d,h)** and patch shuffle with a grid sizes of 15. For the BigEarthNet dataset, RGB channels are visualized.



Examples of patch rotation transformation applied to images of the ImageNet dataset [9] shown in first row, and BigEarthNet-S2 dataset [11] shown in second row. Effect of transformation is shown for different grid sizes, with **(a,e)** no patch rotation applied, **(b,f)** patch rotation with a grid sizes of 5, **(c,g)** patch rotation with a grid sizes of 9, and **(d,h)** patch rotation with a grid sizes of 15. For the BigEarthNet dataset, RGB channels are visualized.



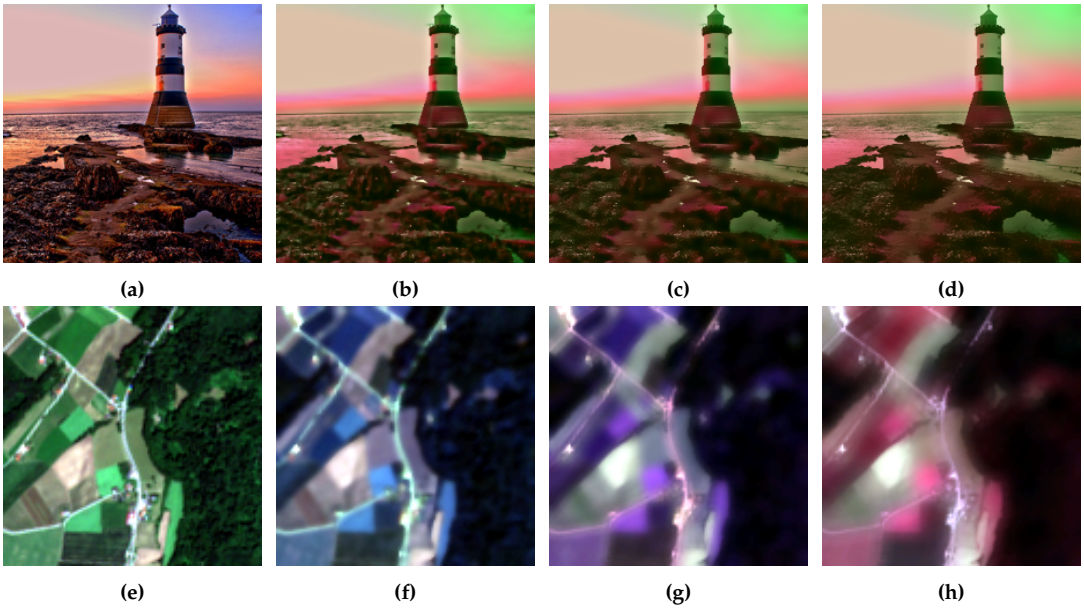
Examples of spectral features remaining with bilateral filter and patch rotation transformations applied to images of the ImageNet shown in first row, and BigEarthNet dataset shown in second row. Effect of transformation is shown for different kernel diameters and grid sizes, with **(a,e)** no transformations applied, **(b,f)** bilateral filter with kernel diameter of 5 and patch shuffle with a grid size of 4, **(c,g)** bilateral filter with kernel diameter of 9 and patch shuffle with a grid size of 8, and **(d,h)** bilateral filter with kernel diameter of 15 and patch shuffle with a grid size of 15. For the BigEarthNet dataset, RGB channels are visualized.



(a) (b) (c) (d)

(e) (f) (g) (h)

Examples of texture features remaining with patch shuffle and patch rotation transformations applied to images of the ImageNet shown in first row, and BigEarthNet dataset shown in second row. Effect of transformation is shown for different grid sizes, with **(a,e)** no transformations applied, **(b,f)** patch shuffle with grid size of 4 and channel shuffle with a share of 2/3, **(c,g)** patch shuffle with grid size of 8 and channel shuffle with a share of 1, and **(d,h)** patch shuffle with grid size of 15 and channel shuffle with a share of 1. For the BigEarthNet dataset, RGB channels are visualized.



(a) (b) (c) (d)

(e) (f) (g) (h)

Examples of shape features remaining with bilateral filter and channel shuffle transformations applied to images of the ImageNet shown in first row, and BigEarthNet dataset shown in second row. Effect of transformation is shown for different grid sizes, with **(a,e)** no transformations applied, **(b,f)** bilateral filter with kernel diameter of 5 and channel shuffle with a share of 2/3, **(c,g)** bilateral filter with kernel diameter of 9 and channel shuffle with a share of 1, and **(d,h)** bilateral filter with kernel diameter of 15 and channel shuffle with a share of 1. For the BigEarthNet dataset, RGB channels are visualized.