UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GABRIEL HENRIQUE MORO

# Quantitative descriptors for a range of visual biologic and synthetic pigmentation patterns

Thesis presented in partial fulfillment of the requirements for the degree of Master of Computer Science

Advisor: Prof. Dr. Marcelo Walter

Porto Alegre
September 2022

## CIP — CATALOGING-IN-PUBLICATION

Moro, Gabriel Henrique

Quantitative descriptors for a range of visual biologic and synthetic pigmentation patterns / Gabriel Henrique Moro. – Porto Alegre: PPGC da UFRGS, 2022.

67 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2022. Advisor: Marcelo Walter.

1. Quantitative descriptor. 2. Image processing. 3. Pattern generation. 4. Pattern evaluation. I. Walter, Marcelo. II. Título.

# ABSTRACT

*Nature is exuberant in visual patterns: spots on mammals, veins on leaves, maze-like structures in fishes, and many others. Computer Graphics has achieved success in modeling and rendering many of these patterns. Nevertheless, a fundamental problem remains: how to validate the results beyond visual comparison. Here we propose a set of quantitative descriptors tailored to describe visual biological patterns found in many species of mammals and fishes. We are interested in analyzing the most common pattern structures: spots, labyrinths, and stripes. First, we compute a set of metrics from a dataset of real patterns – curated from scratch from publicly available repositories – and synthetic images generated using reaction-diffusion simulations. We calculate our first four descriptors as ratios of "global" measures. The last eight descriptors are the standard deviation of the average measure of pattern region features, giving us twelve descriptors. We validate the descriptors through two machine learning tasks on an augmented dataset with real and synthetic patterns. First, the descriptors are used as features of a supervised classifier with an overall accuracy of 98.4%, a result better than that obtained with a general state-of-the-art convolutional classification network. Second, we tested the descriptors in an unsupervised clustering task, differentiating natural from artificial patterns and identifying species studied in the natural pattern set. Clustering was also used for unsupervised detection of regions of biologic patterns over larger images.*

**Keywords:** Quantitative descriptor. image processing. pattern generation. pattern evaluation.

# RESUMO

*A natureza é exuberante em padrões visuais: padrões de manchas em mamíferos, veias em folhas, estruturas labirínticas em peixes e muitos outros. A Computação Gráfica alcançou sucesso na modelagem e renderização de muitos desses padrões. No entanto, um problema fundamental permanece: como validar os resultados além da comparação visual. Neste trabalho, é proposto um conjunto de descritores quantitativos adaptados para descrever padrões biológicos visuais encontrados em várias espécies de mamíferos e peixes. Foram analisadas as estruturas de padrões mais comuns, como padrões de manchas, labirintos e listras. Primeiro, é calculado um conjunto de métricas para uma base de imagens de padrões reais – selecionadas de repositórios disponíveis publicamente – junto com imagens sintéticas, geradas usando simulações de reação-difusão (reaction-diffusion). Os quatro primeiros descritores são calculados a partir de razões entre medidas "globais". Os oito descritores restantes são definidos como o desvio padrão da média das métricas obtidas a partir das características das regiões do padrão, totalizando doze descritores. Estes descritores são validados em duas tarefas de aprendizado de máquina em um conjunto de dados com padrões reais e sintéticos. Primeiro, os descritores são usados como entradas para um classificador supervisionado, obtendo uma precisão geral de 98,4%, um resultado melhor do que o obtido em rede de classificação convolucional de propósito geral de última geração. Para a segunda validação, os descritores são testados em uma tarefa de agrupamento (clustering) não supervisionado, sendo capaz de diferenciar padrões naturais de artificiais e também identificar espécies estudadas no conjunto de padrões naturais. O agrupamento também foi usado para detecção não supervisionada de regiões de padrões biológicos em imagens maiores.*

**Palavras-chave:** Descritores quantitativos, processamento de imagens, geração de padrões, avaliação de padrões.

# LIST OF FIGURES

# LIST OF TABLES

# CONTENTS

# 1 INTRODUCTION

Nature (BALL, 2016) has a huge diversity of patterns. This characteristic attracted the attention of Computer Graphics research from the very beginnings of the field (SPRINGMEYER, 1988; FOURNIER, 1989). A particular challenge within the modeling of Natural Phenomena is validating results: how to assess the output quality compared to the real phenomenon being simulated. Most works present empirical, qualitative methods, comparing their results visually with the target natural ones, making these evaluations subjective and not as precise (SADEGHI et al., 2012; WANG et al., 2018; GALIN et al., 2019). The Mathematical Biology field (MURRAY, 2001) also increasingly needs better quantitative tools to assess natural pattern results (KONDO; SHIROTA, 2009; KIM et al., 2020). Therefore, the need for a quantitative approach is perceived.

Among the many natural phenomena already addressed in Computer Graphics before, pigmentation patterns have received much attention, spanning 30 years of research, from the seminal work of Turk (1991) and Witkin and Kass (1991) to state-of-the-art results of Malheiros, Fensterseifer and Walter (2020). In those works, many patterns are spot-based. Although these patterns are defined on a 3D object, they are expressed on the skin or surface of animals. Therefore, despite perspective distortions, they can be analyzed in 2D.

In this work, based on the careful study of a range of spot, labyrinthine, and stripe patterns, we first define a set of descriptors suited to capture their main characteristics. For this work, the term descriptor is used in the same sense as in content-based image retrieval (CBIR) systems, that is, as a general scalar measurement that captures a particular type of image structure. Such descriptors would facilitate the evaluation of results, resulting in a better understanding of the pattern structure, bringing consistency to the Computer Graphics field. We then explore and validate this set through three machine learning tasks: supervised pattern classification, unsupervised image clustering, and parameter estimation for a mathematical model using a regression method.

For the studied topologies (spotted, labyrinthine, and striped patterns), we defined a limited set of natural and synthetic patterns, using only regular patterns with two distinct colors, so that the pattern structure may be easily obtained and analyzed. Thus, for this work we create a set of descriptors focusing only on the pattern structure, not taking into account features such as color or frequency domain measures.

Although there are powerful classification methods using neural networks today, such as *NASNet* (ZOPH et al., 2018) and *EfficientNetV2* (TAN; LE, 2021), these are general solutions that do not provide explicit measures to compare and understand pigmentation patterns, being black box models. With straightforward descriptors as proposed in our work, future procedural biological-based engines for pattern formation can be improved.

## 1.1 Justification

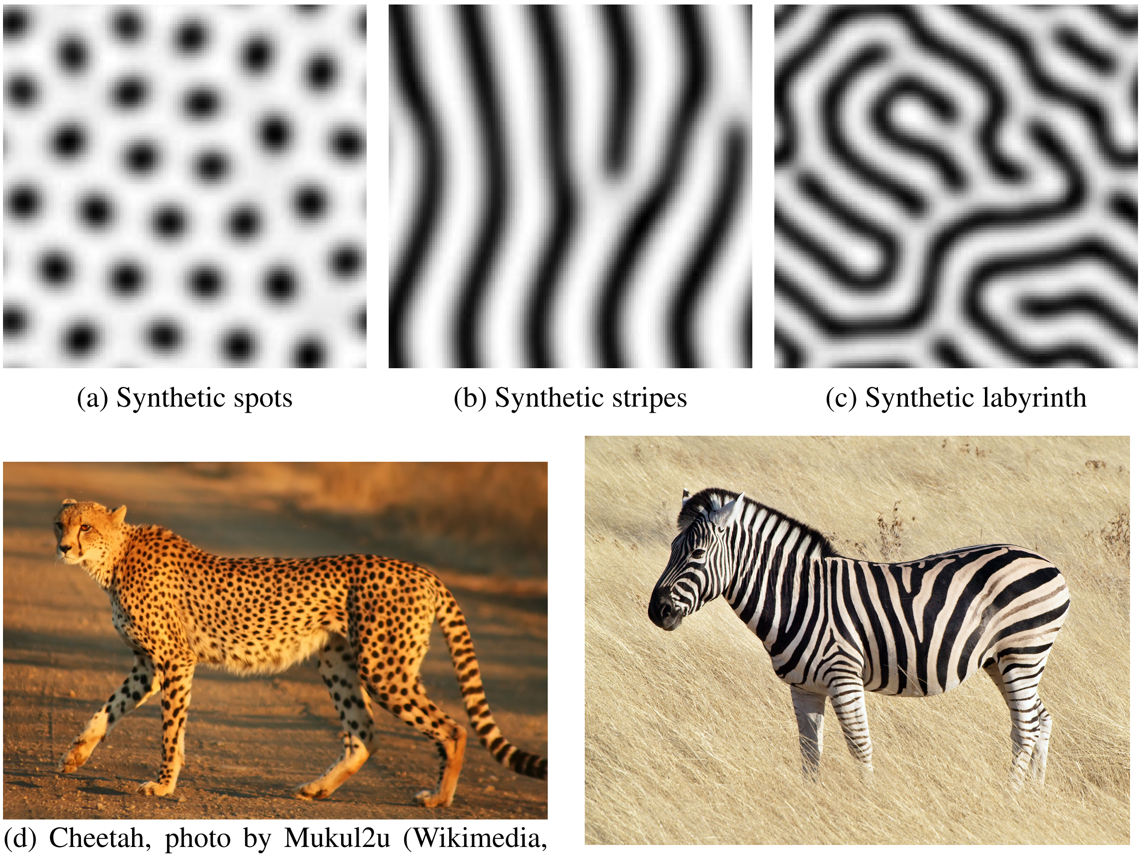
As stated before, most works synthesizing natural patterns evaluate their results using qualitative methods due to a lack of quantitative methods for synthetic pattern evaluation. When present, they are usually either too general or too context-specific and, therefore, are not capable of describing common features in nature, such as spotted and striped patterns. There is also a lack of work focusing on analyzing the structure of natural patterns, and with this, useful quantitative descriptors for pattern evaluation.

We focused on the design and evaluation of descriptors for natural and synthetic patterns. The availability of such descriptors helps the evaluation of other research by giving quantitative measures. The target patterns were the most frequent animal patterns: regular spots, stripes, and labyrinthine patterns. These are straightforwardly generated with mathematical models, such as reaction-diffusion, and encompass a large portion of existing natural patterns. Examples of synthetic patterns we generated with a reaction-diffusion model are shown in Figure 1.1, illustrating common pattern topologies.

## 1.2 Objectives

Our main goal is to design a set of descriptors to evaluate natural and synthetic visual patterns, focusing on the description of the patterns structure. The target patterns are regular spots, stripes, and labyrinthine patterns. These simple patterns and their combinations account for diverse visual patterns in the animal kingdom. We want to propose quantitative descriptors that enable the separate measurement of specific characteristics of a given pattern, such as the size of elements, spatial distribution, among others. With these descriptors, we expect to be able to evaluate both the topology and regularity of patterns. Besides, our descriptors may be used on various applications, such as identification of animal species.

Figure 1.1: Example of synthetic (a) spotted, (b) striped, and (c) labyrinthine patterns generated through reaction-diffusion simulations, and two animals whose patterns were used in this work.



(a) Synthetic spots

(b) Synthetic stripes

(c) Synthetic labyrinth

(d) Cheetah, photo by Mukul2u (Wikimedia, CC BY 3.0)

(e) Zebra, photo by Gusjer (Flickr, CC BY 2.0)

### 1.3 Organization

This dissertation is organized as follows. The next chapter first explains reaction-diffusion systems, followed by a review of related works, divided into two categories: Mathematical Biology and Computer Science. Chapter 3 presents the methodology, describing how the descriptors were designed, the pipeline used for their computation, and the individual descriptor definitions. Then, Chapter 4 presents the results obtained through the validation methods, explaining the datasets used for these tests and the different test setups executed. Concluding, we present final remarks and comments on possible future works.

# 2 BACKGROUND AND RELATED WORK

In this chapter, we first present background material on Reaction-Diffusion (RD), a paradigm first proposed by Alan Turing Turing (1952) and then followed by many others (MEINHARDT, 1982; MURRAY, 2001). RD is used in many fields, including modeling of biologic pigmentation patterns (BARD; LAUDER, 1974), seashells (FOWLER; MEINHARDT; PRUSINKIEWICZ, 1992), animal coat patterns (TURK, 1991) (WITKIN; KASS, 1991) (MALHEIROS; FENSTERSEIFER; WALTER, 2020), fruit decay (JR; RAJA; BADLER, 2011) and flower pigmentation patterns (RINGHAM et al., 2021). Modeling, simulation, and rendering of visual patterns are addressed not only by Computer Graphics but also by other fields such as Mathematical Biology. Considering this, we review related work from both fields, focusing on individual animal identification, pattern generation automation based on input images, and limited attempts at quantitative pattern analysis and feature extraction, among others.

## 2.1 Reaction-diffusion

A reaction-diffusion system is defined by a system of partial differential equations (PDEs) used for chemical and biological simulations. Such a system describes the interactions between two reagents in space and time, where one reagent works as an activator, accelerating the reagent production, and the other as an inhibitor that slows it down.

Turing (1952) first proposed a reaction-diffusion model for pattern formation in 1952, which was later explored by Bard and Lauder (1974). This model is described by Equations 2.1 and 2.2 that describe the behavior of two reagents $a$ and $b$, where the term with $S_p$ describes the reaction process while the term with Laplacian operator $\nabla^2$ describes the diffusion process.

$$\frac{\partial a}{\partial t} = S_p(16 - ab) + D_a \nabla^2 a \tag{2.1}$$

$$\frac{\partial b}{\partial t} = S_p(ab - b - \beta) + D_b \nabla^2 b \tag{2.2}$$

A simplification of this model was proposed by Malheiros, Fensterseifer and Walter (2020), featuring a reparameterization of the original model, as seen in Equations 2.3

and 2.4.

$$\frac{\partial a}{\partial t} = 16 - ab + rs\nabla^2 a \tag{2.3}$$

$$\frac{\partial b}{\partial t} = ab - b - 12 + s\nabla^2 b \tag{2.4}$$

From the original equations, the variables $S_p$ and $\beta$ have their values set as 1 and 12, respectively, and $D_a$ and $D_b$ are replaced by a ratio $r$ and scale $s$, where $r = \frac{D_a}{D_b}$ and $s = D_b$.

As for applications, Prum and Williamson (2002) use a reaction-diffusion model to simulate the pigmentation patterns found on feathers, also simulating a pigmentation pattern transition phenomena observed in nature. The work of Jr, Raja and Badler (2011) uses the biological nature of reaction-diffusion simulations to emulate fruit decay, simulating the growth of fungal and bacterial infections and how bacterial growth and nutrient depletion interact with the fungal spread. Ringham et al. (2021) simulate a range of flower patterns directly on geometric models of the flower species studied, using different mathematical models to simulate these patterns, including some variations of reaction-diffusion models.

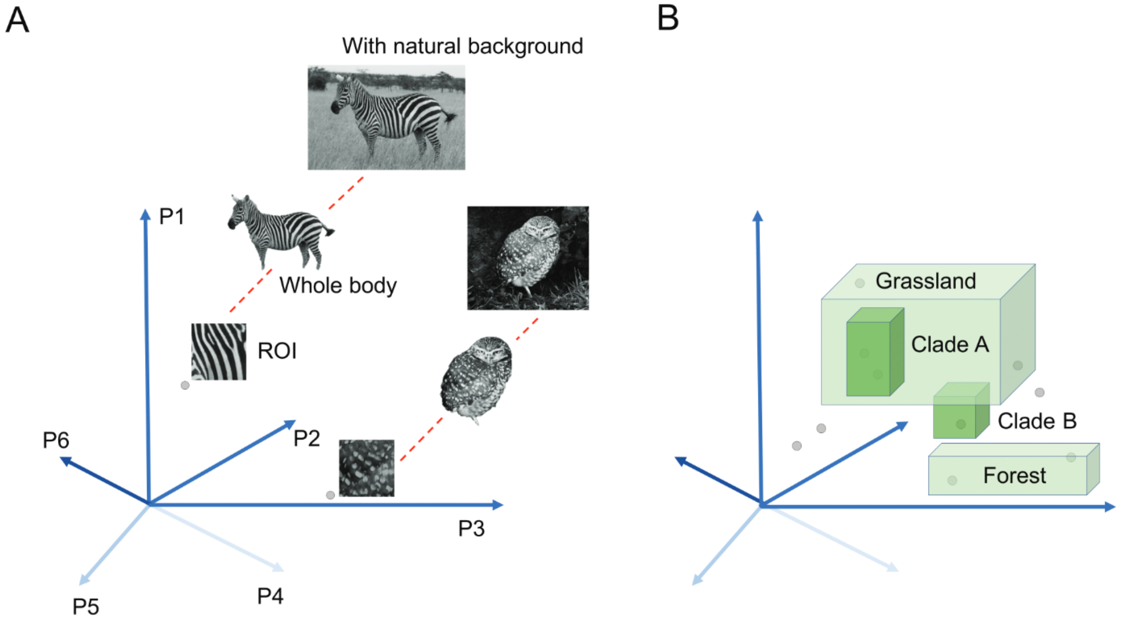
## 2.2 Related Work

As stated before, there is a lack of works that use explicit quantitative methods. We review in this section a group of papers that vary in focus but are related to the main subject of our research: individual animal identification, pattern generation automation, and limited attempts at quantitative pattern analysis and feature extraction.

First, we review Stoddard and Osorio (2019). This work uses the analogy that color spaces can represent any color and introduces the idea of a low dimensional *pattern space*, where any animal pattern could be described in this space by a few perceptual dimensions. The support for such space comes mainly from the conclusion that animal patterns are not random but have a definite statistical structure (SIMONCELLI; OLSHAUSEN, 2001). Figure 2.1 exemplifies an possible pattern space, where the parameters $P_n$, represented as axes, define the space.

Their work explores possibilities for creating pattern space, listing five steps: “Capturing an Image”, “Modeling Visual Acuity”, “Calculating Power Spectra and Image

Figure 2.1: Representation of an hypothetical pattern space where each axis $P_n$ represents a pattern parameter.



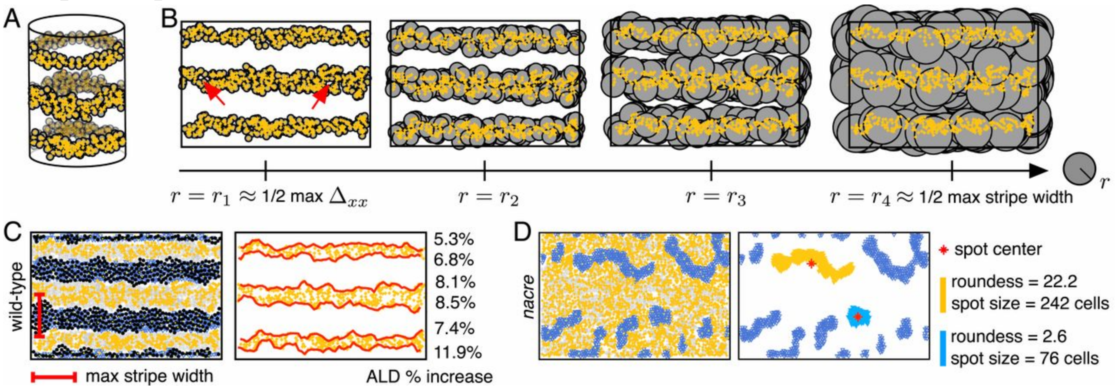
Source: Stoddard and Osorio (2019) (Figure 1)

Statistic", "Detecting Local Features and Objects", and "Detecting Higher-Level Object". However, they do not define one.

Relating to the usage of quantitative descriptors or features, we reviewed two works. McGuirl, Volkening and Sandstede (2020) recognize the difficulties in a purely subjective assessment of results. Using a combination of topological data analysis (TDA) and machine learning (ML) tools, they propose a method for quantifying self-organized patterns applied to zebrafish skin patterns. They employ their technique on *in silico* zebrafish patterns, synthesized with a cell-based model previously defined by the same authors (VOLKENING; SANDSTEDE, 2018). This model analyzes pigment cells as point masses, tracking their spatial coordinates on a growing horizontally continuous 2D domain (continuous on the $x$ axis), as shown in Figure 2.2A, while they move and interact, also modeling cell death, birth, and transition between five different behavior types.

As stated before, the authors' approach to quantifying patterns uses a combination of TDA and ML. Both processes rely on their discrete cell-based model. The TDA technique used is known as *persistent homology* and is illustrated in Figure 2.2B. For this, a ball of radius $r$ is placed into each cell's center and the analysis is made with the shapes generated by the union between these balls and how it changes with each increment of $r$. As for ML, they use a single-linkage clustering method to divide their agents (pigment cells) into different classes trying to determine topology clusters for spots or stripes. For this, the authors start with one cluster for each cell and then consecutively merge two clusters closer to each other until there are either one or a predetermined number of clusters.

Figure 2.2: Visualizations of (A) the horizontal continuity, (B) *persistent homology* applied on the authors' simulation, (C) the detection and measuring of stripes, and (D) an example of spot measures.



Source: McGuirl, Volkening and Sandstede (2020) (Figure 3)

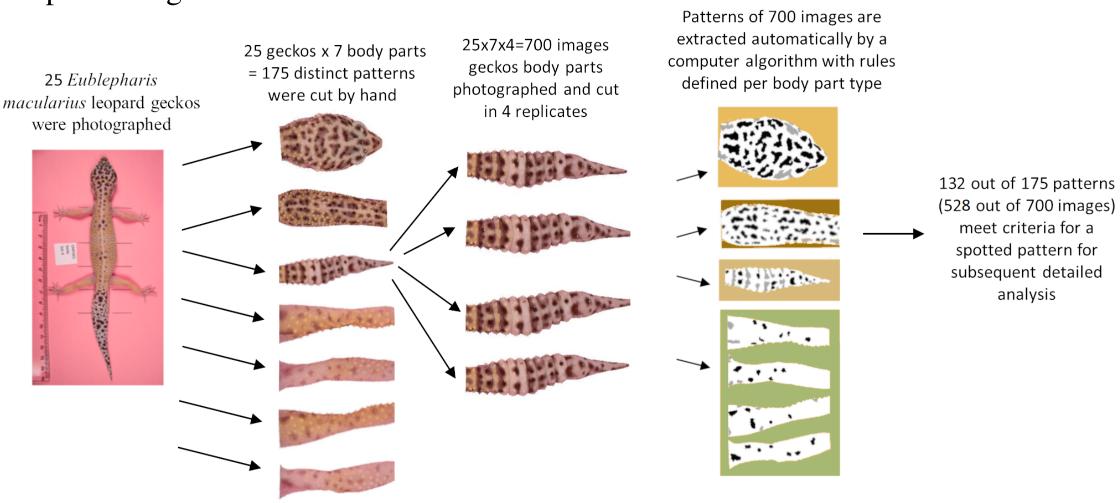
This quantification is calculated by six pattern features: the number of spots and stripes on the pattern, the width and straightness of the stripes, and the size, roundness, alignment, and center width of the spots. They compute these features using TDA tools, using points and clusters of points, and on their simulation scenario. Considering this, some of these features may not be useful nor possible in different or general settings.

In addition, they also analyze *pattern formation events*, that is, an estimation of the time at which specific features emerged, this, however, relying on their simulation and not being useful on general scenarios.

Another work that uses quantitative features is the recent work of Glimm et al. (2021), that shares many goals with our own, mainly the definition of quantitative methods to analyze and describe pattern variation. They used 25 leopard geckos (*Eublepharis macularius*) as a case study, analyzing seven distinct "regions of pattern" on the bodies with four photos for each region, for a total of 175 regions of pattern and 700 images. The process of acquisition of these images is shown in Figure 2.3, as well as the image processing.

As the authors use flat 2D images that are photos taken from above the animal samples, they divide the spots into *edge spots*, those partially cut off by the edges duo to this flat representation, and *interior spots*, defined as all other spots. The authors use 14 indices for quantitative pattern description. From these, 6 are computed using all spots. The first is named *fraction of melanistic area*, defined as the ratio between the area of the black pixel by the total area (similar to R1 from Sun et al. (2017)). They also have two

Figure 2.3: Flowchart from Glimm et al. (2021), showing the process of image acquisition and processing.


25 *Eublepharis macularius* leopard geckos were photographed
25 geckos x 7 body parts = 175 distinct patterns were cut by hand
25x7x4=700 images geckos body parts photographed and cut in 4 replicates
Patterns of 700 images are extracted automatically by a computer algorithm with rules defined per body part type
132 out of 175 patterns (528 out of 700 images) meet criteria for a spotted pattern for subsequent detailed analysis

Source: Glimm et al. (2021) (Figure 4)

distance-based indices, the fist being called *peak length*, described as a “typical” distance between spots or as a characteristic wavelength described by Miura, Komori and Shiota (2000). The other distance index is the *mean minimum distance* (MD), computed as the mean of the average distance between a spot centroid and the three closest spot centroids for all spots. The last index computed using all spots is called *spot intensity* (SI), defined as the average green value of the spots (considering the RGB values, each pixel ranging from 0 to 255). They also measure the standard deviation of MD and SI as new indices, where the MD standard deviation is measured using all the distances used to compute the index, instead of using the averages.

The remaining eight indices are computed using only the interior spots. Two of these indices are measure using the axes of an ellipse that has the same second moment of the spots. One is named *mean spot diameter* (SS), being the average of all lengths of the major axes of the ellipses. The other is called *mean ellipticity* (EE), calculated as the mean ratio between major and minor axes of said ellipses. They also have a *spot area* (SA) index, defined as the mean area of the spots (in $cm^2$), and a *mean spot elongation*, defined by the equation

$$EL = \frac{area}{2d^2} \tag{2.5}$$

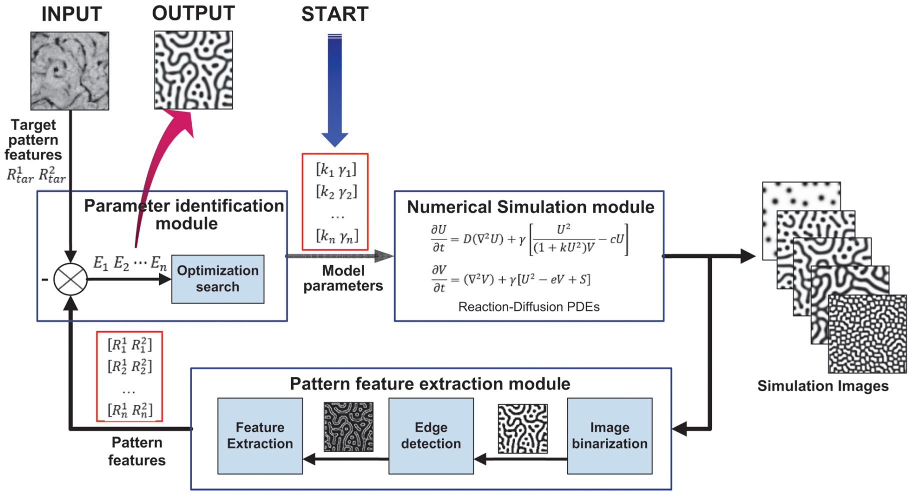
where $d$ is a number got by eroding the spots until they disappear, counting the number of erosion steps needed to do so. The last four indices are computed as the standard deviation of SS, EE, SA and EL.

Using these 14 indices they define a composite distance between two patterns and the pattern variation among distinct body parts using a standard Mahalanobis distance. They also conclude how the patterns were established through the species body based on the observation related to the presence of pigmentation patterns on these distinct body parts.

Some attempts to describe patterns quantitatively address the problem of *inverse rendering*, that is, given an input image, try to synthesize a visually similar pattern. Sun et al. (2017) addresses this by defining a framework that, with an input image, infers the image pattern topology and calculates the parameters for the reaction-diffusion model used in their simulations. Their simulation results are divided into two categories: vascular mesenchymal cells (VMCs), consisting of spotted, striped, and labyrinthine patterns, and lung development, with "Normal branching pattern" and "Branching pattern without side branching".

The framework has three modules, as shown in Figure 2.4. The first module evaluates the reaction-diffusion model equations result and saves it as an image; the second module extracts the pattern topology and quantitative features using two quantitative measures; the third module compares those measures with the ones of the target image (given as input) to evaluate the model parameters. A feedback loop then uses this evaluation to calculate new parameters.

Figure 2.4: Illustration of the framework, showing their three modules and an example of input and output.



Source: Sun et al. (2017) (Figure 1)

They use two ratio parameters to evaluate the model quantitatively. The first is the ratio between the pattern area and the total area of the image, called $R^1$, used to determine the pattern topology. The second, called $R^2$, is the perimeter area ratio (PA ratio) for the pattern, which describes the pattern scale. There is also a distance function between target pattern ratios and the obtained pattern ratios, defined as

$$f_{VMCs} = [(R^1 - R^1_{tar})^2 + (R^2 - R^2_{tar})^2] * 10^5 \tag{2.6}$$

with $R^1_{tar}$ and $R^2_{tar}$ as the target ratios and the multiplication by $10^5$ to approximate the value to an integer.

Their experimentation consists of three VMC target patterns, from which they obtain the starting $R^1$ and $R^2$ target values for the simulation and then the model parameters. The result values obtained for spotted, striped, and labyrinthine target patterns are, respectively, 0.22, 0.4, and 0.46 for R1 and 0.31, 0.56, and 0.67 for R2. The simulation results are, respectively, 0.2017, 0.4145 and 0.4524 for R1 and 0.3044, 0.5469 and 0.6397 for R2.

With a motivation similar to the work from Sun and colleagues presented earlier, the work by Mordvintsev et al. (2021) also addressed the problem of synthesizing example-based reaction-diffusion textures, although not focusing on natural patterns. They inspired their model on the Neural Cellular Automata (NCA) described by Niklasson et al. (2021). Based on the PDE

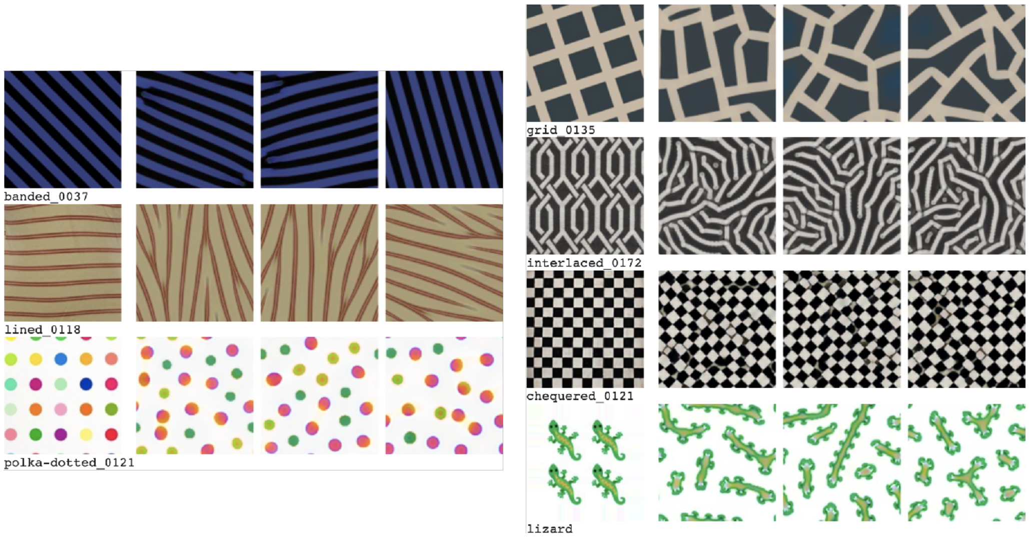
$$\frac{\partial b}{\partial t} = c_i \nabla^2 x_i + f_\theta(x_0, ..., x_{n-1}) \tag{2.7}$$

where $(x_0, ..., x_{n-1})$ represent the $n$ reagents, $c_i$ are the diffusion coefficients, and $f_\theta$ is the reaction function, they propose a new model defined by

$$x_i^{t+1} = x_i^t + c_i K_{lap} * x_i^t d + f_\theta(x_0^t, ..., x_{n-1}^t) r \tag{2.8}$$

where $c_i$ and $\theta$ are Cellular Automata (CA) parameters that control the model behavior, $K_{lap}$ is a 3x3 Laplacian convolution kernel, $d = \Delta_t / \Delta_h^2$ and $r = \Delta_t$ are parameters that control the diffusion and reaction, and the function $f_\theta$ is a two layer neural network. The authors use 32 reagents and 8320 neural network parameters. Their matrices start with a number of scattered Gaussian blobs, each iteration updated using Euler method, adding *seed state* injections periodically, so that the model does not forget how to develop patterns from the seeds. The parameters are learned through a differentiable optimization method, and as the work of Sun et al. (2017), the learned rules are not explicitly available and therefore not general. Also, they validate their results using side-by-side comparison, that is, using qualitative methods. Figure 2.5 shows some of these comparisons.

Figure 2.5: Results presented by Mordvintsev et al. (2021), with the first image of each line as a target pattern and the three next are different RD learned model results with 5000 iterations.



Source: Mordvintsev et al. (2021) (Figure 2)

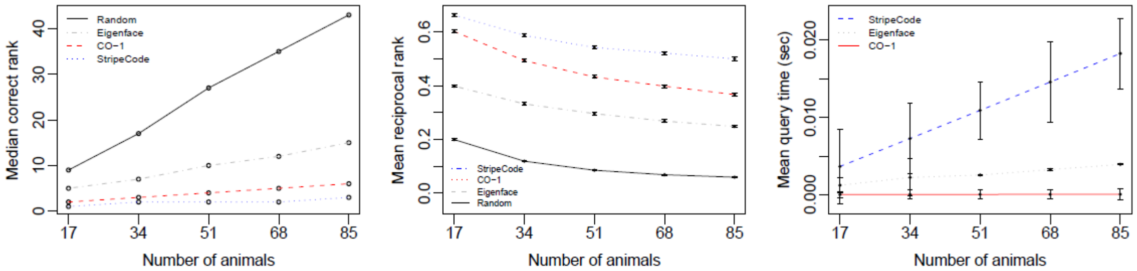
We also reviewed two more works related to animal identification. Lahiri et al. (2011) describe an algorithm for identification of individual animals based on their coat patterns. Although their image dataset consists only of zebra images, the authors state that the technique is applicable to different animals if they present some morphological characteristics such as stripes. Their dataset was collected with the help of field ecologists and has images of two species of zebras, each image associated with an animal identifier. These images have their regions of interest (ROI) manually cropped with no restrictions on body location, but assuming to be cropped “as consistently as possible”.

These cropped images are filtered into $k$ horizontal bands, each band having a summary row of pixels that keeps the mean value of a column. With this, for each row, they create a sequence of $n$ pairs of color and length values, calling this sequence a *StripeString*. An ordered set of these *StripeStrings* creates a *StripeCode* for an image in the dataset. With this, they can define the similarity of two images by calculating the distance between two *StripeCodes*, defined as “the average of the distance between corresponding *StripeStrings*”.

They compared their results with two other animal recognition algorithms, Eigenface (TURK; PENTLAND, 1991) and an algorithm based on multi-resolution histograms of differential image features (CO-1) (RAVELA; GAMBLE, 2004). These tests consisted of, for $N$ unique animals, randomly selecting a number of images per animal (*ipa*, fixed as 1 for the tests) to create a database, choosing randomly an image of one of the $N$ animals, and then ranking the animals in the database by the minimum distance to the selected image.

Those tests used three evaluation metrics: the median rank of the correct animal, the mean reciprocal rank (MRR), and the average query time. The results are presented as three graphs shown in Figure 2.6, where the better results show lower values for the median rank and higher values for MRR.

Figure 2.6: Comparison of results of different algorithms with the one proposed by Lahiri et al. (2011).



(a) Median rank, $ipa = 1$.

(b) MRR, $ipa = 1$.

(c) Mean query time, $ipa = 1$.

Source: Lahiri et al. (2011) (Figure 5)

The last reviewed work is *HotSpotter*, presented by Crall (2013). *HotSpotter* is an algorithm for individual animal identification, matching animal images with images in a labeled dataset. The dataset has images of five different species of animals: giraffes, jaguars, lionfish, plains zebras, and Grevy’s zebras. The paper focuses on the experiments with zebras because there are more images of zebras in the databases.

They present two algorithms: the first, described as *one-vs-one*, and the second described as *one-vs-many*, both with five steps: preprocessing, matching, image scoring, spatial reranking, and label scoring. For the *one-vs-one* algorithm, every image in the database and query image is cropped to a rectangular ROI and resized to a standard dimension with the same aspect ratio. After this, the image features and descriptors are extracted and a small forest of k-d trees is built as a fast search data structure. Then each image in the database receives an initial score. For the fifty top-scored images it is computed a set of spatially consistent “inliers”. With these inliers is created a final correspondence set, in which is applied a scoring function that defines the final scores. Using these scores, the authors present two methods to score each label: the first selects the label of the highest-scoring image, while the second combines the scores from all images with the same label.

The *one-vs-many* algorithm preprocessing step creates a small forest of k-d trees using one nearest neighbor data structure to index all database descriptors. Then, for each query image, the nearest neighbors are found by searching this forest, with scores being calculated based on the distance to the query images. For this algorithm, they present four scoring methods: difference in distances between nearest neighbors, a ratio score to $k$ nearest neighbors, a log-ratio score, and match counting. For the giraffe, jaguar, and lionfish datasets, they achieved perfect results: 95% of correct top-ranked labels and 98% top-five labels for the Grevy's zebra dataset, and over 99% for the plain zebra dataset.

Furthermore, it is worth mentioning works on state-of-art deep learning classification models. Wäldchen and Mäder (2018) reviews current deep learning approaches to automated classification tasks. In Miao et al. (2019), the authors use convolutional neural networks to classify twenty African wildlife species and, from a dataset of 111,467 images, obtained an accuracy of 87.5%, and the already mentioned Tan and Le (2021), that provides a general classification model. These are generalist approaches, focusing on a different objective than our work.

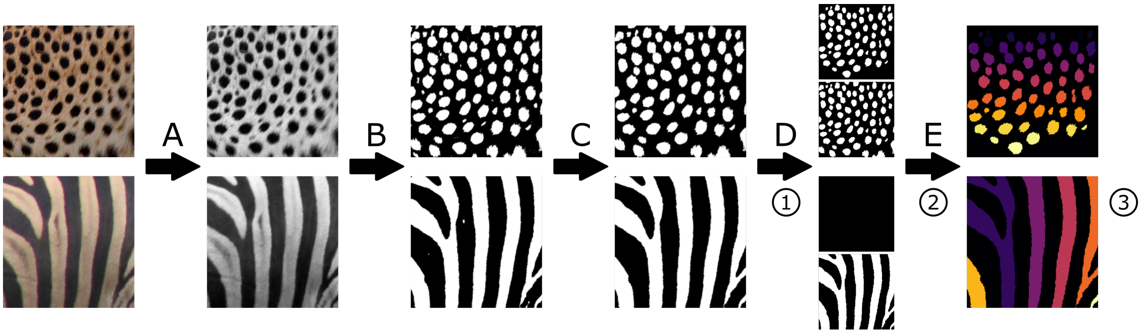
## 2.3 Discussion

As we show in the following sections, our work brings a more general study in terms of target patterns and delivers explicit descriptors, useful for quantitative synthetic pattern assessment. The reviewed works, when presenting quantitative descriptors, employed them on specific scenarios with a limited scope or focusing them as solution for feature matching. Instead, we focus on a general approach for pattern description values, outside of a black-box deep-learning context or a limited species description problem.

# 3 METHODOLOGY

In this chapter, we describe the steps needed for the descriptor extraction process, which is schematically illustrated in Figure 3.1 and further explained in Section 3.2. Focusing on biological pigmentation patterns found in the skin of mammal and fish species, and synthetic reaction-diffusion patterns, we limited our scope to spots, stripes, and labyrinthine patterns.

Figure 3.1: Sequence of operations for descriptors computation: (A) conversion to grayscale, (B) thresholding, (C) noise removal, (D) partial region removal, and (E) measurement of individual regions. The step between (D) and (E) shows two patterns, the upper image shows the pattern after the partial region removal while the bottom image shows the pattern before. In this example, the zebra pattern is completely removed by this operation, and because of this it is not done for this pattern. (1), (2), and (3) represent where specific descriptors are computed within the sequence of operations, where: (1) pattern area percentage and border perimeter ratio, (2) pattern loss percentage and border pattern ratio, and (3) local descriptors.



Source: The author.

We assume that all these patterns are presented in uniformly colored regions that stand out from a distinct and single-colored background. Moreover, we assume that all input images have already been cropped from a larger image, where elements such as a background or other animal details are eliminated. We also define that the input images have a square aspect ratio (1:1) but with an arbitrary resolution. Being square, we need not be concerned with distortion between horizontal and vertical axes. However, as a more general approach, such difference could be propagated to later steps, generating a correction factor to measure areas and distances. We are interested in analyzing patterns with no dependency on the overall scale or image resolution, so all local descriptors are normalized, as described later.

The next section explains how the descriptors were designed and chosen. Then, we describe the processes we use on the images for computing the descriptors, as well as a brief explanation of thresholding and the experiments done to find the best parameters.

The last two sections explain our descriptors, divided into two categories, global and local descriptors, and how they are calculated.

## 3.1 Design of descriptors

We explored a series of measures that combined the simplicity of implementation, generality, and robustness, tested against a dataset of biologic and synthetic patterns, described in Section 4.2. We used a relevance validation process to determine which ones were useful and which ones proved to be redundant, keeping only the most significant descriptors for characterizing the pattern images. This process included the validation results obtained through our classification process, described in Section 4.3, and an analysis of value consistency based on graphs plotted with the descriptor values for each dataset, and groups of datasets with some common characteristics, such as whether the images are synthetic or real, or whether they have the same topology. Mainly, we tested setups of descriptors against a “current default setup”, keeping the setups that presented better results for our validation methods, explained in Chapter 4, while discarding descriptors that showed little relevance and worse results, and defining the default descriptor setup as the setup that showed the best results overall.

Descriptors are grouped into two categories: global and local. Global descriptors are calculated based on the whole image and are ratios of pixel counts. Global measures cannot explain specific details of the images but showed to be very useful in indicating the overall pattern structure. At the beginning of our process, we use global descriptors, hinting at the pattern topology and overall structure. Local descriptors consider the shapes and distribution of pattern details and are computed after the segmentation and labeling of individual regions of the pattern.

Here, we search for solutions outside of the black box models presented by DL models. We set this objective because we expect to obtain straightforward and general scalar values capable of explicitly measuring and describing the studied patterns in a “white box” solution.

## 3.2 Image processing

Before computing our descriptors we process the input images as exemplified in Figure 3.1. The first three operations are done for all descriptors. These operations are: (A) converting the image to grayscale, (B) applying a thresholding algorithm to segment the foreground (the pattern itself) and the background, and (C) filtering small regions as to remove noise. The next operation is (D) the removal of partial regions, regions that are partially cut by the images border. This operation is done exclusively to spotted patterns, as stripped and labyrinthine patterns are left with almost no pattern or the pattern is completely removed, as illustrated with the zebra pattern in Figure 3.1. The last step (E), is the measurement of the features of individual pattern regions. In this process, each region of the remaining pattern is segmented and has some features extracted, which are then used to compute our local descriptors.

The threshold operation (step B) uses the adaptive Sauvola (SAUVOLA; PIETIKÄINEN, 2000) algorithm to cope with lighting variation. It defines a localized cutoff point with a clear separation of boundaries between the pattern and the background. The algorithms parameters, our experiments on them, as well as a short background on thresholding are explained in Section 3.3.

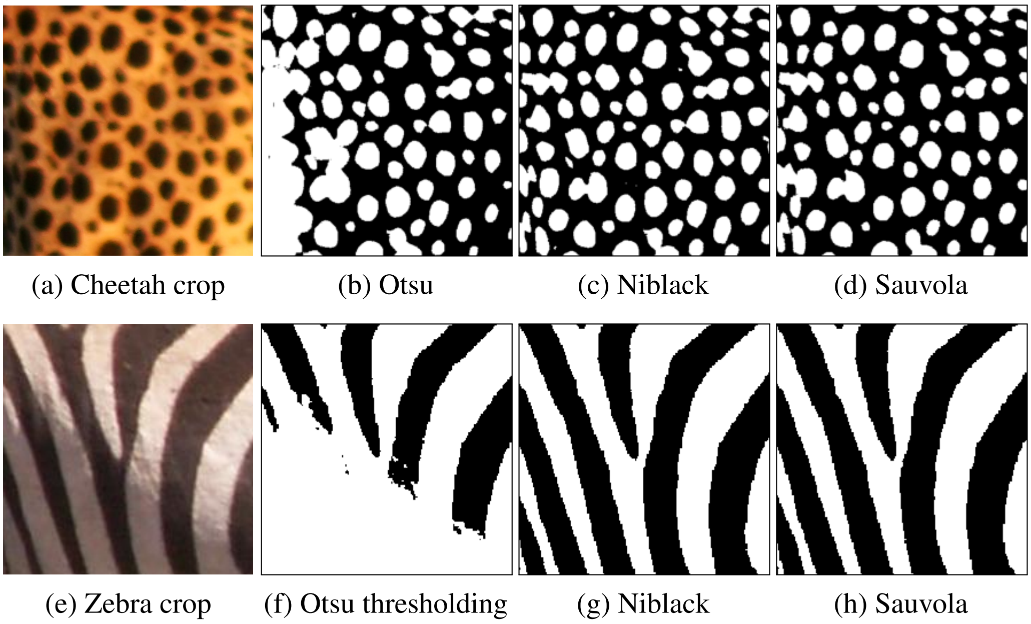
After thresholding, we perform an image cleaning process to reduce artifacts arising from inadequate image capture, low resolution, or thresholding noise (step C). All these artifacts directly affect the generation of descriptors, so we do a cleanup operation that applies to all descriptors. This operation is the elimination of small pixel groups, defined as having a minimum area of $0.1\%$ of the image area (e.g., an image with a resolution of 150x150 pixels has an area of 22,500 pixels, thus for this image we remove groups with an area of less than 23 connected pixels). Any group smaller than this minimum area is set to the background color.

## 3.3 Thresholding

A thresholding function is a process by which an image is converted from a grayscale range of $[0, 255]$ into a binary 0 or 1 value. For this work, we define that, for the binarized image, the pattern (the foreground) is set to 1 while the background is set to 0. These functions may be divided into six categories, according to Sezgin and Sankur (2004), but we explore only two: global and local (or adaptive) methods.

A global threshold method selects only one threshold value for the image. An example of a global threshold method is Otsu (1979). This algorithm analyzes the histogram of a given image and tries to find a threshold value $t$ that minimizes, or maximizes, its within-class variance. All pixels with a value lower than $t$ are considered to be background and their value is set to 0, and all pixels with a value higher or equal to $t$ are set to 1. Although these methods are faster and simple, they may present problems when the background is not constant or if there is a contrast variation across the image (or over image objects), for example with the variation of the illumination (ROGOWSKA, 2000). Figure 3.2 shows two examples of this illumination problem.

Figure 3.2: Thresholding methods applied to a (a) cheetah and a (e) zebra patterns. Otsu's thresholding problem on (b) and (f) where the image presents a variation of illumination. Local thresholding perform better, (c) and (g) using Niblack's method, and (d) and (h) using Sauvola's algorithm.



(a) Cheetah crop (b) Otsu (c) Niblack (d) Sauvola

(e) Zebra crop (f) Otsu thresholding (g) Niblack (h) Sauvola

Souce: The author

Local threshold methods are computationally more expensive but do not present these problems. For this, multiple threshold values are selected according to local features for each image region. These methods may work in two ways: calculating a local threshold for "sub-images" of an image or through analysis of near pixels intensities for each pixel in the image. Niblack (1986) and Sauvola (2000) are examples of local methods that calculate a threshold $t$ for each pixel on the image with a window centered on the analyzed pixel. Niblack local threshold is calculated by

$$t = m(x, y) - k * s(x, y) \tag{3.1}$$

and Sauvola is computed by

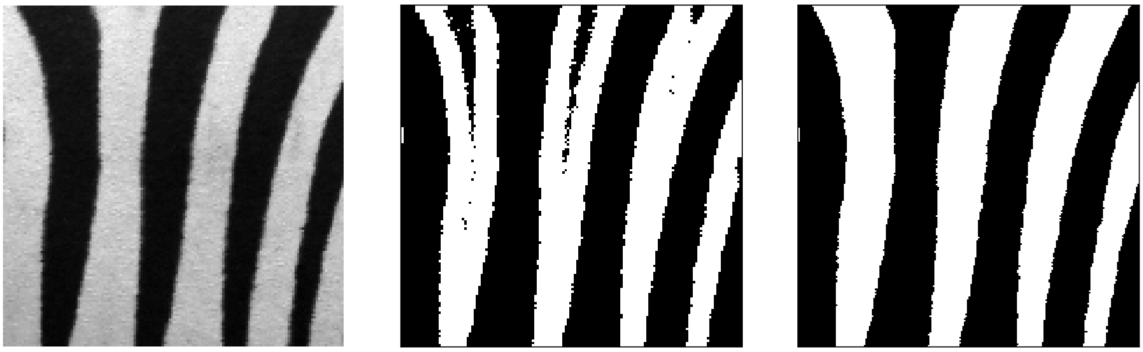
$$t = m(x, y) * (1 + k * ((s(x, y)/R) - 1)) \tag{3.2}$$

For both equations, $m(x, y)$ and $s(x, y)$ are, respectively, the mean and standard deviation of the pixels on the window centered in pixel $(x, y)$, and $k$ is a value defining the weight of the standard deviation on the threshold equation. Exclusive to Sauvola, $R$ defines the maximum value for the standard deviation $s(x, y)$.

Figure 3.2 exemplifies the cited local threshold techniques. For these examples, both methods use $k = 0.2$ and a window size of 30% of the image's smallest dimension rounded up. As stated before, the global method (3.2b and 3.2f) present a problem related to the illumination variation on the original image, this problem not being present on the local methods.

For our pipeline, as previously mentioned, we chose to use the Sauvola method. The algorithm has two parameters. The first is the size of the window, usually defined as 15% of the image resolution. In our tests, the 15% value resulted in hollow regions for some images when the analyzed regions are larger than the window, as shown in Figure 3.3b. Values smaller than 35% still present "holes" in some stripe regions, while windows of 35% show the desired binarization. Thus, we defined the window size as a 35% percentage of the input image resolution, as exemplified in Figure 3.3c.

Figure 3.3: Pipeline using a window size of (b) 15%, resulting in hollow regions, and (c) 35% for Sauvola thresholding.



(a) Original crop

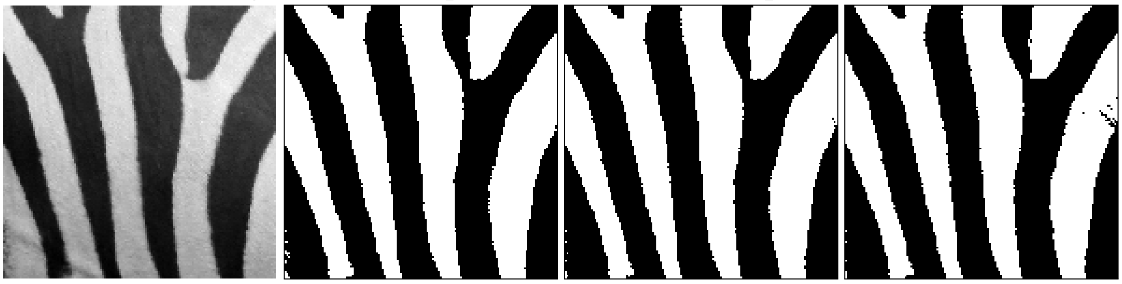
(b) window size = 15%

(c) window size = 35%

Source: The author

The second parameter is the adjustment $k$ factor, which weights the standard deviation on the thresholds formula (Equation 3.2). For this parameter, we defined its value as 0.2, as in Figure 3.4b. This value is commonly found on literature (BATAINEH et al., 2011) and was chosen after experimentation as it showed consistent results; higher values created noisy regions, as shown in Figure 3.4d.

Figure 3.4: Example of variation for the $k$ parameter.



(a) Original crop (b) k = 0.2 (c) k = 0.3 (d) k = 0.4

Source: The author

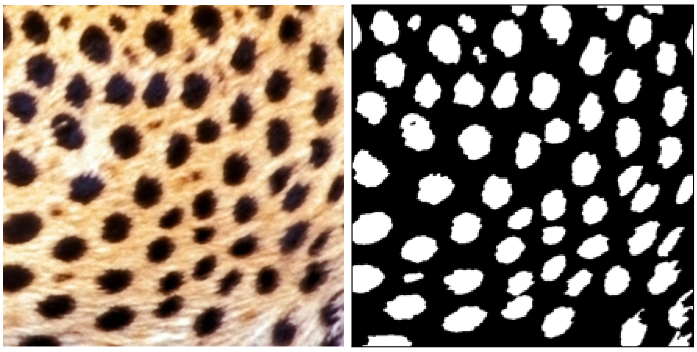
## 3.4 Global descriptors

The first descriptor is called *pattern area percentage*. It is calculated by the ratio of pattern pixels (after the preprocessing) divided by the total number of pixels. We may write this descriptor as

$$\frac{area_{pattern}}{area_{img}} \tag{3.3}$$

computed as the division of pattern area (the number of pixels with value 1) by the image area (the total number of pixels of the image), as exemplified in Figure 3.5. This measure is dimensionless, indicating the percentage of area covered by the pattern in the foreground.

Figure 3.5: A cheetah crop (a) and its binarization (b), where the white pixels are the analyzed pattern. From (b) we measure the pattern area (90,022 px) and the total area ($530^2$ = 280,900 px), from which we obtain an *pattern area percentage* of 32.05%.

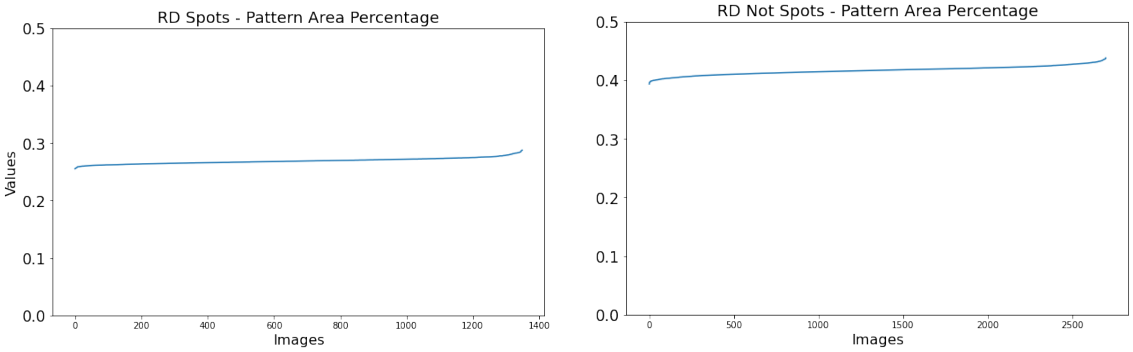


(a) Original crop (b) Pattern borders

Source: The author.

In our reaction-diffusion experiments, using the model from Malheiros, Fensterseifer and Walter (2020), we found that the proportionality between fore and background is directly linked to the pattern topology. As the pattern evolves (adjusting the $ratio$ parameter from the RD model), the spot structure slowly transitions to a labyrinth, reaching a proportion close to 50% between front and back. If we continue the process, the pattern changes smoothly into inverse spots (dark spots on a light background), where most pixels will be on a light background. Figure 3.6 show the graphs for our synthetic reaction-diffusion datasets, with Figure 3.6a plotting the values for the spotted pattern images, ranging from 0.255 to 0.287, and Figure 3.6b plotting the other topologies, with values ranging from 0.393 to 0.438.

Figure 3.6: Graph plots of the descriptor *pattern area percentage* for our reaction-diffusion *Spots* and *Not Spots* (Stripes and Labyrinths) datasets.



(a) Reaction-diffusion spots

(b) Reaction-diffusion stripes and labyrinths

Source: The author

This proportion is a good indicator of the topological structure of a pattern and a good predictor for the $ratio$ parameter of the reaction-diffusion model used. The pattern area percentage measure had already been proposed by Sun *et al.* (SUN et al., 2017) as $R^1$. In our results, this measure has been one of the most significant descriptors in validation processes via automated classification.

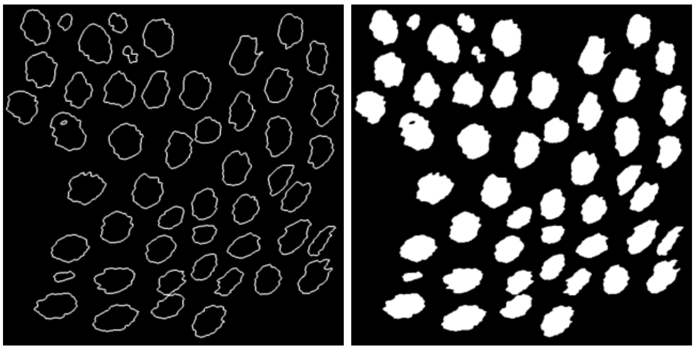
The second global descriptor is called *border pattern ratio*, inspired on the $R^2$ metric from Sun et al. (2017). We tested some variations, but the formulation that showed to be most useful is the number of pixels on the foreground border, divided by the total number of pixels in the foreground (pattern area given in pixels) after removing partial pattern regions , represented in Figure 3.7a. This descriptor may be written as

$$\frac{border_{clean}}{area_{clean}} \tag{3.4}$$

with $border_{clean}$ being the number of pixels of the pattern border, as in Figure 3.7a, and

$area_{clean}$ the pattern area, as in Figure 3.7b, both measured after removing partial regions. This descriptor roughly indicates the level of fragmentation of the image's details. It is a helpful descriptor because it generates a distinct value for irregular structures and even labyrinthine patterns when compared to the simpler borders of round spots.

Figure 3.7: Figure 3.5 after removing partial regions, with the white pixels in (a) being the border area and in (b) the pattern area. In this example the border area is 6,726 px, the pattern area is 74,144 px, and the ratio of these values gives us a *border pattern ratio* of 0.029



(a) Pattern borders (b) Pattern area

Source: The author

We also created another global descriptor based on border analysis called *border perimeter ratio*. To define this descriptor, we experimented with different ratios. These revolve mostly around using an approximated perimeter instead of the number of foreground border pixels for the ratio numerator and/or experimentation with the ratio denominator, changing it between the area ($height * width$) and the perimeter of the image. This descriptor is calculated as the ratio between the number of pixels on the foreground border (border area in pixels) and the image perimeter, also measured in pixels. It may be written as

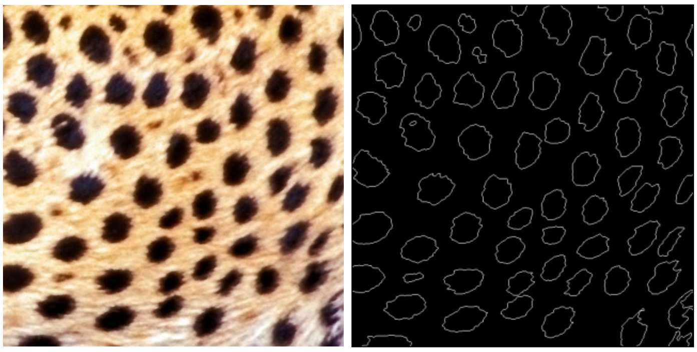
$$\frac{border_{pattern}}{perimeter_{img}} \tag{3.5}$$

where $border_{pattern}$ is the area, in pixels, of the pattern border without removing partial regions and $perimeter_{img}$ is the perimeter of the image computed as $2 * height + 2 * width - 4$. This descriptor is dimensionless and is not influenced by changes in image resolution as the border pattern ratio.

Our last global descriptor is called *pattern loss percentage*, measuring the percentage of pattern area lost by removing all regions touching the image border. This descriptor is computed as

$$1 - \frac{area_{clean}}{area_{pattern}} \tag{3.6}$$

Figure 3.8: A (a) cheetah crop and (b) its pattern borders. Using (b) we compute the number of white pixels (a border area of 8,117 px) and the perimeter of the image (530 × 4 - 4 = 2,116 px), from which we obtain an *border perimeter ratio* of 3.836.

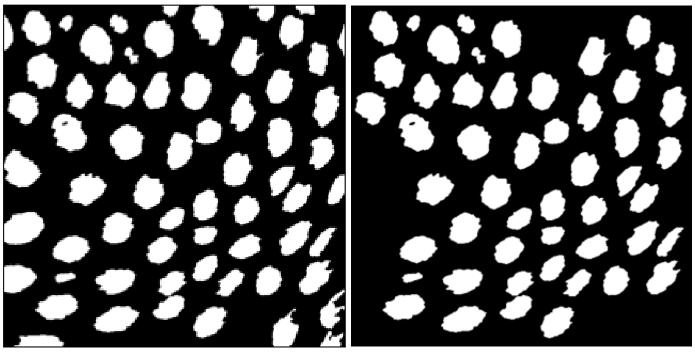


(a) Original crop (b) Pattern borders

Source: The author.

where the ratio of the pattern area after removing the partial regions ($area_{clean}$) and the pattern area before this operation ($area_{pattern}$) gives us a value ranging from 0 to 1. This value represents the remaining pattern area, and by removing it from 1 (or 100%) we obtain a percentage of pattern lost by removing partial regions. These areas are represented in Figures 3.9a and 3.9b, and by subtracting the ratio of 3.9b and 3.9a from 100% we obtain the *pattern loss percentage* descriptor, as exemplified by Figure 3.9.

Figure 3.9: The same binarized crop (a) before and (b) after the removal of partial regions. From these images we compute an pattern area of 90,022 px for (a) and 74,144 px for (b). The result for the ratio of these areas is 0.8236 (a remaining area of 82.36% after removing partial regions), and with this value we obtain 1 - 0.8236 = 0.1764, which means that (a) lost 17.64% of its pattern area by removing partial regions.

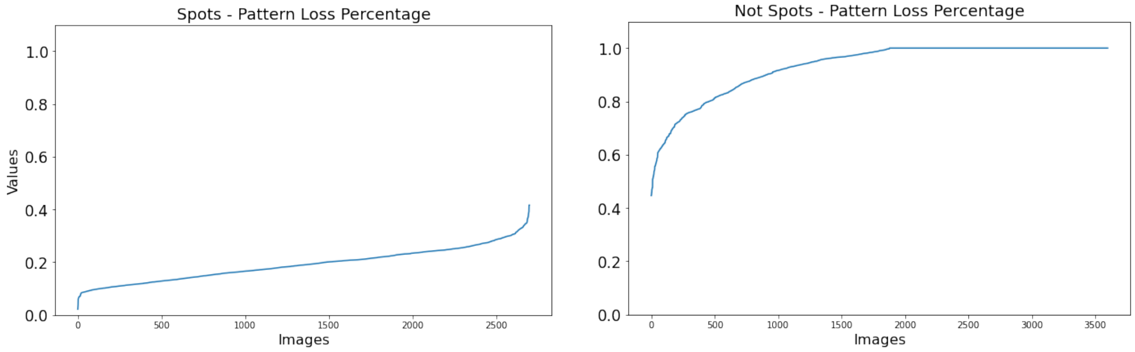


(a) Area before (b) Area after

Source: The author.

By analyzing dataset plots for this descriptor (Figure 3.10), we found that regular spot patterns lose less than 42% of their pattern area with this process while other topologies lose at least 44% of their patterns. So, by calculating a simple mean, we define that if the pattern loss percentage is less than 43%, the pattern may be defined as a regular spot pattern. This value, however, may vary with different thresholding parameters, but, for all thresholding parameters tested, all images of striped and labyrinthine topologies present a value of at least 44%. The largest value found for a spot pattern image was of 45.2%, the second largest value being 42.3%, both using $k = 0.2$ and $ws = 0.4$ for Sauvola thresholding.

Figure 3.10: Graph plots of the descriptor *pattern loss percentage* for our *Spots* and *Not Spots* (Stripes and Labyrinths) datasets, ranging from 2.2% (0.022) to 41.7% (0.417) for spot patterns and from 44.7% (0.447) to 100% (1) for other topologies.



(a) Pattern loss for spots

(b) Pattern loss for stripes and labyrinths

Source: The author

## 3.5 Local descriptors

We divide our local descriptors into two categories: pixel-based and centroid-based descriptors. The first analyzes variation in some pattern region features related to their shape. Our second category focuses on distances, taking into account only the region centroids. All local measurements consider the specific extraction of metrics for each segmented region. These individual measures are then condensed into an overall measure to generate the descriptor.

As explained earlier, we perform a second cleaning operation for all of our local descriptors when spot patterns are detected, eliminating any regions directly connected to the edges of the image (partial spots). We do not perform this operation earlier as it potentially may remove many regions of the pattern. However, after the spot structure is found, this removal provides better precision for the local descriptors, as partial spots are entirely removed.

### 3.5.1 Pixel-based local descriptors

Each of these measurements starts from the segmentation and labeling of the regions, extracting characteristics such as area, border area, equivalent-area circle radius, and eccentricity of the ellipse that has the same second-moments as the region. The first three measurements are given in pixels and their values are divided by the mean to avoid dependence on the resolution. This causes the measure zero to remain at zero and the average of the values to become 1. The eccentricity values are already measured in the [0, 1) interval, with 0 being a circle.

We then calculate the population standard deviation of the individual pattern regions values as a measure of dispersion. This procedure has proven to provide very robust descriptors. Our focus is on having a single numerical value for each descriptor over a given image. This yields the following descriptors: *region area STD*, *region border STD*, *region radius STD*, and *region eccentricity STD*. Additionally, we simply compute the mean of the region eccentricities as a new descriptor, *region eccentricity mean*, as it gives an overall estimation of region shapes.

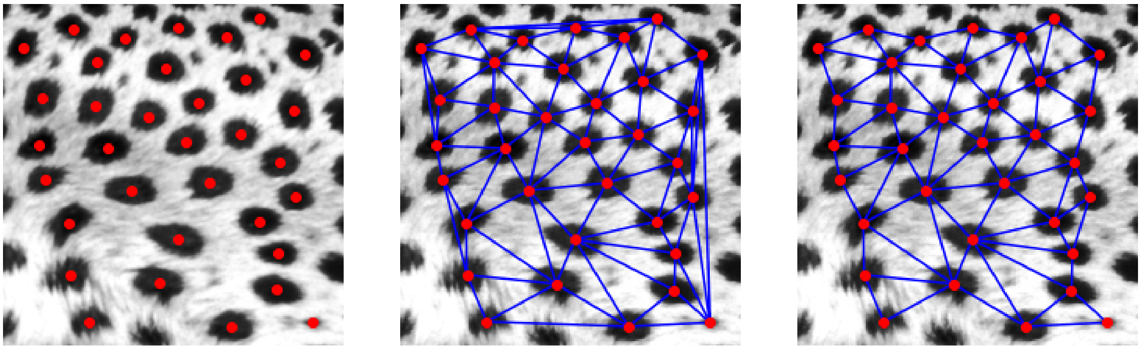
The features for the individual regions are computed based on the regions own pixel count, with the exception of two measures: radius and eccentricity. The radius value is computed based on the radius of a circle that has the same area as the region being analyzed. The eccentricity value is measure using a ellipse that has the same second-moments as the analyzed region, computed as the ratio of the distance between the focal points over the major axis length, resulting in a value in the [0, 1) interval.

Although there is a direct relationship between area, border area, and radius for circular spots, given the irregularity of the patterns, these three metrics were shown to be statistically significant and not so much correlated in the experiments made. Thus all three were kept. We also experimented with measures of orientation, but these measures were discarded as they gave different values for a same pattern with different rotations, which led to inconsistencies to the pattern analysis and validation results.

### 3.5.2 Centroid-based local descriptors

The remaining descriptors follows a more elaborate process, for which after the regions are segmented, each region centroid is calculated. For the first descriptor a Delaunay triangulation is fitted over those centers. This triangulation may generate skewed triangles at the borders, and we use the method described by Ijiri et al. (2008) to discard them, as shown in Figure 3.11. The sum of lengths of all edges in non-skewed triangles gives a measure of the distance among all nearest neighbors. These distances are then normalized by the mean, and the population standard deviation is extracted, producing the *region distance STD* descriptor. All distances for the centroid-based descriptors are computed using the Euclidean distance.

Figure 3.11: Sequence of operations for our distance descriptor.

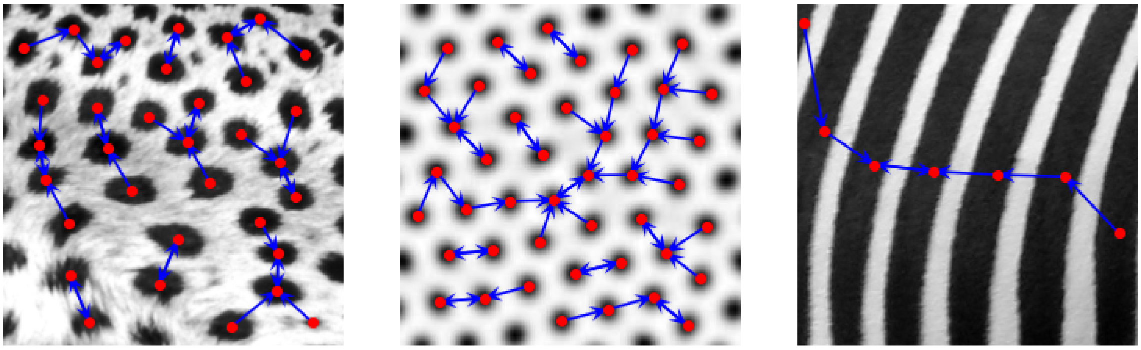


(a) Centroids (b) Delauney (c) Non-skewed triangles

Source: The author

We also use two point pattern analysis (PPA) methods as descriptors (GIMOND, 2022). While other local descriptors are normalized by the mean, both PPA methods are normalized by the average between the image height and width, as they result in a single distance value measured in pixels and conditioned to the image size.

Figure 3.12: Visual representation of the ANN method, where the red points are the region centroids and the blue arrows point to the nearest neighbor.



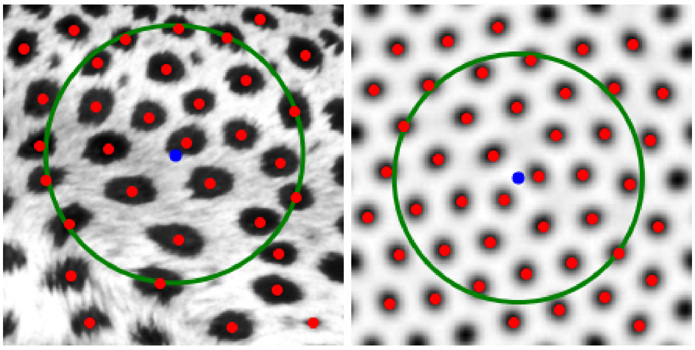
(a) Cheetah spots (b) Reaction diffusion spots (c) Zebra stripes

Source: The author

The first methods is known as average nearest neighbors (*ANN*), where, for each region centroid, we measure the distance to the nearest centroid and average these distances. This method is illustrated in Figure 3.12: in (a) a real spotted pattern, (b) a synthetic spotted pattern, and (c) an extrapolation of the idea applied to a striped pattern.

The second method is called *standard distance*, illustrated in Figure 3.13, a metric obtained by measuring a “mean center” and calculating the variance between the distances to this mean center. This method could also be divided into a *standard deviational ellipse*, giving two center distance values, one for each ax. We tested these values with our validation methods and they show little to no improvement in our results, and consequently, we discarded these measures.

Figure 3.13: Visual representation of the *standard distance* method, where the red points are the region centroids, the blue point is the mean center, and the green circle radius is the standard distance.



(a) Cheetah spots (b) Reaction diffusion spots

Source: The author

### 3.6 Summary of descriptors

In summary, our 12 descriptors are presented in Table 3.1, presenting their names, categories, and range of values for each.

Table 3.1: Summary of our 12 descriptors, each one with its name, possible range of values, and categorized into *global*, *local (pixel)* or *local (centroid)*, as defined in Sections 3.4 and 3.5.

| | Name | Category | Value Range |
|---|---|---|---|
| 1 | Pattern Area Percentage | Global | [0, 1] |
| 2 | Border Pattern Ratio | Global | (0, +) |
| 3 | Border Perimeter Ratio | Global | [0, +) |
| 4 | Pattern Loss Percentage | Global | [0, 1] |
| 5 | Region Area STD | Local (pixel) | [0, 2] |
| 6 | Region Radius STD | Local (pixel) | [0, 2] |
| 7 | Region Border STD | Local (pixel) | [0, 2] |
| 8 | Region Eccentricity Mean | Local (pixel) | [0, 1) |
| 9 | Region Eccentricity STD | Local (pixel) | [0, 1) |
| 10 | Region Distance STD | Local (centroid) | [0, 2] |
| 11 | Standard Distance | Local (centroid) | [0, $\sqrt{2}$] |
| 12 | Average Nearest Neighbors | Local (centroid) | [0, $\sqrt{2}$] |

# 4 RESULTS

To validate our descriptors we propose four tasks: a supervised classifier, an unsupervised clustering application, an unsupervised clustering-based bounding box detection, and a regressor that estimates the $ratio$ and $scale$ reaction-diffusion parameters for a given input image. We first explain the datasets we used, followed by the four validation tasks.

## 4.1 Tools

The implementations of the following sections were written using the programming language *Python 3.9.6*, with the usage of additional libraries for mathematical functions, computer vision, and data processing and visualization. These libraries are: *NumPy 1.23.0*, *Matplotlib 3.5.2*, *Scikit-learn 1.1.2*, *Scikit-image 0.19.3*, and *Pandas 1.4.3*.

Additionally, the base features for our local descriptors were extracted using the *measure.regionprops* function of the Scikit-image library. Our global descriptors were computed mainly using NumPy *count_nonzero* function, with the usage of the Scikit-image *segmentation.clear_border* and *segmentation.find_boundaries* functions when necessary to compute the descriptor.

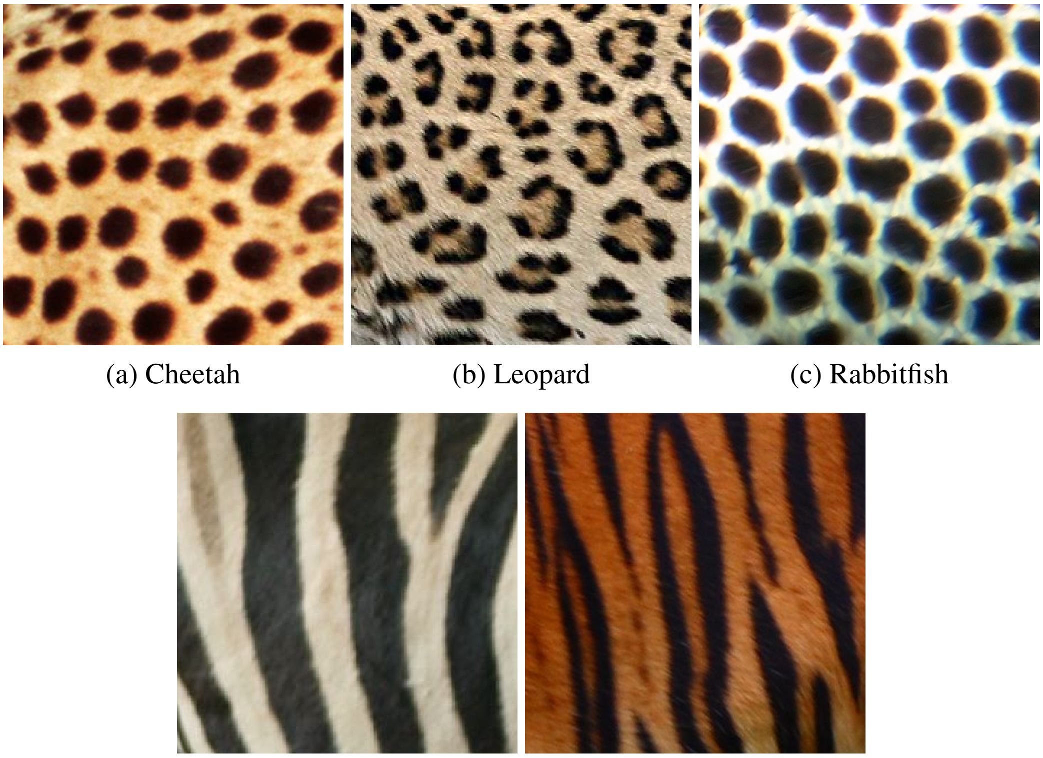
## 4.2 Datasets

For our first two validation tasks we created an input dataset containing a collection of two types of images: cropped images of real animal patterns, and results of reaction-diffusion simulations, illustrated in Figures 4.1 and 4.2. Both types are subdivided into subgroups that afterward are used as labels for the validation tasks. These subgroups are:

1. Real images of animal patterns: (i) cheetah (*Acinonyx jubatus*), (ii) leopard (*Panthera pardus*), (iii) rabbitfish (*Plectorhinchus chaetodonoides*), (iv) zebra (*Equus quagga/Equus grevyi/Equus zebra*), and (v) tiger (*Panthera tigris*).
2. Reaction-diffusion images:(i) small spots, (ii) medium spots, (iii) large spots, (iv) small stripes, (v) medium stripes, (vi) large stripes, (vii) small labyrinths, (viii) medium labyrinths, and (ix) large labyrinths.

These datasets may also be divided into three topologies: spotted, striped, and labyrinthine patterns. Each topology has both real and synthetic datasets, except for labyrinths, where we have no real animal pattern dataset. This is a rare animal pattern, making it hard to keep consistent and same-sized samples for the required dataset size.

Figure 4.1: Examples of images in our real animal pattern dataset.

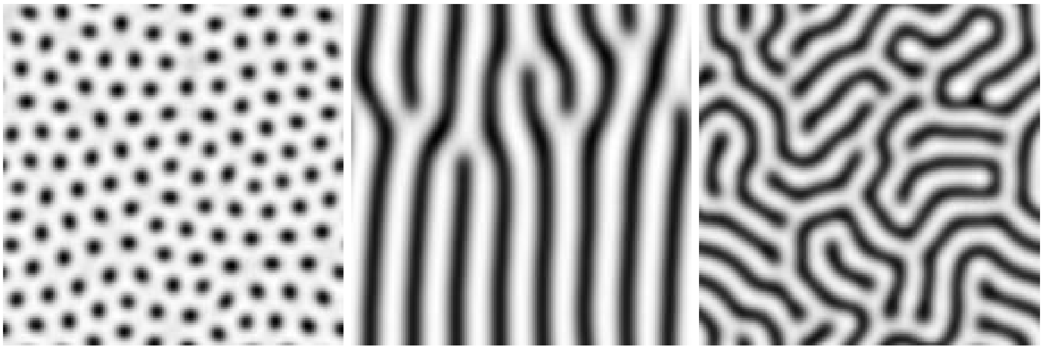


(a) Cheetah (b) Leopard (c) Rabbitfish

(d) Zebra (e) Tiger

Source: The author

Figure 4.2: Examples of images in our reaction-diffusion dataset.



(a) Small Spot (b) Medium Stripe (c) Small Labyrinth

Source: The author

For the real images, an initial raw set was downloaded from Google Images, Flickr, and Pixabay using an automatic search for copyright-free images. From this raw set, we selected the images by analyzing their resolution and the animal's orientation on the image, searching for well-defined patterns. The final pattern images were manually cropped with at least 150x150 pixels resolution. We tried to avoid distortions in the pattern and background presence in this cropping process whenever possible. Our crops also have the pattern regions visually detached from each other and cover as much of the image as possible.

The reaction-diffusion images were computed using the pattern generation model presented by Malheiros, Fensterseifer and Walter (2020), with a fixed $ratio$ value for images of the same topology, changing the $scale$ parameter to obtain different pattern sizes while keeping the same topology. All images were generated at 150x150 resolution. The spotted and labyrinthine patterns were created using an isotropic kernel with different simulation parameters, while the stripped patterns used an anisotropic kernel, all topologies using the resulting matrix $a$ as the dataset image, as explained in Section 4.2.1.

All of our images are two-colored "black" patterns and "white" backgrounds, except for the leopard patterns, which have spots of two colors. We chose to keep a leopard dataset duo to the thresholding process focusing only on the black part of the rosettes, ignoring their brown-colored interior. Each subgroup has 150 images plus two downsampled copies, one with $75\%$ and the other with $50\%$ of the original resolution, for a total of 450 images for each subgroup (thus having 2,100 unique images, with a total of 6,300 images). The resized images are used to analyze the influence of a low input resolution on the descriptors.

### 4.2.1 Synthetic patterns

For the simulation of our synthetic patterns, the RD reagents $a$ and $b$ are represented as two discrete matrices of the same size, one starting with all values set to a fixed value and the other to a pseudo-random state based on a seed. We solve the PDEs numerically with the Euler method, also known as *forward Euler method*, given by

$$y_{n+1} = y_n + hf'(y_n, t_n) \tag{4.1}$$

where $y_n$ is the PDE result at a given time $t_n$ and with a step $h$ (sometimes written as $\Delta_t$) multiplied by its derivative $f'$. There are other more complex methods that obtain results with increased precision, such as Runge Kutta methods (LINGE; LANGTANGEN, 2020), but we chose to use the Euler method due to presenting similar results in comparison to other methods and having a simpler implementation.

The resolution of Equations 2.3 and 2.4 with Euler method gives us

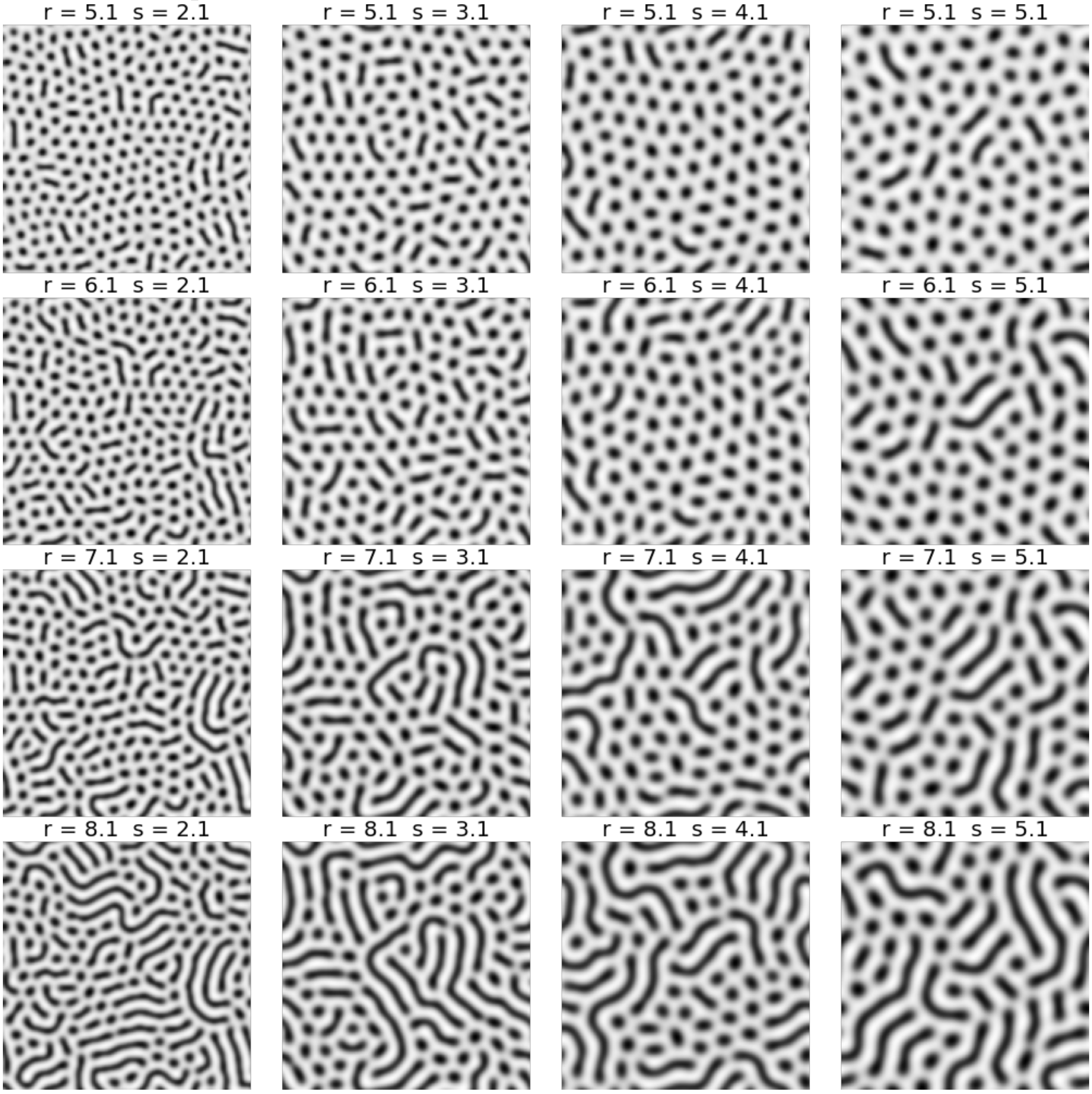
$$a_{n+1} = a_n + h(16 - a_n b_n + rs\nabla^2 a_n) \tag{4.2}$$

and

$$b_{n+1} = b_n + h(a_n b_n - b_n - 12 + s\nabla^2 b_n) \tag{4.3}$$

where $a_n$ and $b_n$ the current state of the matrices and $a_{n+1}$ and $b_{n+1}$ are their next state.

Figure 4.3: Example of pattern change with the variation of $ratio$ $(r)$, going from 5.1 to 8.1 in the vertical axis, and $scale$ $(s)$, ranging from 2.1 to 5.1 in the horizontal axis, using the kernel of Equation 4.4, and $h = 0.0075$ with 10000 iterations.



Source: The author

As stated above, the diffusion rates are derived from the $ratio$ and $scale$ parameters ($rs$ for reagent $a$ and $s$ for $b$). The influence of these parameters on the final pattern is exemplified in Figure 4.3, where a variation in $ratio$ changes the overall topology while a variation in $scale$ influences the size of the pattern.

We obtain an approximation of the Laplacian $\nabla^2 a$ and $\nabla^2 b$ as a convolution of the discrete matrices $a_n$ and $b_n$ by a given kernel. This kernel may be isotropic, with the diffusion happening at the same rate in every direction, or anisotropic, where the diffusion does not happen evenly. An isotropic kernel[1], such as

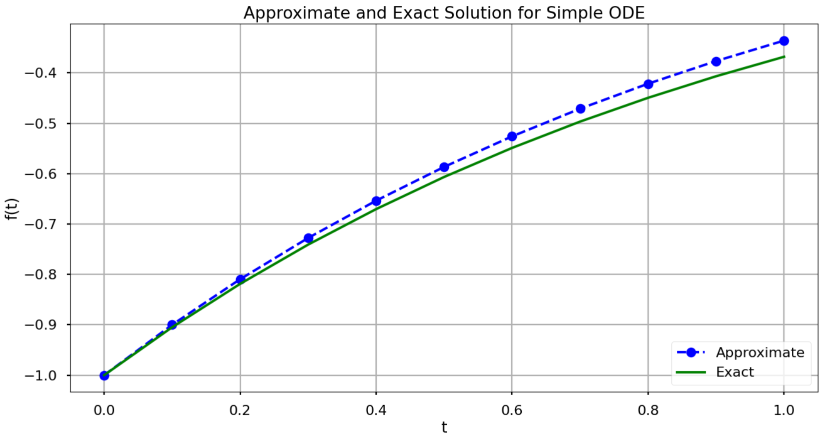
$$K_{iso} = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}, \tag{4.4}$$

results on a regular spotted or labyrinthine pattern, while using an anisotropic kernel, for example

$$K_{aniso} = \begin{bmatrix} 0.135 & 0.249 & 0.135 \\ 0.539 & -2.116 & 0.539 \\ 0.135 & 0.249 & 0.135 \end{bmatrix}, \tag{4.5}$$

results on a striped pattern, as shown in Figure 4.6.

Figure 4.4: Approximation of $f(t) = -e^{-t}$ solved with *forward Euler* compared to the actual solution.



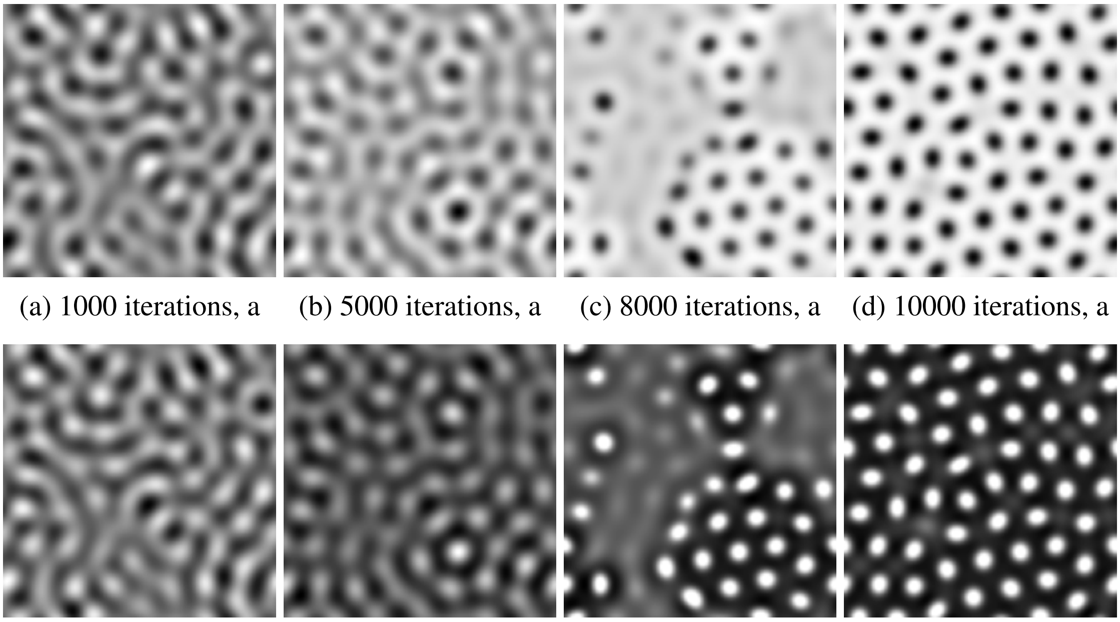
Source: Kong, Siauw and Bayen (2020)

The last parameter for the PDE resolution is the step $h$, a value directly linked to the error on the approximation results, where smaller values represent smaller errors but need a larger number of iterations the get to the desired solution. The graph presented in

[1] from (MALHEIROS; FENSTERSEIFER; WALTER, 2020)

Figure 4.4 shows the difference between an approximation of $f(t) = -e^{-t}$ using Euler method with $h = 0.1$ and the actual solution.

Figure 4.5: Simulation steps for the matrices of a reagent $a$ and a reagent $b$, using $ratio = 4.1$, $scale = 8.8$, $h = 0.01$ up to 10000 iterations, and a isotropic kernel.

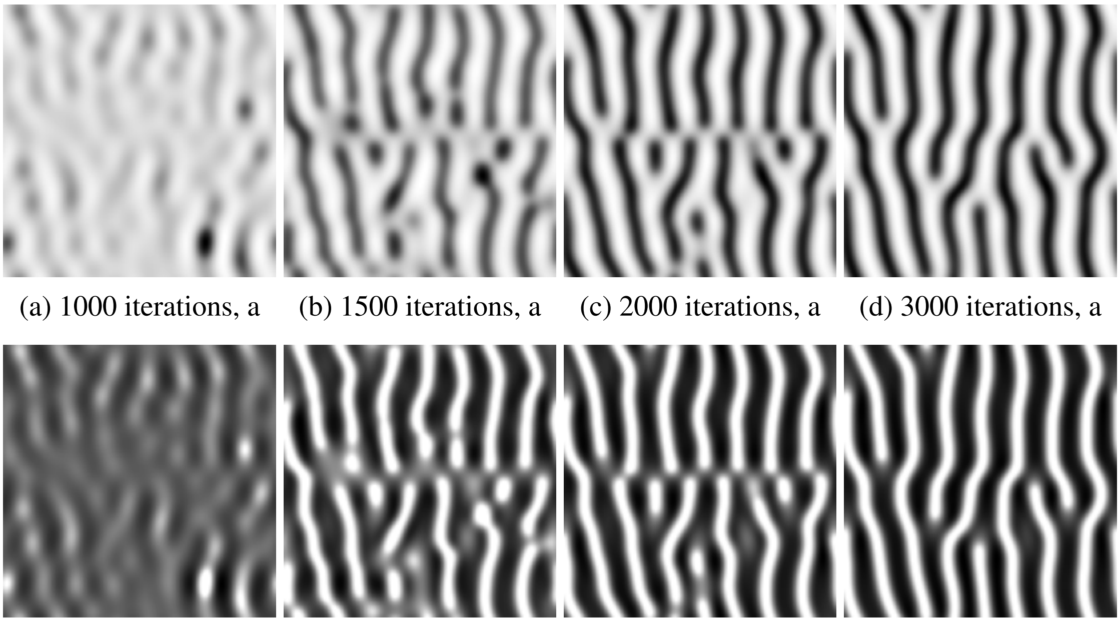


(a) 1000 iterations, a (b) 5000 iterations, a (c) 8000 iterations, a (d) 10000 iterations, a

(e) 1000 iterations, b (f) 5000 iterations, b (g) 8000 iterations, b (h) 10000 iterations, b

Source: The author

Figure 4.6: Simulation steps for the matrices of a reagent $a$ and a reagent $b$ using an anisotropic kernel (Equation 4.5) and $h = 0.075$ with up to 3000 iterations.



(a) 1000 iterations, a (b) 1500 iterations, a (c) 2000 iterations, a (d) 3000 iterations, a

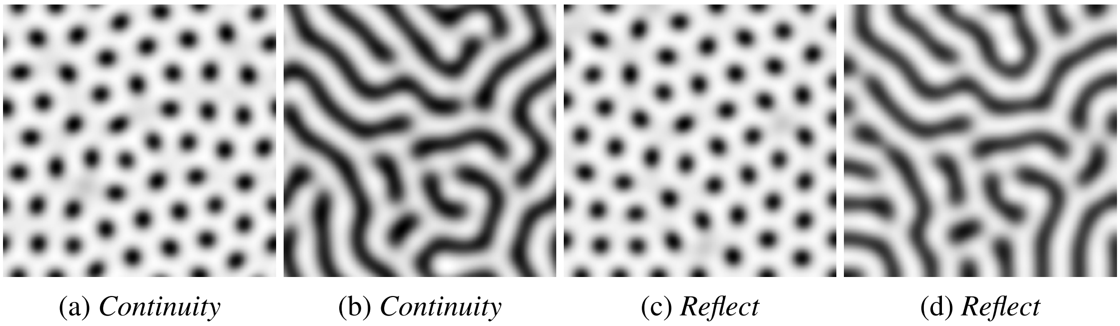
(e) 1000 iterations, b (f) 1500 iterations, b (g) 2000 iterations, b (h) 3000 iterations, b

Source: The author

Figure 4.5 shows simulation steps for matrices $a$ and $b$, up to 10000 iterations with step $h = 0.01$, with a ratio of 4.1, scale of 8.8, and using an isotropic kernel, generating a spotted pattern. The end result is a grayscale interpretation, normalized to a [0, 255] range, such as Figures 4.5d and 4.5h.

The simulation may use different conditions for diffusion on the matrix's borders, such as reflection on the borders or continuity on the $x$ and $y$ axes. Figure 4.7 shows patterns created using these different border conditions, where Figures 4.7a and 4.7c, and Figures 4.7b and 4.7d are generated using the same parameters.

Figure 4.7: Example of the influence of the border conditions showing the end result of matrix $a$.



(a) *Continuity* (b) *Continuity* (c) *Reflect* (d) *Reflect*

Souce: The author

### 4.3 Classification

Our first validation method is a supervised classifier using our descriptors as features for a Random Forest (HO, 1995) learning model. We created seven dataset test setups with the image subsets of Section 4.2, classifying each image as part of one of the subsets present on the setup, that is, each image receives a label. These setups are: *All images*, *Real patterns* (animal patterns or non-synthetic patterns), *Spotted animal patterns* (Real patterns setup excluding zebras and tigers), *Spot patterns* (Spotted animal setup plus synthetic spotted patterns), *Not spot patterns* (striped and labyrinthine patterns), *Stripe patterns* (zebras, tiger, and synthetic stripes) and *Reaction-diffusion (RD) patterns*. They are summarized in Table 4.1, presenting each setup together with the image subsets present on them, where the RD subsets present small, medium, and large scale patterns. For our tests, the setup data was divided into training and test sets with 70% and 30% of the setup images, respectively. This division is done using a stratified sampling, where each possible label has the number of images on the training and test set relative to the total number of images of the label. In our case, all labels have the same number of images, due to all image subsets being the same size.

Table 4.1: Summary of the test setups and the image subsets present in each one.

| **Setup** | **Subsets (Labels)** |
|---|---|
| All | All subsets |
| Real | Cheetah, Leopard, Rabbitfish, Zebra, Tiger |
| Real Spots | Cheetah, Leopard, Rabbitfish |
| Spots | Cheetah, Leopard, Rabbitfish, RD Spots |
| Not Spots | Zebra, Tiger, RD Stripes, RD Labyrinths |
| Stripes | Zebra, Tiger, RD Stripes |
| RD | RD Spots, RD Stripes, RD Labyrinths |

Besides the Random Forest classifier, we also tested three other ML classification methods: *Support Vector Classifier* (SVC), *K-nearest Neighbors* (KNN), and Decision Trees. Table 4.2 shows the mean accuracy scores of 100 tests obtained for the *All images* setup for each method. These results show that the random forest method presents consistently better results than other methods, further validating the choice of this method.

Table 4.2: Comparison of the mean accuracy score for 100 tests for each classification method tested.

| **Method** | **Accuracy** | **Accuracy (NC)** |
|---|---|---|
| SVC | 89.091% | 81.056% |
| KNN | 93.968% | 91.148% |
| Decision Tree | 97.025% | 94.313% |
| Random Forest | 98.524% | 96.933% |

After testing different descriptor combinations, we defined a *default* setup that includes all the descriptors present in Table 3.1. This descriptor setup was chosen since we obtained the overall best results for all test setups. All dataset setups were tested with and without the downsampled copies. We then calculate three scores: f1, precision, and accuracy. Table 4.3 shows the mean scores for 15 different training/test splits, with the sets including the downsampled images. The results including them are shown in Table 4.4.

Based on the results obtained, our descriptors were shown to be capable of differentiating between different pattern types and, despite not as efficiently, distinguishing different real animal patterns. This may be conclude since the results for real patterns are considerably lower than those mixing real and synthetic patterns or the RD setup. The *Real*, *Spots*, and *Spotted animals* setups presented both the highest score drop when removing the downsampled images and the overall lowest scores, maybe due to the similarities between the patterns. Meanwhile, the *Not Spots* and *Stripes* setups show consistent

Table 4.3: Classifier mean scores of 15 tests, for each test setup, using our default descriptors without the downsampled images.

| Setup | f1 | precision | accuracy |
|---|---|---|---|
| All | 96.983% | 96.984% | 97.040% |
| Real | 91.965% | 91.970% | 92.144% |
| Spotted animal | 87.885% | 87.901% | 88.158% |
| Spots | 94.071% | 94.074% | 94.238% |
| Not Spots | 98.997% | 99.000% | 99.019% |
| Stripes | 98.726% | 98.726% | 98.755% |
| RD | 99.984% | 99.984% | 99.984% |

Table 4.4: Classifier mean scores of 15 tests, for each test setup, using our default descriptors setup including downsampled images.

| Setup | f1 | precision | accuracy |
|---|---|---|---|
| All | 98.414% | 98.413% | 98.447% |
| Real | 96.476% | 96.474% | 96.546% |
| Spotted animal | 94.712% | 94.713% | 94.834% |
| Spots | 97.009% | 97.008% | 97.095% |
| Not Spots | 99.542% | 99.543% | 99.545% |
| Stripes | 99.417% | 99.417% | 99.425% |
| RD | 99.951% | 99.951% | 99.951% |

high scores, which could be explained in the view of the differences between stripes and labyrinths, and pattern differences inside the same topologies, for instance the difference between tiger and zebra stripes.

We also compared our results with pre-trained general purpose deep learning (DL) models available through *Keras* (CHOLLET et al., 2015). These models return a classification rank based on a confidence score, from which we considered the Top 1 labels. For this comparison, we calculated the accuracy for a new setup created with the common labels between our dataset and *ImageNet*. These labels are our real patterns datasets excluding the rabbitfish set, that is, cheetahs, leopards, zebras, and tigers. To obtain a "fair" comparison we tested the DL models using the same images used for the classifier tests.

The tested models are: VGG16 and VGG19 (SIMONYAN; ZISSERMAN, 2014), ResNet152V2 (HE et al., 2016), DenseNet121 and DenseNet201 (HUANG et al., 2017), Xception (CHOLLET, 2017), InceptionResNetV2 (SZEGEDY et al., 2017), NASNet-Large (ZOPH et al., 2018), and EfficientNetV2L (TAN; LE, 2021).

Table 4.5 shows the comparison between our classifier, with a mean accuracy of 500 tests, and the tested DL models. In this table, the column *Accuracy* display the results including the downsampled images, and the columns containing (NC) show results without these images. As shown, for both scenarios our classifier obtained a better accuracy than the pre-trained general purpose DL models, even though the best DL models achieved similar results when removing the downsampled images.

Table 4.5: Comparison of some deep learning classifier models precision and the mean accuracy of 500 tests for our results. The columns with (NC) present the results without the downsampled images.

| **Model** | **Accuracy** | **Accuracy (NC)** |
|---|---|---|
| VGG16 | 60.556% | 72.778% |
| VGG19 | 68.333% | 78.889% |
| DenseNet121 | 69.444% | 80.000% |
| DenseNet201 | 74.630% | 81.667% |
| ResNet152V2 | 84.815% | 83.333% |
| EfficientNetV2L | 85.000% | 93.889% |
| Xception | 93.148% | 90.556% |
| NASNetLarge | 94.444% | 94.444% |
| InceptionResNetV2 | 94.630% | 94.444% |
| Random Forest | **97.719%** | **94.997%** |

Last, we measured our descriptors' *feature importance* (Gini importance) for each Random Forest classification setup. Table 4.6 shows our descriptors' importance for one test using the *All images* setup sorted from most to least important, without including the downsampled images. To test the influence of resolution on our descriptors, Table 4.7 shows the score for the same test as Table 4.6 while including the downsampled images.

Table 4.6: Feature importance for one test using our *All images* setup, without the downsampled images, sorted by importance.

| **descriptor** | **importance** |
|---|---|
| border pattern ratio | 19.87% |
| border perimeter ratio | 17.51% |
| pattern area percentage | 13.47% |
| region eccentricity mean | 11.40% |
| pattern loss percentage | 7.92% |
| ANN | 7.75% |
| region border STD | 5.17% |
| region area STD | 4.81% |
| region radius STD | 4.66% |
| region distance STD | 2.90% |
| standard distance | 2.75% |
| region eccentricity STD | 1.79% |

Noticeable differences are the *border pattern ratio*, dropping from 19.87% to 7.82%, since it is highly influenced by the image resolution as consequence of how it is computed, dividing a measure of perimeter (given in px) by a measure of area (given in px$^2$). Another difference worth commenting is how the *ANN* descriptor went from a score

Table 4.7: Feature importance for the same test as Table 4.6, including the downsampled images, sorted by importance.

| descriptor | importance |
|---|---|
| border perimeter ratio | 24.11% |
| pattern area percentage | 14.27% |
| ANN | 11.49% |
| region eccentricity mean | 10.84% |
| pattern loss percentage | 8.32% |
| border pattern ratio | 7.82% |
| region border STD | 5.09% |
| region area STD | 4.96% |
| region radius STD | 4.59% |
| standard distance | 3.21% |
| region distance STD | 2.95% |
| region eccentricity STD | 2.36% |

of 7.75% to 11.49%, likely due to it consistently measuring distances independently of resolution.

Overall, the border perimeter ratio descriptor was the most relevant descriptor for most of our tests, followed by the pattern area percentage and ANN descriptors. Our pattern loss percentage descriptor was also appropriate when the test setup included spots together with other topologies. The least relevant descriptors were the standard distance and region eccentricity STD. However, even though these descriptors were the least relevant, our tests got better results than those without them.

## 4.4 Clustering

The second validation method used unsupervised clustering. For these tests, we used the Gaussian Mixture method (GÉRON, 2019) with our default descriptor setup as training attributes, based on the results from the classifier. We created test setups for the clustering application using the entire dataset and some subsets, similar to the classifier setups (i.e., real images only or reaction-diffusion only) and a different number of clusters, trying to achieve a good balance in the number of the clusters and their internal homogeneities, such as clusters of real and synthetic images, clusters of different individual species, and clusters for different topologies.

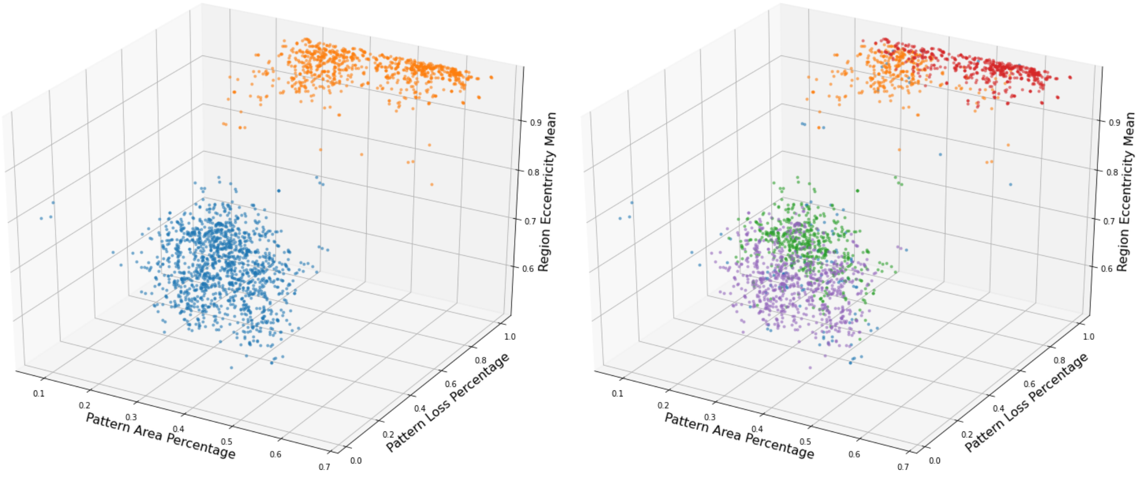
After identifying the clusters, we found, for each cluster, its predominant dataset and then calculated a percentage of error based on the number of other datasets in that same cluster, making an ad-hoc measure of cluster *homogeneity*. Another metric used is a measure of *completeness*, calculated for each dataset present on a cluster and computed as the percentage of images of the dataset present in the cluster. Both metrics give us a measure of cluster consistency. The first relates to labeling consistency, that is, not mixing different categories into the same cluster, while the measure of completeness offers an idea of category consistency for each dataset present on the cluster, specifically, consistency in grouping images of the same category together.

As an example, a cluster with a single image will always have $100\%$ homogeneity but has little completeness for the only dataset it includes, while a cluster with $n$ complete datasets will always have $100\%$ completeness for all $n$ sets but $\frac{100\%}{n}$ homogeneity. A "perfect cluster" would be the one scoring $100\%$ homogeneity and completeness, that is, a cluster of a single complete dataset. We also generated 2D and 3D scatter plots using the most important descriptors for better visualization of the results.

Figure 4.8a shows the plot for a test where we divided the *Real* images test setup, dividing it into two topologies clusters. In this test, we identified a "spots cluster" and a "stripes cluster" containing all the images of their topology. Then we tested the same setup with five clusters, one for each species (Figure 4.8b). For these tests we obtained good results for the striped patterns, with a well-defined tiger cluster containing 318 images out of 450 ($93.53\%$ homogeneity and $70.67\%$ of completeness), and a zebra cluster containing 424 images out of the 450 total ($78.08\%$ homogeneity and $94.22\%$ of completeness). As for the spotted patterns, while we obtained a cheetah cluster and a leopard cluster with high completeness ($82.89\%$ and $86.89\%$, respectively), both contained most of the rabbitfish images, 310 images for the cheetah cluster and 102 for the leopard cluster. The remaining 38 images were grouped into a last cluster, together with 31 leopard images, 28 cheetahs, 13 tigers, and 4 zebras.

Table 4.8 presents the results of a test using our "All images" setup with 14 clusters. As seen, this test presents problems such as the zebra cluster, which has homogeneity of $46.91\%$ due to the presence of 363 large reaction-diffusion stripes ($40.83\%$ of the cluster and $80.67\%$ of the dataset). Another notable problem is the fact that the small reaction-diffusion spots dataset was split into two clusters, both with $100\%$ homogeneity. Nevertheless, we also achieved clusters with high homogeneity and completeness, that is, clusters close to a perfect cluster, such as most synthetic pattern clusters (clusters 1, 3, 4, 6, 7, 8, and 11) and the cheetah cluster (cluster 0). The results obtained without the downsampled images are shown in Table 4.9, where is notable the presence of a small cluster with 15 images total (cluster 7), where the tiger label presents a relatively high

Figure 4.8: Our *Real images* setup divided into (a) two clusters, one for striped animals and another for spotted animals, and (b) five clusters, one for each species.



(a) 2 clusters

(b) 5 clusters

Source: The author

homogeneity but just 6% completeness.

Table 4.8: All clusters obtained for the *All images* setup using 14 clusters, one for each subset. The column *size* shows the total number of images in the cluster, *label* the predominant subset of the cluster (represented as *topology/topology subset*), *images* displays the number of images in the cluster that belong to *label*, and *homogeneity* and *completeness* are the metrics defined at the start of the section.

| Cluster | Size | Label | Images | Homogeneity | Completeness |
|---|---|---|---|---|---|
| 0 | 426 | Spots/Cheetah | 385 | 90.38% | 85.56% |
| 1 | 385 | Stripes/RD Medium | 385 | 100.00% | 85.56% |
| 2 | 889 | Stripes/Zebra | 417 | **46.91%** | 92.67% |
| 3 | 448 | Spots/RD Large | 448 | 100.00% | 99.56% |
| 4 | 449 | Stripes/RD Small | 449 | 100.00% | 99.78% |
| 5 | 304 | Spots/Leopard | 250 | 82.24% | 55.56% |
| 6 | 450 | Labyrinthine/RD Small | 450 | 100.00% | 100.00% |
| 7 | 450 | Spots/RD Medium | 450 | 100.00% | 100.00% |
| 8 | 449 | Labyrinthine/RD Medium | 448 | 99.78% | 99.56% |
| 9 | 621 | Spots/Rabbitfish | 382 | 61.51% | 84.89% |
| 10 | 150 | Spots/RD Small | 150 | 100.00% | **33.33%** |
| 11 | 522 | Labyrinthine/RD Large | 443 | 84.87% | 98.44% |
| 12 | 457 | Stripes/Tiger | 355 | 77.68% | 78.89% |
| 13 | 300 | Spots/RD Small | 300 | 100.00% | 66.67% |

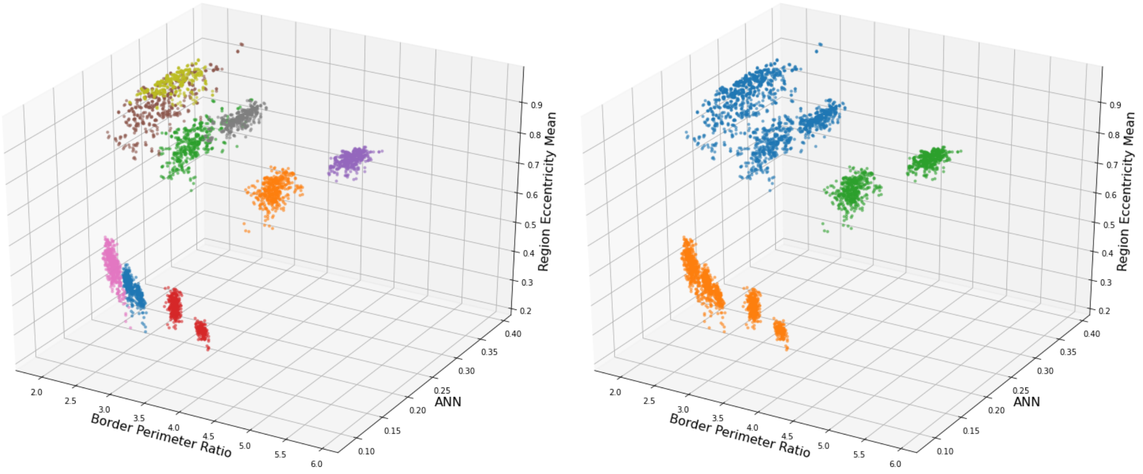
As for synthetic patterns, Figure 4.9a shows the plot where the *RD* setup was divided into nine clusters, one for each topology and pattern region size. Of the nine clusters, two clusters had less than 100% homogeneity: a large labyrinth cluster, with 466 correct images (96.13%) and 19 large stripes images (4.08%), and a large stripes cluster with 414 correct images (99.52%) and two large labyrinth images (0.48%). Without the

Table 4.9: Results for the same test setup as Table 4.8 without the downsampled images.

| Cluster | Size | Label | Images | Homogeneity | Completeness |
|---|---|---|---|---|---|
| 0 | 150 | Spots/RD Medium | 150 | 100.00% | 100.00% |
| 1 | 225 | Spots/Rabbitfish | 131 | 58.22% | 87.33% |
| 2 | 202 | Stripes/Tiger | 139 | 68.81% | 92.67% |
| 3 | 104 | Spots/Leopard | 86 | 82.69% | 57.33% |
| 4 | 197 | Labyrinthine/RD Large | 148 | 75.13% | 98.67% |
| 5 | 150 | Spots/RD Small | 150 | 100.00% | 100.00% |
| 6 | 150 | Stripes/RD Small | 150 | 100.00% | 100.00% |
| 7 | 15 | Stripes/Tiger | 9 | 60.00% | **6.00%** |
| 8 | 150 | Spots/RD Large | 150 | 100.00% | 100.00% |
| 9 | 150 | Labyrinthine/RD Small | 150 | 100.00% | 100.00% |
| 10 | 210 | Stripes/Zebra | 105 | **50.00%** | 70.00% |
| 11 | 117 | Spots/Cheetah | 107 | 91.45% | 71.33% |
| 12 | 150 | Labyrinthine/RD Medium | 150 | 100.00% | 100.00% |
| 13 | 130 | Stripes/RD Medium | 130 | 100.00% | 86.67% |

downsamples, the large stripes cluster contained 142 correct images (99.3% homogeneity and 94.67% completeness) with one large labyrinth image, while the large labyrinth cluster had 149 correct images (94.9% homogeneity and 99.33% completeness) and the remaining 8 large stripes images. When grouped into three clusters, this setup divides the spotted patterns into one cluster and the not-spotted patterns into the other two based on pattern size rather than by topology, with one cluster for small stripes and labyrinths, and another for large and medium patterns.

Figure 4.9: Our *RD* setup divided into (a) nine clusters, one for each topology and pattern size, and (b) 3 clusters, one for spotted patterns, one with not-spotted small patterns, and the last containing not-spotted medium and large patterns.



(a) 9 clusters

(b) 3 clusters

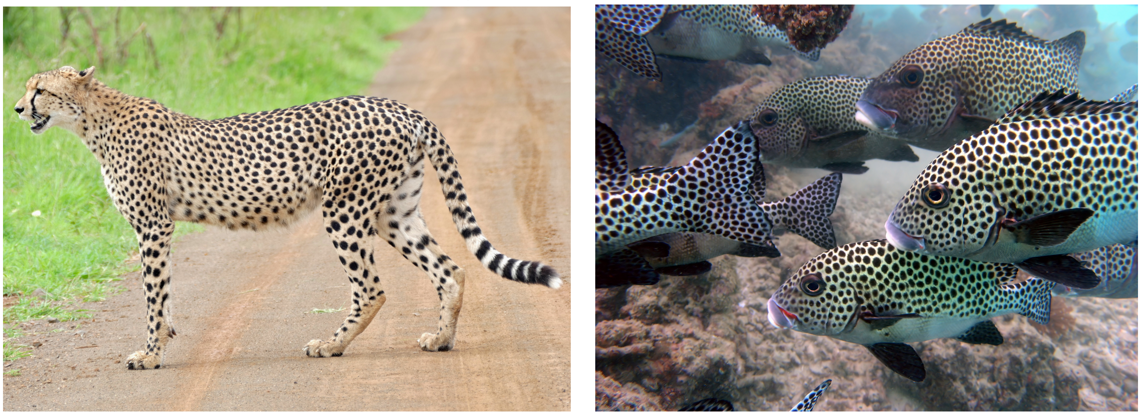
Source: The author

Based on the tests, we believe that our descriptors can distinguish between real and synthetic patterns, between different topologies, and, to some extent, between different subsets of real and synthetic patterns.

### 4.5 Bounding Box Detection

Another validation method also uses the idea of clusterization, but in a different context, analyzing whole animal images instead of cropped patterns. For this application, we divide an animal photo into a grid of ten sub-images in the smaller dimension or in a way that these sub-images are at a resolution of at least 150x150 pixels each. Then, we group these sub-images into three clusters using the same method as in Section 4.4. One cluster usually groups well-defined biologic patterns (our focus), another cluster sub-images with little detail (usually the background), and a third, high frequency and contrasting images with no discernible structure (everything else). We find the biologic patterns cluster (target cluster) by defining it to be the same as the one from a given target image.

Figure 4.10: Example of images used for the bounding box validation task.



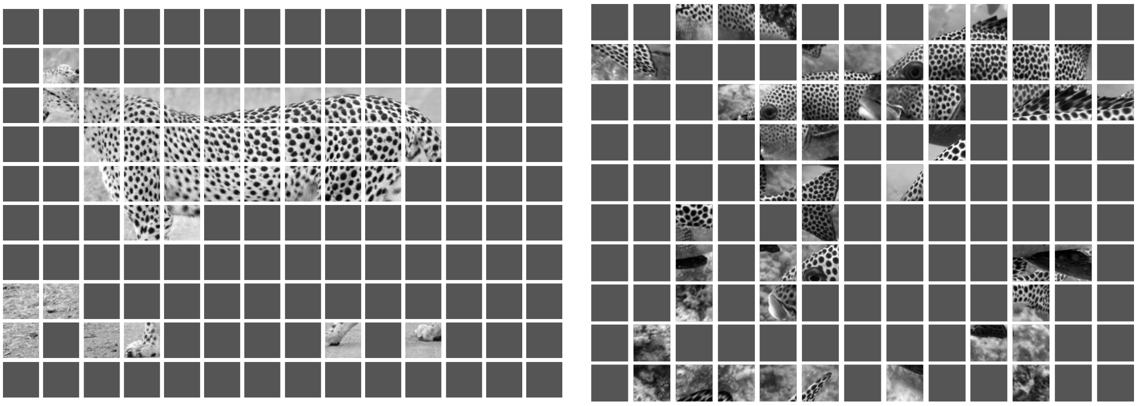
(a) Cheetah, photo by Bernard Dupont (Flickr, CC BY-SA 2.0)

(b) Rabbitfishes, photo by Jens Petersen (Wikimedia, CC BY 2.5)

We obtain different results based on different descriptor setups, and testing out default setups we obtained the clusters shown in Figure 4.11. By removing the ANN descriptor we achieved satisfying results, as those from Figures 4.12 and 4.13.

Still, using this default setup excluding the ANN descriptor do not always gives us the best results, as shown in Figure 4.14, where Figure 4.14b, not using the ANN descriptor, presents more “wrong” sub-images grouped as biological pattern than Figure 4.14c, using all default descriptors. However, both descriptor setups share some wrong sub-images, from which we may conclude that none is the optimal setup for this image.

Figure 4.11: Results obtained using all descriptors as clustering features for the images of Figure 4.10, where the shown sub-images are grouped as our target biologic pattern.

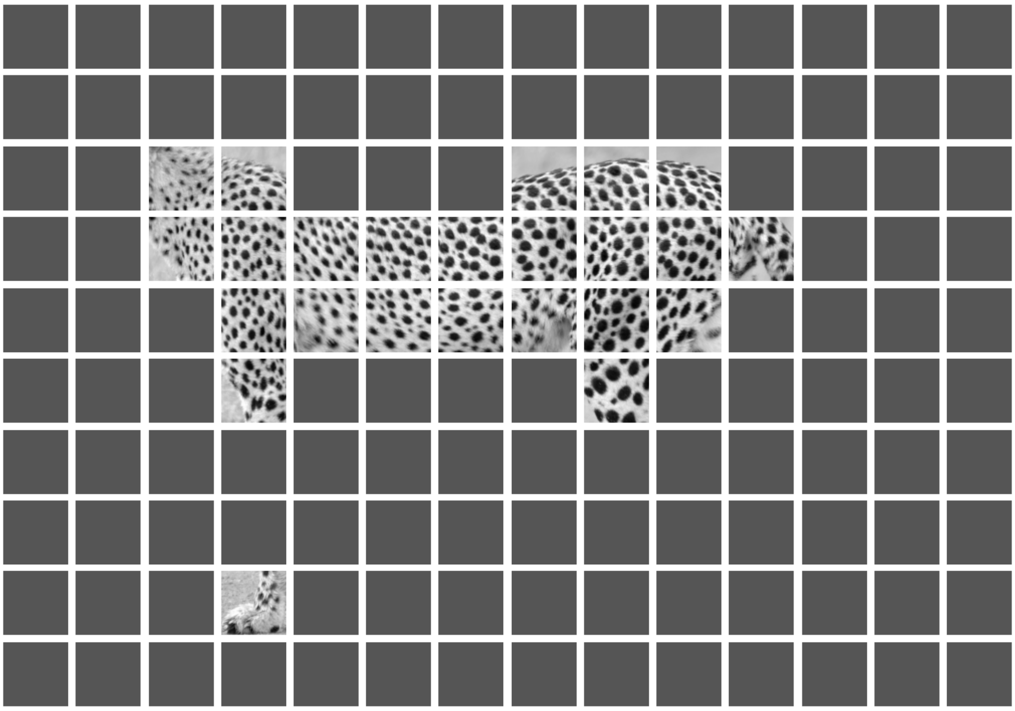


(a) Cheetah patterns mixed with background noise (282x282 pixels, grid of 10x14)

(b) Rabbitfish patterns mixed with background noise (210x210 pixels, grid of 10x13)

Source: The author

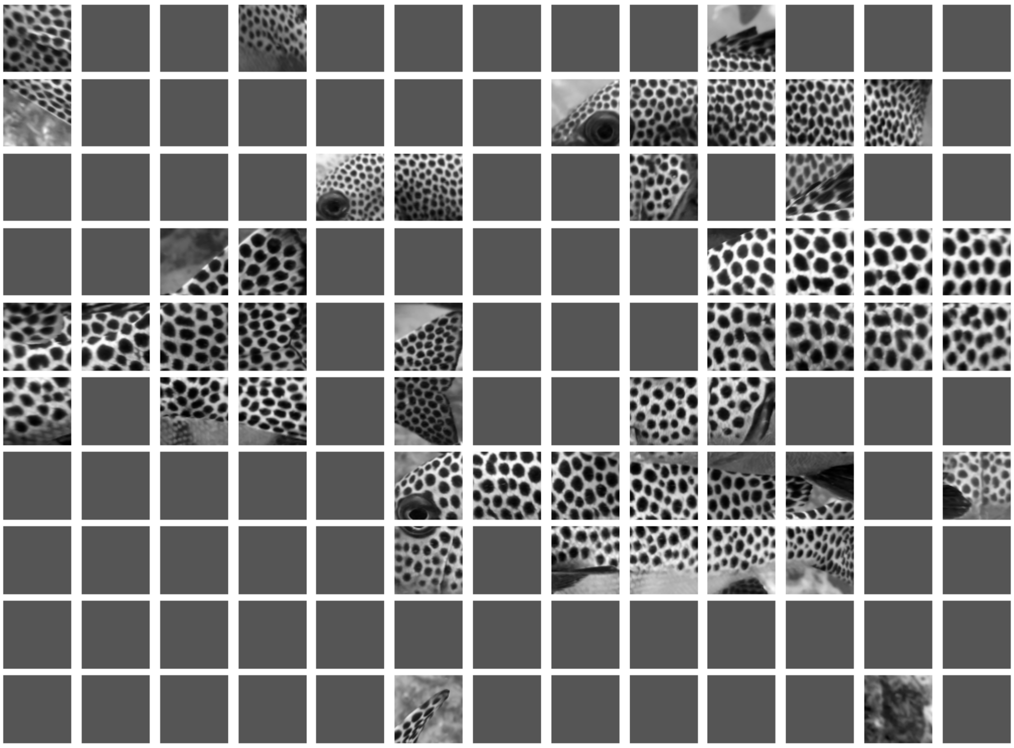
Figure 4.12: Grid for Figure 4.10a showing the sub-images grouped as biologic patterns using all descriptors except ANN.



Source: The author

Based on these results, we believe that our descriptors are, at least to some extent, able to distinguish between regular biologic and irregular background patterns. Still, some sub-images that could be part of the pattern cluster were classified as background and, in some tests, background patterns were grouped together with biological patterns. Also, the arbitrary grid size influences the pattern present in the sub-images, therefore changes in these values could improve the results. This process thus provides an unsupervised

Figure 4.13: Grid for Figure 4.10b showing the sub-images grouped as biologic patterns using all descriptors except ANN.



Source: The author

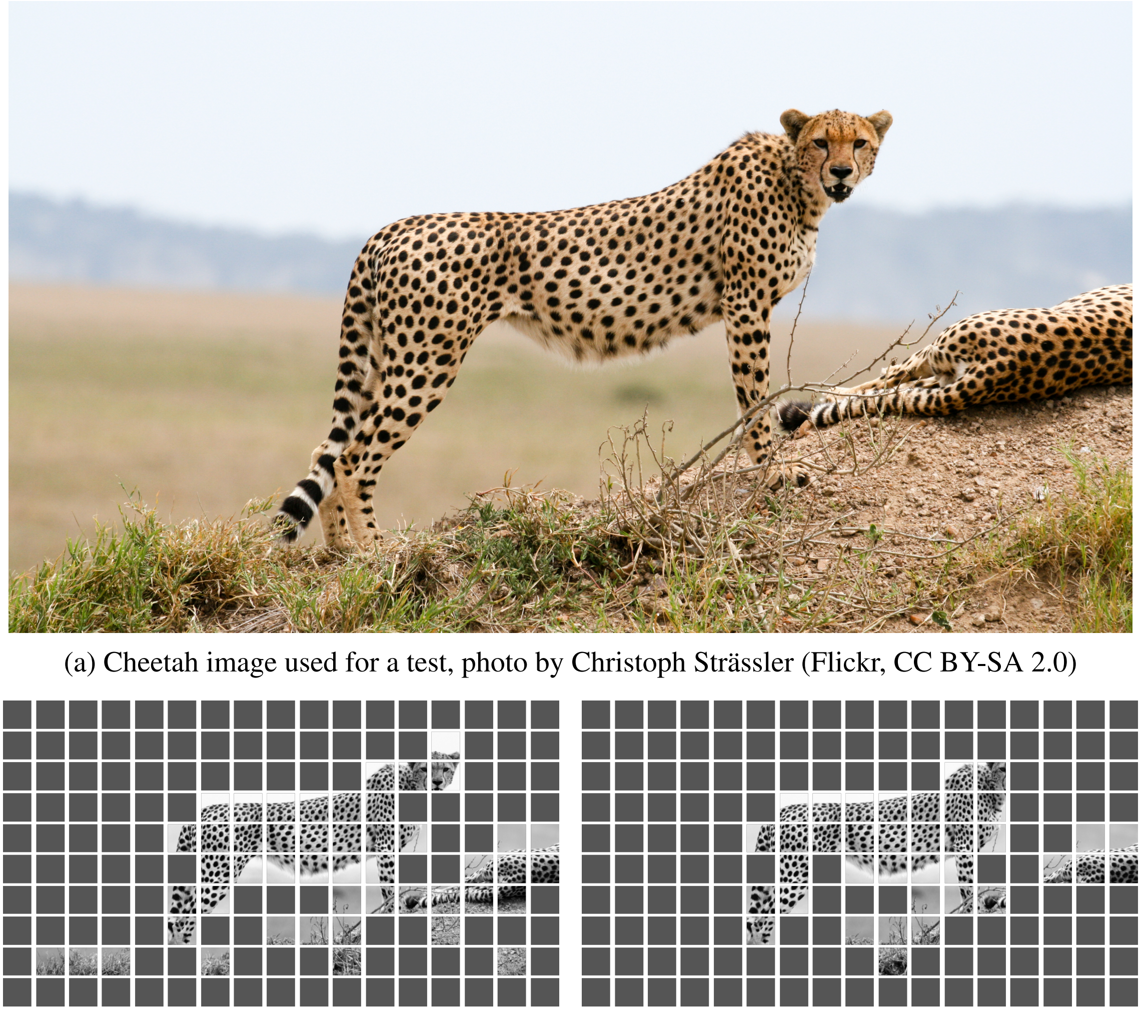
bounding box detection that identifies regions of interest, which could be used for wildlife monitoring or for automation of the cropping process. Nonetheless, this task may be improved by finding optimal descriptor setups and parameters for the sub-image grid.

## 4.6 Parameter estimation

The last validation method we employed for our descriptors is parameter estimation, similar to Sun et al. (2017). We used regression analysis with our descriptors as inputs for a Random Forest algorithm and estimated the $ratio$ and $scale$ parameters of the reaction-diffusion model presented in Malheiros, Fensterseifer and Walter (2020).

For these tests, we created a new dataset of reaction-diffusion images from spotted patterns to “pre-labyrinthine” patterns, with three different images for each $ratio$ and $scale$ pair, created using different seeds. This new dataset was created using simulations as explained in Section 4.2.1, using the isotropic kernel shown in Equation 4.4, ratio values ranging from 4.1 to 10.1, scale values ranging from 3.0 to 10.5. Each ratio-scale pair was simulated three times with three different seeds that define the pseudo-random starting condition of matrix $b$. Additionally, all simulations used 15,000 iterations with a

Figure 4.14: Example of a test where the best results use all default descriptors.



(a) Cheetah image used for a test, photo by Christoph Strässler (Flickr, CC BY-SA 2.0)

(b) Resulting clusters from (a) using the default descriptors without ANN

(c) Resulting clusters from (a) using all default descriptors

step value of 0.01.

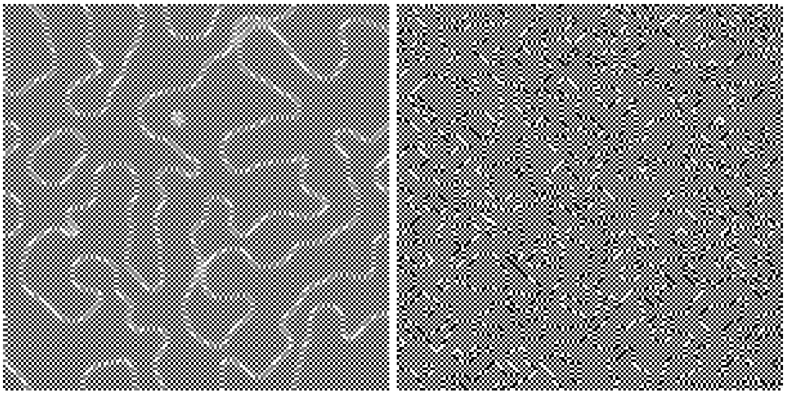
We totalize 11,649 images with 150x150 pixels resolution. Some images are results of non-convergent simulations, that is, parameter pairs that did not create a visual pattern, such as the resulting images shown in Figure 4.15. By analyzing the number of regions of the resulting images, we automatically eliminate the non-convergent ones, as these images do not have pattern regions. The remaining images were divided into 70% training images and 30% testing images.

To determine the efficiency of the model, we used three metrics: the first is a coefficient of determination, usually referred to as $R^2$, given by

$$R^2 = (1 - \frac{\sum(y_{true} - y_{pred})^2}{\sum(y_{true} - \overline{y}_{true})^2}) \tag{4.6}$$

with a maximum value of 100% (values on a 0 to 1 range). The second metric is the

Figure 4.15: Two examples of non-convergent resulting images.



(a) Ratio of 4.2 and scale of 9.5 (b) Ratio of 4.2 and scale of 15.0

Source: The author

mean score of a cross-validation scheme with five folds, calculating the mean squared error (MSE), where 0 represents no error. This measure is given by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y}_i)^2 \tag{4.7}$$

where $n$ is the number of values, $y_i$ are the target values, and $\overline{y}_i$ are the predicted values. The last metric used to evaluate our descriptors is the *feature importance* (Gini importance), which presents a percentage of importance for each feature of a given set.

The model trained with our default descriptor setup as features obtained a $R^2$ score of 99.29$\%$ and cross-validation mean MSE score of 0.019 (the scores for each fold were 0.016, 0.020, 0.017, 0.014, and 0.027). As for the feature importance, Table 4.10 shows the importance scores obtained for each descriptor, sorted from most to least important.

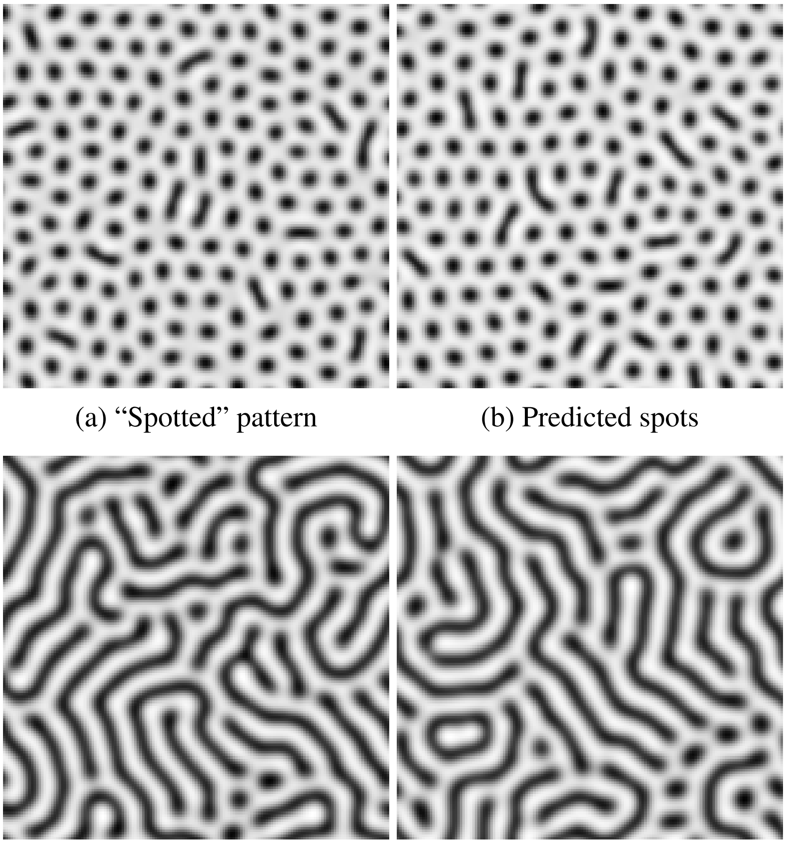
Table 4.10: Feature importance for the random forest regressor model sorted by the importance score.

| **descriptor** | **importance** |
|---|---|
| region radius STD | 21.42% |
| border perimeter ratio | 21.38% |
| region eccentricity mean | 21.18% |
| pattern area percentage | 17.42% |
| ANN | 06.04% |
| region border STD | 05.80% |
| region distance STD | 04.24% |
| border pattern ratio | 02.14% |
| pattern loss percentage | 00.15% |
| region eccentricity STD | 00.10% |
| standard distance | 00.09% |
| region area STD | 00.05% |

Similar to the importance given by the classifying task, the *border perimeter ratio*, *pattern area percentage*, and *region eccentricity mean* descriptors show great relevance. Nevertheless, the regressor feature importance presents the *region radius STD* descriptor with the largest importance score, in contrast to the no more than 5% obtained for the classifier importance, possibly due to the dataset being composed solely by synthetic reaction-diffusion images while the classifier dataset contains real patterns images. Considering this, we expect different importance scores and descriptor ranking if a better, more complete dataset was used.

Due to the limitations of the model used and the number of parameters, we limited our scope to simple reaction-diffusion spotted and labyrinthine patterns. As an example of limitations, we could not generate striped patterns, as we do not predict the diffusion kernel that generates these patterns, as explained in Section 2.1.

Figure 4.16: Two examples of target image given as input and output pattern generated using the regressor parameters.



(a) "Spotted" pattern

(b) Predicted spots

(c) Labyrinthine pattern

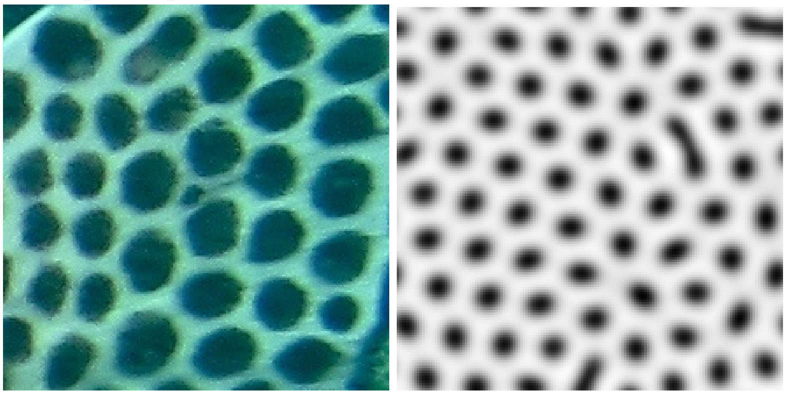
(d) Predicted labyrinth

Source: The author

Figure 4.16 shows two examples of target images and the pattern obtained using the parameters predicted by the regressor. For these tests, we calculate a MSE (Equation

4.7) using the descriptors computed for the target image and the ones obtained for the image created with the predicted parameters. From Figure 4.16a, generated with $ratio = 4.8$ and $scale = 3.1$, we obtained $ratio = 4.7$ and $scale = 3.17$, resulting on Figure 4.16b and a MSE of 0.00029. From the labyrinthine pattern of Figure 4.16c, computed with $ratio = 9.8$ and $scale = 3.1$, the regressor predicted $ratio = 9.41$ and $scale = 3.27$, resulting on Figure 4.16d and a MSE of 0.012.

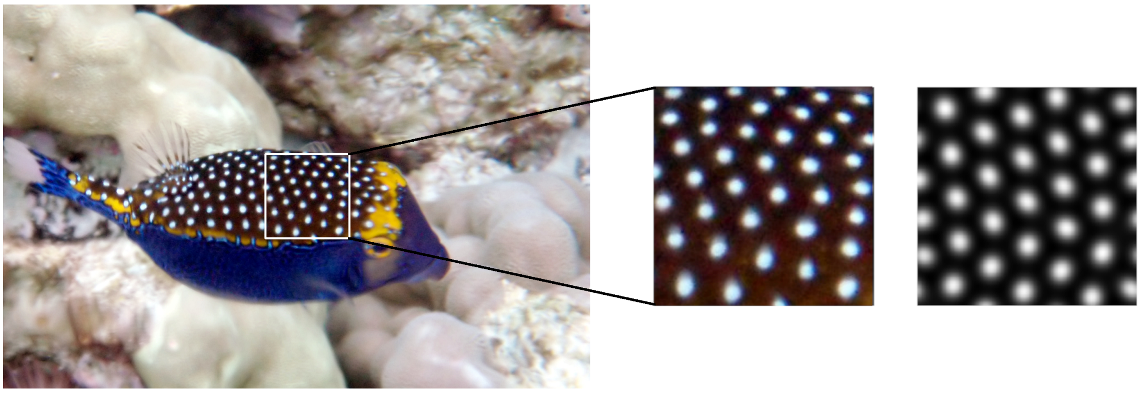
Figure 4.17: Example of a extrapolation of the model using a rabbitfish (Plectorhinchus chaetodonoides) crop as input.



(a) Rabbitfish pattern (b) Predicted pattern

Source: The author

Figure 4.18: Example of a extrapolation of the model using a simpler rabbitfish (Ostracion meleagris) crop as input. Rabbitfish photo by Brocken Inaglory (Wikimedia, CC BY-SA 4.0).



Source: The author

Our parameter estimation shows promising results for the limited scenario with simple synthetic patterns. When extrapolating it to natural patterns, our regressor model correctly predicted parameters for the given topology, as exemplified in Figure 4.17, from which we computed an MSE of 0.026, even though the pattern scale was wrong. Figure 4.17b was generated with the predicted values of $ratio = 4.92$ and $scale = 7.38$ using a rabbitfish (Plectorhinchus chaetodonoides) crop as the target image. When estimating for a simpler natural pattern, as shown in Figure 4.18, we obtained a resulting image similar to the rabbitfish (*Ostracion meleagris*) pattern used as target, from which a ratio of 4.22 and a scale of 7.51 were estimated.

For more complex patterns, cheetahs as an example, the RD model still seems not capable of producing the breadth of spot variations. We used a straightforward regressor with a limited dataset. Possibly more complex results could be achieved by an implementation that estimates more parameters, such as the diffusion kernel, using a more extensive and diverse dataset with multiple image and simulation resolutions, also allowing the idea of matrix growth. Another possible result of these improvements is a better approach for the *inverse rendering* problem, enabling the usage of non-synthetic images as input.

# 5 CONCLUSION

In this work, we justified the need for quantitative descriptors to evaluate biologic patterns comprised of spots, stripes, and labyrinths. We then proposed an automated pipeline to extract such descriptors, quantitatively finding the best set of descriptors and validating them in two different machine learning tasks, one supervised classification and one unsupervised clustering, a cluster-based segmentation application, and a parameter estimation model for a limited set of possible patterns.

Albeit we used a dataset with a few thousand images and a limited set of different species, we believe the diversity already present among the images provided enough challenge for a starting point. Besides, we achieved good results on our validation tasks presented in Chapter 4. When comparing to state-of-art deep learning pre-trained models we obtained better accuracy scores in a scenario containing downsampled images, 97.72% compared to best DL results, 94.63%. When removing these downsamples, although our results showed a significant accuracy drop (approximately 2.7%), we still obtained better results than the DL models (around 0.5% better). These results, though, are obtained in a limited and specific scenario, and as such, more tests are needed for better comparison.

## 5.1 Future Works

We worked with a limited set of patterns and species, and therefore we wish to continue expanding the range of species in future work, thus including other topologies, such as inverse spot patterns and irregular patterns.

We also plan to explore more precise image-to-image similarity metrics, such as analyzing and comparing histograms based on individual pattern detail characteristics. Another idea not explored is the use of different orders of the neighborhood for the ANN descriptor, that is, the average $n_{th}$ nearest neighbor, which presents the possibility of creating graphs of all neighborhood orders, where the curve may be used to examine the relative spatial arrangement of points, that is, their position relative to one another. More possible unexplored set of measures are color-based metrics, which could be computed for both the pattern and the “background”, and measures related to frequency domain. One last approach would be *metric learning*, either as means of obtaining new descriptors or as to compare obtained metrics with our descriptors.

For further validation, we also plan to use transfer learning on the Deep Learning models tested in Section 4.3. Some potential improvements are related to our parameter estimation task (Section 4.6), in which we could improve the dataset and model used, as well as estimating more parameters, such as the diffusion kernel, allowing exploring more pattern structures and topologies, and, possibly, natural patterns.

## REFERENCES

BALL, P. **Patterns in nature**. [S.l.]: University of Chicago Press, 2016.

BARD, J.; LAUDER, I. How well does turing's theory of morphogenesis work? **Journal of Theoretical Biology**, Elsevier, v. 45, n. 2, p. 501–531, 1974.

BATAINEH, B. et al. Adaptive thresholding methods for documents image binarization. In: SPRINGER. **Mexican Conference on Pattern Recognition**. [S.l.], 2011. p. 230–239.

CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1251–1258.

CHOLLET, F. et al. **Keras**. GitHub, 2015. Available from Internet: <https://github.com/fchollet/keras>.

CRALL, J. P. a. Hotspotter—patterned species instance recognition. In: IEEE. **IEEE WACV**. [S.l.], 2013. p. 230–237.

FOURNIER, A. The modelling of natural phenomena. In: **Graphics Interface '89**. [S.l.: s.n.], 1989. p. 191–202.

FOWLER, D. R.; MEINHARDT, H.; PRUSINKIEWICZ, P. Modeling seashells. In: **Proceedings of the 19th annual conference on Computer graphics and interactive techniques**. [S.l.: s.n.], 1992. p. 379–387.

GALIN, E. et al. A review of digital terrain modeling. **Computer Graphics Forum**, v. 38, n. 2, p. 553–577, 2019.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. [S.l.]: O'Reilly Media, 2019. ISBN 9781492032595.

GIMOND, M. **Intro to GIS and Spatial Analysis**. [S.l.]: <https://mgimond.github.io/Spatial/index.html>, 2022.

GLIMM, T. et al. Capturing and analyzing pattern diversity: an example using the melanistic spotted patterns of leopard geckos. **PeerJ**, PeerJ Inc., v. 9, p. e11829, 2021.

HE, K. et al. Identity mappings in deep residual networks. In: SPRINGER. **European conference on computer vision**. [S.l.], 2016. p. 630–645.

HO, T. K. Random decision forests. In: IEEE. **Proceedings of 3rd international conference on document analysis and recognition**. [S.l.], 1995. v. 1, p. 278–282.

HUANG, G. et al. Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 4700–4708.

IJIRI, T. et al. An example-based procedural system for element arrangement. In: **Computer Graphics Forum**. [S.l.: s.n.], 2008. v. 27, n. 2, p. 429–436.

JR, J. T. K.; RAJA, S.; BADLER, N. I. Fruit senescence and decay simulation. In: WILEY ONLINE LIBRARY. **Computer Graphics Forum**. [S.l.], 2011. v. 30, n. 2, p. 257–266.

KIM, H. et al. Pattern formation in reaction–diffusion systems on evolving surfaces. **Computers & Mathematics with Applications**, Elsevier, v. 80, n. 9, p. 2019–2028, 2020.

KONDO, S.; SHIROTA, H. Theoretical analysis of mechanisms that generate the pigmentation pattern of animals. **Seminars in cell & developmental biology**, v. 20 1, p. 82–9, 2009.

KONG, Q.; SIAUW, T.; BAYEN, A. **Python Programming and Numerical Methods: A Guide for Engineers and Scientists**. [S.l.]: Academic Press, 2020.

LAHIRI, M. et al. Biometric animal databases from field photographs: identification of individual zebra in the wild. In: **Proceedings of the 1st ACM multimedia retrieval**. [S.l.: s.n.], 2011. p. 1–8.

LINGE, S.; LANGTANGEN, H. P. **Programming for Computations-Python: A Gentle Introduction to Numerical Simulations with Python 3.6**. [S.l.]: Springer Nature, 2020.

MALHEIROS, M. de G.; FENSTERSEIFER, H.; WALTER, M. The leopard never changes its spots: realistic pigmentation pattern formation by coupling tissue growth with reaction-diffusion. **ACM TOG**, v. 39, n. 4, p. 63–1, 2020.

MCGUIRL, M. R.; VOLKENING, A.; SANDSTEDE, B. Topological data analysis of zebrafish patterns. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 117, n. 10, p. 5113–5124, 2020.

MEINHARDT, H. Models of biological pattern formation. **New York**, Citeseer, v. 118, 1982.

MIAO, Z. et al. Insights and approaches using deep learning to classify wildlife. **Scientific reports**, Nature Publishing Group, v. 9, n. 1, p. 1–9, 2019.

MIURA, T.; KOMORI, M.; SHIOTA, K. A novel method for analysis of the periodicity of chondrogenic patterns in limb bud cell culture: correlation of in vitro pattern formation with theoretical models. **Anatomy and Embryology**, Springer, v. 201, n. 5, p. 419–428, 2000.

MORDVINTSEV, A. et al. Differentiable programming of reaction-diffusion patterns. **arXiv preprint arXiv:2107.06862**, 2021.

MURRAY, J. **Mathematical biology II: spatial models and biomedical applications**. [S.l.]: Springer-Verlag, 2001.

NIBLACK, W. **An Introduction to Digital Image Processing (Englewood Cliffs, NJ**. [S.l.]: Prentice-Hall, 1986.

NIKLASSON, E. et al. Self-organising textures. **Distill**, v. 6, n. 2, p. e00027–003, 2021.

OTSU, N. A threshold selection method from gray level histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, p. 62–66, 1979.

PRUM, R. O.; WILLIAMSON, S. Reaction–diffusion models of within-feather pigmentation patterning. **Proceedings of the Royal Society of London. Series B: Biological Sciences**, The Royal Society, v. 269, n. 1493, p. 781–792, 2002.

RAVELA, S.; GAMBLE, L. On recognizing individual salamanders. In: ASIAN FEDERATION OF COMPUTER VISION SOCIETIES JEJU. **Proceedings of Asian Conference on Computer Vision, Ki-Sang Hong and Zhengyou Zhang, Ed. Jeju, Korea**. [S.l.], 2004. p. 742–747.

RINGHAM, L. et al. Modeling flower pigmentation patterns. **ACM Transactions on Graphics (TOG)**, ACM New York, NY, USA, v. 40, n. 6, p. 1–14, 2021.

ROGOWSKA, J. Overview and fundamentals of medical image segmentation. **Handbook of medical imaging, processing and analysis**, Academic, p. 69–85, 2000.

SADEGHI, I. et al. Physically-based simulation of rainbows. **ACM TOG**, Association for Computing Machinery, New York, NY, USA, v. 31, n. 1, feb. 2012. ISSN 0730-0301.

SAUVOLA, J.; PIETIKÄINEN, M. Adaptive document image binarization. **Pattern recognition**, Elsevier, v. 33, n. 2, p. 225–236, 2000.

SEZGIN, M.; SANKUR, B. Survey over image thresholding techniques and quantitative performance evaluation. **Journal of Electronic imaging**, International Society for Optics and Photonics, v. 13, n. 1, p. 146–166, 2004.

SIMONCELLI, E. P.; OLSHAUSEN, B. A. Natural image statistics and neural representation. **Annual review of neuroscience**, Annual Reviews 4139 El Camino Way, PO Box 10139, Palo Alto, CA 94303-0139, USA, v. 24, n. 1, p. 1193–1216, 2001.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SPRINGMEYER, B. Siggraph '87 panel summary: Natural phenomena. **SIGGRAPH Comput. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 22, n. 1, p. 16–20, feb. 1988. ISSN 0097-8930.

STODDARD, M. C.; OSORIO, D. Animal coloration patterns: Linking spatial vision to quantitative analysis. **The American Naturalist**, University of Chicago Press Chicago, IL, v. 193, n. 2, p. 164–186, 2019.

SUN, M. et al. Automated numerical simulation of biological pattern formation based on visual feedback simulation framework. **PloS one**, Public Library of Science San Francisco, CA USA, v. 12, n. 2, 2017.

SZEGEDY, C. et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Thirty-first AAAI conference on artificial intelligence**. [S.l.: s.n.], 2017.

TAN, M.; LE, Q. Efficientnetv2: Smaller models and faster training. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2021. p. 10096–10106.

TURING, A. M. The chemical basis of morphogenesis. **Bulletin of mathematical biology**, Springer, v. 52, n. 1-2, p. 153–197, 1952.

TURK, G. Generating textures on arbitrary surfaces using reaction-diffusion. **Siggraph Computer Graphics**, ACM New York, NY, USA, v. 25, n. 4, p. 289–298, 1991.

TURK, M. A.; PENTLAND, A. P. Face recognition using eigenfaces. In: **Proceedings. 1991 IEEE CVPR**. [S.l.: s.n.], 1991. p. 586–587.

VOLKENING, A.; SANDSTEDE, B. Iridophores as a source of robustness in zebrafish stripes and variability in danio patterns. **Nature communications**, Nature Publishing Group, v. 9, n. 1, p. 1–14, 2018.

WÄLDCHEN, J.; MÄDER, P. Machine learning for image based species identification. **Methods in Ecology and Evolution**, Wiley Online Library, v. 9, n. 11, p. 2216–2225, 2018.

WANG, G. et al. The shape space of 3d botanical tree models. **ACM Trans. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 37, n. 1, jan. 2018. ISSN 0730-0301.

WITKIN, A.; KASS, M. Reaction-diffusion textures. In: **Proceedings of SIGGRAPH 1991**. [S.l.: s.n.], 1991. p. 299–308.

ZOPH, B. et al. Learning transferable architectures for scalable image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 8697–8710.

# APPENDIX A — RESUMO EXPANDIDO

## Descritores quantitativos para uma variedade de padrões visuais de pigmentação biológica e sintética

Gabriel Henrique Moro, Marcelo Walter
Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

***Palavras-chave:*** Descritores quantitativos; processamento de imagens; geração de padrões; avaliação de padrões

### Introdução

A grande diversidade de padrões presentes na natureza atrai a atenção do campo de pesquisa em computação gráfica. Um dos desafios encontrados na área de modelagem de fenômenos naturais é a avaliação da qualidade dos modelos simulados em comparação com a contraparte natural. Diversos trabalhos apresentam avaliações qualitativas (empíricas), realizando comparações visuais dos seus resultados com os padrões naturais desejados, resultando em uma avaliação subjetiva e potencialmente imprecisa. O campo da biomatemática também precisa cada vez mais de melhores ferramentas quantitativas para a avaliação de padrões naturais. Portanto, percebe-se a necessidade de uma abordagem quantitativa. Neste trabalho é definido um conjunto de descritores adequados para capturar as principais características dos padrões analisados, isto é, padrões naturais e sintéticos regulares de manchas, listras e labirintos. Tais descritores facilitam a avaliação dos padrões avaliados, resultando em um melhor entendimento de suas estruturas e trazendo consistência ao campo. Para este trabalho o termo "descritor" é definido no mesmo contexto de técnicas de *content-based image retrieval* (CBIR), isto é, como medidas escalares que capturam alguma característica ou estrutura particular da imagem analisada.

### Metodologia

Foram propostos doze descritores divididos entre descritores globais e descritores locais. Descritores globais são calculados medindo razões de propriedades do padrão e expressam características gerais da estrutura do padrão, enquanto descritores locais medem características referentes as regiões individuais do padrão. Para o cálculo dos descritores, foi definido uma sequência de operações para o processamento das imagens. As três primeiras operações são realizadas para todos os descritores, a primeira sendo a conversão da imagem para escala de cinza, seguido por uma operação de limiarização para a segmentação do padrão, e então uma operação de remoção de ruído. Após estas operações são calculados dois descritores: *pattern area percentage*, que apresenta o percentual de área de padrão na imagem, e *border perimeter ratio*, a razão entre perímetro do padrão e perímetro da imagem. Os próximos dois descritores são calculados com uma operação adicional não executada para padrões de listras e labirintos: a remoção de regiões parciais, isto é, remoção de regiões que tocam as bordas da imagem. Estes descritores também são calculados para os padrões de listras e labirintos, apesar de não ser realizada a operação. Estes descritores são: *pattern loss percentage*, representando o percentual de perda de área do padrão após a remoção de regiões que tocam as bordas da imagem, e *border pattern ratio*, a razão entre o perímetro do padrão e a área do padrão. A última operação é a segmentação das regiões individuais e o cálculo de características destas regiões, realizada para o cálculo dos descritores locais. Os descritores locais podem ser divididos em duas categorias: baseados em pixel (ou em área) e baseados em centroides (ou em distâncias). A primeira categoria possui descritores calculados como desvios padrões de determinadas características das regiões do padrão, sendo estas a variação das áreas, dos perímetros, dos raios de círculos de área equivalente, e das excentricidade de elipses com segundo momento. Adicionalmente, também é calculado a média para a excentricidade. Os últimos descritores locais, baseados em centroides, são calculados como relações de distâncias dos centroides das regiões individuais segmentadas. O primeiro segue a lógica dos anteriores, sendo o desvio padrão das distâncias dos centroides após um precesso de triangularização de Delaunay. Os dois últimos são técnicas de análise de pontos, conhecidos como *Average Nearest Neighbor* (ANN), calculado como a média das distâncias mais próximas, e *Standard Distance*, calculado como a média do desvio padrão das regiões para um centroide médio.

**Resultados**

Os descritores foram validados com conjuntos de imagens de padrões reais e sintéticos, através de três métodos de aprendizado de máquina em quatro aplicações: um classificador supervisionado, um método de agrupamento não supervisionado (clusterizador), uma aplicação de identificação de regiões de interesse baseada na tarefa de clusterização, e uma tarefa de regressão para a estimação de parâmetros de um modelo matemático. A primeira aplicação utilizou os descritores em um classificador supervisionado e apresentou uma precisão média de 98.4%, também sendo medida o índice Gini para determinar a relevância dos descritores nos contextos testados. Por fim, foram feitas comparações com modelos pré-treinados de aprendizado profundo, observando-se que o modelo de classificação supervisionado, para o contexto deste trabalho, apresentou resultados superiores aos modelos pré-treinados de aprendizado profundo. A segunda aplicação buscou, de maneira não supervisionada, agrupar conjuntos de imagens a partir dos descritores calculados, conseguindo separar grupos de imagens reais e sintéticas, divididas por topologia e, até certo ponto, diferenciar animais por espécie. A terceira aplicação utiliza o mesmo método presente na segunda aplicação, porém classificando regiões de uma imagem de animais entre regiões de interesse e plano de fundo. Nesta aplicação foram obtidos resultados variados, vendo-se uma necessidade na realização de melhoras na implementação e realização de mais testes. O último método de validação foi uma aplicação de regressão com o objetivo de, a partir dos descritores calculados com uma imagem de entrada, obter-se os parâmetros de um modelo matemático de reação-difusão que gerem uma imagem similar. Foram observados bons resultados quando utilizados padrões sintéticos simples, porém, devido a limitações da implementação, não foi possível obter resultados satisfatórios para padrões mais complexos.

**Conclusão**

Em conclusão, foram obtidos bons resultados dentro das limitações apresentadas, com resultados de classificação superiores a modelos pré-treinados de aprendizado profundo no contexto definido para o trabalho e resultados esperados para as demais aplicações apresentadas. Observam-se também possíveis trabalhos futuros, como a expansão da variedade de padrões estudados, podendo-se incluir padrões irregulares e padrões inversos. Outra possibilidade é a proposição de novos descritores, como o uso de diferentes ordens de vizinhança para o descritor ANN ou descritores baseados em cores ou em domínio de frequência. Uma última possibilidade referente aos descritores seria explorar técnicas de *metric learning* (aprendizado de métricas), seja para a obtenção de novos descritores ou para comparação com os já existentes. Por fim, há também a possibilidade de melhora nos métodos de validação, como aprimorar a comparação da aplicação de classificação com o uso de técnicas de *transfer learning* para a comparação com os modelos de aprendizado profundo, ou ainda um aprimoramento para o modelo de regressão com a resolução de certas limitações apresentadas.