Feasibility of Spatial Model Checking for Nevus Segmentation

Gina Belmonte
S. C. Fisica Sanitaria Nord,
Azienda Toscana Nord Ovest
Lucca, Italy

Giovanna Broccia, Vincenzo Ciancia, Diego Latella, Mieke Massink
Istituto di Scienza e Tecnologie dell'Informazione 'A. Faedo',
Consiglio Nazionale delle Ricerche
Pisa, Italy

***Abstract*—Recently developed spatial and spatio-temporal model checking techniques have a wide range of application domains, among which large scale distributed systems as well as signal and image analysis. In the latter domain, automatic and semi-automatic contouring in Medical Imaging has shown to be a very promising and versatile application that may facilitate the work of professionals in this domain, while supporting *explainability*, easy *replicability* and *exchange* of medical image analysis methods. In recent work, spatial model-checking has been applied to the 3D contouring of brain tumours and related oedema in magnetic resonance images of the brain. In the present paper we address the contouring of 2D images of nevi. One of the challenges of contouring nevi is that they show considerable inhomogeneity in shape, colour, texture and size. In addition these images often include also extraneous elements such as hairs, patches and rulers. To deal with this challenge we explore the use of a *texture similarity* operator in combination with spatial logic operators. We investigate the feasibility of our technique on images of a large public database. We compare the results with associated ground truth segmentation provided by domain experts; the results are very promising, both from the quality and from the performance point of view.**

***Index Terms*—Spatial Logics; Model Checking; Medical Imaging; Nevi;**

I. Introduction and Related Work

A nevus is a visible, usually small and benign, circumscribed lesion of the skin. Unfortunately, in some cases these are hard to distinguish from their malignant counterpart known as Melanocytic nevus. Melanoma is a very serious form of skin cancer. It may be lethal if the disease is not recognised in a very early stage. In Europe alone melanoma causes over 20,000 deaths each year [21]. One of the difficulties is that reliable early detection requires highly trained specialists but in many countries there is only a limited number of such specialists available. It is therefore no surprise that there is much interest in automated systems that can help recognising the disease reliably and at an early stage so that more lives could be saved and the number of unnecessary biopsies can be reduced [17].

Part of this work has been developed in the context of the Italian MIUR-PRIN 2017 project IT MaTTerS: Methods and Tools for Trustworthy Smart Systems" and was partially supported by POR FESR 2014-2020 project STINGRAY (SmarT station INtelliGent RAilwaY). The names of the authors of the present paper are listed in the front-page in alphabetical order. All co-authors have contributed equally to the work described in the present paper and to the development of the paper.

The most popular and well-performing automated techniques for the diagnosis of melanoma at the moment rely on deep learning [17]. In this paper we take a different approach based on recently developed *spatial* model checking techniques, in particular for the contouring or segmentation of nevi, which is one of the sub-tasks involved in the diagnosis of melanoma. In our previous work on (semi-) automatic contouring using spatial model checking techniques for the contouring of various kinds of brain tissues and brain tumours [2], [4]–[6] we have shown that such techniques can reach a segmentation quality that is competitive with state-of-the-art techniques, while supporting explainability, easy replicability and exchange of medical image analysis methods. The segmentation of nevi poses additional challenges because dermoscopic images of nevi tend to be very inhomogeneous in size, colour, contrast, location and kind of nevus/lesion and the presence of additional objects such as coloured patches, hairs, shadows and other optical effects. In the domain of image analysis, model checking exhibits good scalability, with respect to its classical applications in System Verification (for instance, operating in a few seconds on images containing about 9 millions of voxels). Images are explicitly described by their voxels, therefore the so-called "exponential blowup" of model checking is not an issue in this context (in contrast, *systems* are usually described via process languages, and the semantics of parallel processes may give rise to models that grow exponentially with their descriptions).

Spatial (and spatio-temporal) model checkers use high-level specifications written in a logical language to describe spatial properties in order to automatically and efficiently identify spatial patterns and structures of interest. The origins of spatial logic can be traced back to the forties of the previous century when McKinsey and Tarski recognised the possibility of reasoning on space using topology as a mathematical framework for the interpretation of modal logic (see [1] for a thorough introduction). In [13], [14] the Spatial Logic for Closure Spaces (SLCS), and related efficient model checking algorithms, have been proposed that use *closure spaces* [22]–[24], [34], a generalisation of topological spaces, as the underlying model. In [12] a spatio-temporal logic, combining Computation Tree Logic with the spatial operators of SLCS was introduced. Spatial and spatio-temporal model checking have recently been applied in a variety of domains,

ranging from Collective Adaptive Systems [11], [15], [16] to signals [33] and medical images [2]–[6].

Several proposals of use of computational methods for the analysis of medical images are available in the literature. *Computer-Aided Diagnosis* (CAD) aims at the classification of areas in images, based on the presence of signs of specific diseases [19]. *Image Segmentation* [25] is focused on the identification of areas that have specific features or perform specific functions. *Automatic contouring* of Organs at Risk or target volumes [7] is specifically devoted to supporting radiotherapy applications. Finally, specific *indicators* can be computed from the acquired images that can enable early diagnosis—or the understanding of microscopic characteristics of specific diseases—or can help in the identification of prognostic factors to predict a treatment output [10], [41]. In [26] spiral electric waves—a precursor to atrial and ventricular fibrillation—are detected and specified using a spatial logic and model-checking tools. The formulas of the logic are learned from the spatial patterns under investigation and the onset of spiral waves is detected using bounded model checking. In [28] a logic called Spatial-Temporal Logic (SpaTeL) is defined that is a unification of signal temporal logic (STL) and tree spatial superposition logic (TSSL). The logic can be used for describing high-level spatial patterns that change over time.

In our previous work on the use of spatial model-checking for the analysis of medical images mentioned earlier, we focused on image segmentation, in particular for the identification of glioblastomas—which are the most common malignant intracranial tumours—but also of regions of interest in healthy organs [5].

In this paper we investigate the feasibility of the application of a technique based on SLCS for the analysis of images of nevi from a public database. One of the challenges of contouring nevi is that they show considerable inhomogeneity in shape, colour, texture and size. In addition these images often include also extraneous elements such as hairs, patches and rulers.

We show that, despite the challenges mentioned above, such images can be analysed in a semi-automatic way, by taking profit of the intrinsic rigour of a logic-based approach, using an efficient implementation of spatial model-checking algorithms. We compare the results with associated ground truth segmentation provided by domain experts; the results are very promising, both from the quality and from the performance point of view.

The public dataset we used was released by the International Skin Imaging Collaboration (ISIC) for the 2016[1] International Symposium on Biomedical Imaging (ISBI 2016) challenge titled "Skin Lesion Analysis toward Melanoma Detection" [17]. This dataset contains 900 annotated dermoscopic images, obtained by specialised high-resolution imaging of the skin that reduces skin surface reflectance. Among this set are 173 images of melanomas. Each image in the dataset has been segmented manually by experts and their segmentation result is available as ground truth images, which makes comparison with results of other state-of-the-art segmentation techniques applied to the same dataset possible. The original challenge consisted of three parts: Lesion Segmentation, lesion Dermoscopic Feature Extraction, and Lesion Classification. In the present work we focus on lesion segmentation.

The outline of the paper is as follows. Section II provides some background on spatial model checking, the spatial model checker `VoxLogicA` and in particular its texture similarity operator. Section III presents the spatial logic specification for the segmentation of nevi and Section IV presents the model checking results on the ISIC 2016 training and test datasets. Finally, Section V presents the conclusions of this work.

II. Background on Spatial Model Checking

ImgQL (*Image Query Language*), first proposed in [2], [6], is a spatial-logic language developed for the analysis of medical images. It is based on SLCS (*Spatial Logic for Closure Spaces*) [13], [14]. ImgQL is also the input language for the spatial model checker `VoxLogicA` presented in [6]. In this section we first recall the definition of the logic kernel of ImgQL and the underlying basic notions and then we show its extension supported by the tool. We refer to our earlier work for further details on theoretical aspects and the spatial model checking algorithms [2], [6], [13], [14].

A. The logical kernel of ImgQL

SLCS is interpreted over *closure spaces*. A closure space—CS for short—is a pair $(X, \mathcal{C})$ where X is a set (of points) and $\mathcal{C} : 2^X \to 2^X$ is a function satisfying the following three axioms: (i) $\mathcal{C}(\emptyset) = \emptyset$; (ii) $Y \subseteq \mathcal{C}(Y)$ for all $Y \subseteq X$; (iii) $\mathcal{C}(Y_1 \cup Y_2) = \mathcal{C}(Y_1) \cup \mathcal{C}(Y_2)$ for all $Y_1, Y_2 \subseteq X$. The *interior* of a set $Y \subseteq X$ is obtained by duality, i.e. $\mathcal{I}(Y) = \overline{\mathcal{C}(\overline{Y})}$ where $\overline{Y} = X \setminus Y$ is the complement of Y. Given any relation $R \subseteq X \times X$, $(X, \mathcal{C}_R)$, with $\mathcal{C}_R(Y) = Y \cup \{x \mid \exists y \in Y . y\, R\, x\}$, is a CS. In particular, a digital image can be modeled as a finite CS where X is the set of voxels and R their (reflexive and symmetric) *adjacency relation*[2].

$(\mathbb{N}, \mathcal{C}_{succ})$ is the CS of the natural numbers $\mathbb{N}$ with the binary successor relation $succ = \{(m, n) \in \mathbb{N}^2 \mid n = m + 1\}$. A (discrete) *path* over $(X, \mathcal{C})$ is a continuous function[3] from $(\mathbb{N}, \mathcal{C}_{succ})$ to $(X, \mathcal{C})$.

It is often convenient to equip the elements of X with *attributes* in a given set A over a given set of values V; an *attributed* CS is a structure $((X, \mathcal{C}), \mathcal{A})$ where $(X, \mathcal{C})$ is a CS and $\mathcal{A} : A \times X \to V$, is the attribute evaluation function, such that $\mathcal{A}(a, x)$ maps attribute (named) a of point x to its value in V. For instance, if x is a voxel, then $\mathcal{A}(red, x)$ may represent the intensity of red of x, and similarly for $\mathcal{A}(green, x)$ and

[1] We currently focus on the 2016 challenge data, which is a well-established dataset with good ground truth, and a published leaderboard. In future work we will test our approach against others datasets and the upcoming published results of subsequent challenges.

[2] All the theory and related model checkers work both for 2D and 3D even though we use only 2D in the current work. Similarly, in the current work we use the word 'voxel' both for 3D 'pixels' and for 2D pixels.

[3] A *continuous* function from CS $(X_1, \mathcal{C}_1)$ to CS $(X_2, \mathcal{C}_2)$ is a function $f : X_1 \to X_2$ such that $f(\mathcal{C}_1(Y)) \subseteq \mathcal{C}_2(f(Y))$ for all $Y \subseteq X_1$.

$\mathcal{A}(blue, x)$. Attribute values can be used in expressions α over V; consequently function $\mathcal{A}$ is assumed lifted to such expressions in the standard way.

In this paper we will use *distance* CS, i.e. structures $((X, \mathcal{C}), d)$ where $d : X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ is a *distance function*[4], i.e. it satisfies $d(x, y) = 0$ if and only if $x = y$; d is lifted to sets in the usual way: $d(x, \emptyset) = \infty$ and for $\emptyset \subset Y \subseteq X$ $d(x, Y) = \inf\{d(x, y) \mid y \in Y\}$.

ImgQL is interpreted over *attributed distance closure models*, i.e. structures $((X, \mathcal{C}), d, \mathcal{A}, \mathcal{V})$ where $(X, \mathcal{C})$ is a CS, d and $\mathcal{A}$ are the distance and the attribute evaluation functions, respectively, and $\mathcal{V} : P \to 2^X$ is a valuation which maps the *atomic predicates* of a given set P to the points satisfying them. In the sequel we recall the formal definition of the logical kernel of ImgQL:

Def. 1: For given set P of *atomic predicates* p, and interval I of $\mathbb{R}$, the syntax of ImgQL is the following:

$$\Phi ::= p \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \overrightarrow{\rho}\ \Phi_1[\Phi_2] \mid \overleftarrow{\rho}\ \Phi_1[\Phi_2] \mid \mathcal{D}^I\Phi.$$

Defined predicates are elements p of P for which a *defining equation* $p := \alpha$ is given, where α is an expression.

Satisfaction $\mathcal{M}, x \models \Phi$ of a formula Φ at point $x \in X$ in model $\mathcal{M} = (((X, \mathcal{C}), d), \mathcal{A}, \mathcal{V})$ is defined recursively on the structure of formulas, where $[\![\Phi]\!]^{\mathcal{M}}$ is the set $\{x \in X \mid \mathcal{M}, x \models \Phi\}$ of points satisfying Φ in $\mathcal{M}$, π is a path in $(X, \mathcal{C})$ and ℓ, j are indexes in $(\mathbb{N}, \mathcal{C}_{succ})$:

$$\begin{array}{lcll}
\mathcal{M}, x & \models & p \in P & \Leftrightarrow \quad x \in \mathcal{V}(p) \\
\mathcal{M}, x & \models & \neg\Phi & \Leftrightarrow \quad \mathcal{M}, x \models \Phi \text{ does not hold} \\
\mathcal{M}, x & \models & \Phi_1 \wedge \Phi_2 & \Leftrightarrow \quad \mathcal{M}, x \models \Phi_1 \text{ and } \mathcal{M}, x \models \Phi_2 \\
\mathcal{M}, x & \models & \overrightarrow{\rho}\ \Phi_1[\Phi_2] & \Leftrightarrow \quad \text{there are } \pi, \ell \text{ s.t. } \pi(0) = x, \\
 & & & \quad \mathcal{M}, \pi(\ell) \models \Phi_1, \text{ and } \pi(j) \models \Phi_2 \\
 & & & \quad \text{and for all } j \text{ s.t. } 0 < j < \ell \\
\mathcal{M}, x & \models & \overleftarrow{\rho}\ \Phi_1[\Phi_2] & \Leftrightarrow \quad \text{there are } \pi, \ell \text{ s.t. } \pi(\ell) = x, \\
 & & & \quad \mathcal{M}, \pi(0) \models \Phi_1, \text{ and } \pi(j) \models \Phi_2 \\
 & & & \quad \text{and for all } j \text{ s.t. } 0 < j < \ell \\
\mathcal{M}, x & \models & \mathcal{D}^I\,\Phi & \Leftrightarrow \quad d(x, [\![\Phi]\!]^{\mathcal{M}}) \in I
\end{array}$$

Whenever p is a *defined* predicate with defining equation $p := \alpha$, we extend the satisfaction relation by letting $x \in \mathcal{V}(p)$ if and only if $\mathcal{A}(x, \alpha)$ is true. •

Classical derived operators are defined as usual: $\bot \equiv p \wedge \neg p$, $\top \equiv \neg\bot$, $\Phi_1 \vee \Phi_2 \equiv \neg(\neg\Phi_1 \wedge \neg\Phi_2)$ etc. In addition, we have the following more specific derived operators:

$$\begin{array}{lcl}
\mathcal{N}\Phi & \equiv & \overleftarrow{\rho}\ \Phi[\bot] \\
\Phi_1\,\mathcal{S}\,\Phi_2 & \equiv & \Phi_1 \wedge \neg\ \overrightarrow{\rho}\ \neg(\Phi_1 \vee \Phi_2)[\neg\Phi_2] \\
touch(\Phi_1, \Phi_2) & \equiv & \Phi_1 \wedge\ \overrightarrow{\rho}\ \Phi_2[\Phi_1] \\
grow(\Phi_1, \Phi_2) & \equiv & \Phi_1 \vee touch(\Phi_2, \Phi_1) \\
smoothen(r, \Phi_1) & \equiv & \mathcal{D}^{<r}(\mathcal{D}^{\geq r}\neg\Phi_1).
\end{array}$$

Intuitively, a point x satisfies $\mathcal{N}\Phi$ if it is *near* Φ, i.e. if it can be reached *in one step* from a point laying in $[\![\Phi]\!]$; it is easy to see that $\mathcal{M}, x \models \mathcal{N}\Phi$ if and only if $x \in \mathcal{C}([\![\Phi]\!]^{\mathcal{M}})$. A point x satisfies $\Phi_1\,\mathcal{S}\,\Phi_2$ if it lays in an area, where all points satisfy Φ_1, that is *surrounded* by points satisfying Φ_2, i.e. it is impossible to find a path starting from x that can reach a point satisfying neither Φ_1 nor Φ_2, without first passing through a point satisfying Φ_2[5]. The meaning of $touch(\Phi_1, \Phi_2)$ should be clear. A point satisfies $grow(\Phi_1, \Phi_2)$ if it satisfies Φ_1 or it lays in a path of points all satisfying Φ_2 and leading to a point satisfying Φ_1. A formula $smoothen(r, \Phi_1)$ is satisfied by points that are at a distance of less than r from a point that is at least at distance r from points that do not satisfy Φ_1. This operator works as a filter; only contiguous areas satisfying Φ_1 that have a minimal diameter of at least $2r$ are preserved; these are also smoothened if they have an irregular shape (e.g. protrusions with a width that is less than the indicated distance).

We close this section with the description of an additional logical operator of ImgQL, namely the *Texture Analysis* operator $\triangle\!\!\!\!\triangle$ introduced in [6]. Texture Analysis (TA) is an approach used for finding patterns in (medical) images. The approach has proved promising in a large number of applications in the field of medical imaging [8], [18], [31], [32]; in particular it has been used in *Computer Aided Diagnosis* [29], [30], [42] and for classification or segmentation of tissues or organs [9], [35], [37]. The ImgQL TA operator $\triangle\!\!\!\!\triangle$ uses *first order* statistical methods[6] and differs from those in the classical setting, e.g. [39], [40], where the various moments (*mean*, *variance* etc.) of distributions of the two pictures to be compared are analysed. In ImgQL, instead, the statistical distributions—actually the histograms, as we will see below—of the two pictures are compared directly, using, as similarity measure, their *cross-correlation* (also called *Pearson's correlation coefficient*).

The intuitive semantics of $\triangle\!\!\!\!\triangle$ is presented schematically in Figure 1. Let $\mathcal{M}$ represent the image of the figure and suppose we want to study the similarity of a *relevant area*, laying around a point x of $\mathcal{M}$, and a *sample area of interest*—also in $\mathcal{M}$—specified by a ImgQL formula Φ. In addition, suppose the feature that makes the points around x relevant is represented by the (numeric) values of a certain attribute, say a, and, similarly, the interesting feature of the points in $[\![\Phi]\!]^{\mathcal{M}}$ is coded in attribute b, that can—but not necessarily need to—be the same as a. Let us focus on the sample area of interest; a common representation of the distribution of the values in $[\![\Phi]\!]^{\mathcal{M}}$ is the *histogram* H_Φ of $[\![\Phi]\!]^{\mathcal{M}}$, with respect to the attribute of interest b. In such a histogram, the range of values of the attribute is split into adjacent intervals of equal width—called *bins*—and for each bin, say j, $H_\Phi(j)$ is the total number of points that have a value of b falling in j. A point x is considered *similar* to the sample area of interest

[4]Several distance functions are defined in the literature; the specific distance to be used depends on the application. The interested reader is referred to [2]. In this work we use the Manhattan distance where 1 voxel is the unit distance.

[5]Note that in [2], [6], [13], [14] $\mathcal{N}$ and $\mathcal{S}$ (denoted by $\mathcal{U}$ in [13]) have been presented as basic operators while reachability operators have been defined as derived from the formers.

[6]First order statistical methods are statistics based on the probability distribution function of the intensity values of the pixels (of parts) of an image.

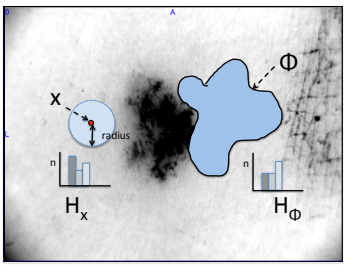

Fig. 1: Illustration of the ImgQL TA operator $\triangle\!\!\!\triangle$

$[\![\Phi]\!]^{\mathcal{M}}$ if the local histogram H_x of the relevant area around x, with respect to attribute a, correlates sufficiently with H_Φ. Of course, the values of the attributes a and b must be of the same type and the two histograms must have the same number of bins.

In the sequel we formalise the notion of histogram, with reference to a model $\mathcal{M} = ((X, \mathcal{C}), \mathcal{A}, \mathcal{V})$, and we recall the definition of the satisfaction relation for the $\triangle\!\!\!\triangle$ operator. Given set of points $Y \subseteq X$, attribute g, numeric values $m, M \in \mathbb{R}$, with $m < M$, and $k \in \mathbb{N}$ with $k > 0$, the histogram $\mathcal{H}^{m,M,k}_{Y,g}$ of the distribution of the numeric values of attribute g in Y, in the interval $[m, M]$, with k bins, and step size $\Delta = \frac{M-m}{k}$, is the function that, for each $j \in \{1, \ldots, k\}$, yields the number of points in Y with the value of attribute g laying in the interval associated with bin j, namely

$$\mathcal{H}^{m,M,k}_{Y,g}(j) = |\{y \in Y \mid (j-1) \cdot \Delta \leq \mathcal{A}(y, g) - m < j \cdot \Delta\}|.$$

The definition of $\triangle\!\!\!\triangle$ is based on the notion of histogram *cross-correlation*, which, in turn, uses the notion of *mean*. Let $h, h_1, h_2 : \{1, \ldots, k\} \to \mathbb{N}$ be histograms; the *mean* $\overline{h}$ of h is the value $\frac{1}{k}\sum_{j=1}^{k} h(j)$; the *cross-correlation coefficient* $\mathbf{r}(h_1, h_2)$ of h_1 and h_2 is defined as follows:

$$\mathbf{r}(h_1, h_2) = \frac{\sum_{j=1}^{k} \left(h_1(j) - \overline{h_1}\right)\left(h_2(j) - \overline{h_2}\right)}{\sqrt{\sum_{j=1}^{k} \left(h_1(j) - \overline{h_1}\right)^2}\sqrt{\sum_{j=1}^{k} \left(h_2(j) - \overline{h_2}\right)^2}}.$$

The coefficient is *normalised* so that $-1 \leq \mathbf{r}(h_1, h_2) \leq 1$; $\mathbf{r}(h_1, h_2) = 1$ indicates that h_1 and h_2 are *perfectly correlated* (that is, $h_1 = \alpha h_2 + \beta$, with $\alpha > 0$); $\mathbf{r}(h_1, h_2) = -1$ indicates *perfect anti-correlation* (that is, $h_1 = \alpha h_2 + \beta$, with $\alpha < 0$). On the other hand, $\mathbf{r}(h_1, h_2) = 0$ indicates no correlation[7].

In order to characterise the relevant area around a point x to be compared with the 'area of interest', the notion of *sphere* of radius r centred in x, $S(x, r)$ is used, that is defined in the usual way $S(x, r) = \{y \in X \mid d(x, y) \leq r\}$.

We now have all the ingredients for completing the definition of the logical kernel of ImgQL. We extend the syntax given in Def. 1 with $\triangle\!\!\!\triangle_{\bowtie c}\left[\begin{smallmatrix} m & M & k \\ r & a & b \end{smallmatrix}\right]$, where m, M, k are as above, a and b are attribute names, $c \in [-1, 1]$ and $\bowtie \in \{=, <, >, \leq, \geq\}$. The definition of the satisfaction relation is extended correspondingly, with the following clause, where $h_a(j) = \mathcal{H}^{m,M,k}_{S(x,r),a}(j)$, $h_b(j) = \mathcal{H}^{m,M,k}_{[\![\Phi]\!]^{\mathcal{M}},b}(j)(j)$:

$$\mathcal{M}, x \models \triangle\!\!\!\triangle_{\bowtie c}\left[\begin{smallmatrix} m & M & k \\ r & a & b \end{smallmatrix}\right]\Phi \Leftrightarrow \mathbf{r}(h_a, h_b) \bowtie c.$$

So $\triangle\!\!\!\triangle_{\bowtie c}\left[\begin{smallmatrix} m & M & k \\ r & a & b \end{smallmatrix}\right]\Phi$ compares the region of the space constituted by the sphere of radius r centred in x against the region characterised by Φ. The comparison is based on the cross correlation of the histograms of the two regions with respect to the chosen attributes, namely attribute a for the points around x and attribute b for the points that satisfy Φ. Both histograms share the same interval ($[m, M]$) and the same bins $(1, \ldots, k)$.

B. The VoxLogicA spatial model checker

VoxLogicA[8] is a spatial model-checker for ImgQL that is specialised for digital image analysis. It is a *global* spatial model-checker in the sense that, given a model $\mathcal{M}$ (i.e. a digital image) and a formula Φ, it computes the set $[\![\Phi]\!]^{\mathcal{M}}$ of all voxels in the image that satisfy Φ. Such a set can be, and usually is, represented by a boolean image—i.e. a closure model of the same dimension and size of $\mathcal{M}$, where each point is assigned the value *true* if the corresponding voxel in $\mathcal{M}$ satisfies Φ, and *false* otherwise. Actually, this feature is pushed forward in VoxLogicA so that one can obtain a resulting "grayscale" image – namely, an image where each point has a numerical value, that may denote, for instance, the cross-correlation score computed for the verification of a $\triangle\!\!\!\triangle$-formula on the corresponding voxel of $\mathcal{M}$. This is precisely what is done in the following example:

```
let scores = crossCorrelation(5,inty,inty,sample,min(inty),max(inty),15)
```

where sample is a formula characterising the sample portion of the image at hand and every point of scores will be associated with the score of the correlation between the intensity (inty) histogram of the sphere of radius 5 centred in the corresponding voxel of the image and intensity histogram of the sample area in the image, both histograms having 15 bins.

Functions and predicates can be defined in VoxLogicA in the usual way. For instance

```
let strongCorr(r,a,b,F,m,M,k,c) = crossCorrelation(r,a,b,F,m,M,k) > c
```

is the VoxLogicA equivalent of $\triangle\!\!\!\triangle_{>c}\left[\begin{smallmatrix} m & M & k \\ r & a & b \end{smallmatrix}\right]F$ so that

```
let interesting = strongCorr(5,inty,inty,sample,min(inty),max(inty),15,9.8)
```

returns in interesting a boolean image where the value *true* is associated to each voxel corresponding to a point in the current image which is the centre of a sphere of radius 5 the intensity of which has a high—higher than 9.8—correlation with the sample portion of the current image, and *false* to any other point.

[7]Note that normalisation makes the value of $\mathbf{r}$ undefined for constant histograms, having therefore standard deviation of 0; in terms of statistics, a variable with such standard deviation is only (perfectly) correlated to itself. This special case is handled by letting $\mathbf{r}(h_1, h_2) = 1$ when both histograms are constant, and $\mathbf{r}(h_1, h_2) = 0$ when only one of the h_1 or h_2 is constant.

[8]VoxLogicA is available at https://github.com/vincenzoml/VoxLogicA.

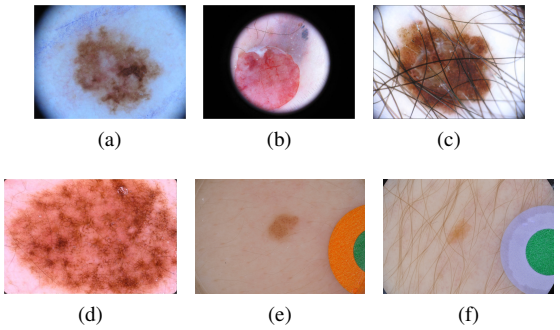

Fig. 2: Example images from the ISIC 2016 dataset illustrating the inhomogeneity of nevi. They also differ greatly in resolution, e.g. the size of (a) is 486 KB and that of (f) 11,3 MB (compared in .png format).

The following additional commands are available in VoxLogicA (more details can be found in [6]):

- load x = "s" loads an image from file "s" and binds it to x for subsequent usage;
- save "s" e stores the image resulting from evaluation of expression e to file "s";
- print "s" e prints to the log the string s followed by the numeric, or boolean, result of computing e;
- import "s" imports a library of declarations from file "s";

Algorithmic complexity: The asymptotic algorithmic complexity of the implementation of ImgQL primitives in VoxLogicA is *linear* in the number of voxels, with the exception of crossCorrelation, which has complexity $O(r \cdot n)$, where n is the number of voxels, and r is the radius of the 'area of interest' around voxel x. For further details we refer to [6]. Note that in spatial model-checking there is no risk of exponential state space explosion, because there is no notion of *behaviour* in the models, therefore there are no concerns related to the interleaving execution semantics of parallel components.

III. Segmentation of Nevi with VoxLogicA

As mentioned briefly in the introduction, a major issue of the segmentation of nevi is their great variability in appearance and the inhomogeneity of the dermoscopic images themselves. Nevi may show very different colour ranges, also within the same nevus, have different sizes, can be more or less regular, appear on more or less regular skin where hairs or sebaceous follicles may be present as well. Furthermore, the images themselves also show quite a variety and may be of different size, showing black corners, rings, or shadows due to the lenses used, showing more or less contrast and intensity or the presence of patches near the nevus. The images in Fig. 2 show a few examples of this inhomogeneity as encountered in the 2016 ISIC dataset[9].

[9]See: https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery

A. Datasets and Methodology

A large set of dermoscopic images is available from the ISIC gallery[9]. We will refer to the ISIC gallery dataset as *ISIC Gallery set*. This dataset includes two sets of images that were made available for the ISIC 2016 Challenge[10]. One set consisting of 900 images for training purposes and a test set consisting of 379 images. From here onward we will refer to these datasets as *training set* and *test set*, respectively. For the latter set, results on the quality of the segmentation procedures that participated to the 2016 Challenge are available in the literature [17], [27]. All sets provide both the original image and an associated "ground truth" segmentation. The ground truth segmentation has been performed by expert dermatologists.

Given the wide variability in dermoscopic images of nevi, as described earlier, it is not easy to select a representative subset of images to start from for the development of an ImgQL specification for their segmentation. As we shall see, the core focus of our approach is to distinguish skin tissue from nevus tissue with the help of the texture analysis and other spatial operators. Consequently, in a *first* phase we considered only images with single compact nevi and a fair amount of healthy skin around it, as well as a reasonably good contrast between skin and nevus. In particular, we started the development of the specification by considering suitable images from the first 10 (and then, in the *second* phase, the first 50) images of the ISIC *Gallery set*[11], assuming they would be sufficiently representative for a considerable sub-class of images in the *training* and *test* sets. In the sequel, we will refer to such a dataset as the *first10* set and *first50* set, respectively.

The results we obtained were very promising and so, in the *third* phase, we continued analysing further images from the *Gallery set*, having additional features, e.g. the presence of coloured patches and the presence of hair, improving the specification. The obtained extended specification, unmodified, generalised nicely to the much larger datasets used in the final phase, where we applied the resulting specification to both the *training* and *test sets*, in order to get further insight in the quality and performance of the analysis using our specification. The current specification does not yet cover all possible features of the images. In particular, images showing nevi consisting of several disconnected parts, nevi covering a very large part of the image or images with very low contrast are not giving optimal results. We leave those for future study.

B. Nevus Segmentation using Texture Analysis

Since there is very little one can take for granted in the dermoscopic images in the ISIC datasets, we start from a very coarse heuristics to initialise the segmentation procedure. In the following we illustrate the core steps of this segmentation procedure. In particular, we describe how the statistical texture

[10]These datasets can be found at https://challenge.isic-archive.com/data

[11]Specifically, images named ISIC 000000 to ISIC 0000050, excluding images 4, 11, 24, 26, 31, 33 and 50. Images 4, 26 and 33 overlap with the border, images 31 and 50 have multiple nevi; for image 11, due to a technical issue, we had the wrong ground truth, namely that of image 00; image 24 has very low contrast.

analysis operator plays a predominant role in approximating a nevus. The main aim is to distinguish voxels that are part of the background (skin) from those that are likely part of the nevus. First we assume that our task is to find all voxels that are likely to be part of the background, so the healthy skin surrounding the nevus. We assume furthermore that at least part of the nevus is somewhere in the middle of the image so that we can take an area relatively close to the border *as a sample of the background*. Let Φ be the ImgQL formula that specifies such an area, shown in Fig 3b as a semi-transparent overlay in cyan on the original image—later in the paper we will show Φ in detail. Note that, in this phase of the analysis, we work with the intensity of the voxels rather than their colour or other attributes.

At this point, statistical texture analysis is used. As described in Section II, the histogram H_Φ of the distribution of the intensity values of all the voxels that satisfy Φ is constructed. Assume that H_Φ has k bins, and a minimum and maximum value that correspond to the minimum and maximum pixel intensity in the whole image.

The local histogram H_x is computed for each pixel x in the image by taking the intensity of all the pixels that are present in a radius rad around pixel x. This second histogram has the same number of bins and minimal and maximal values as those of histogram H_Φ. The Pearson's correlation coefficient of the histograms H_Φ and H_x provides normalised values between -1 and +1. A value equal to 1 indicates perfect correlation between the histograms, a value equal to -1 indicates perfect anti-correlation. A score of value 0 indicates that there is no correlation between the histograms. The result for Fig. 3a is shown in Fig. 3c as a semi-transparent yellow overlay where higher values of the score correspond to a brighter yellow hue. The associated histogram of the cross-correlation scores shown in Fig. 3c are shown in Fig. 3f. Finally, in Fig. 3d, those pixels with a cross-correlation score above 0.05 are shown as an overlay in pink.

Thus, this particular use of the texture operator provides a rather good first approximation of the area covered by the nevus. Clearly, it is not perfect yet, as also some other areas remain that are not identified as part of the background, whereas they should be. But these areas can in principle be identified by other means, such as their relative position with respect to the border of the image and other aspects that distinguish them from the nevus itself. This is done in Specification 1, shown in Section III-C1. Specification 1 uses a predicate, patch and a derived operator relDist. The former is a predicate specifying voxels that are part of a patch. The latter is a derived operator that defines the *relative distance* in an image, depending on its size. The definition of both the operators are provided and explained after Specification 1.

Moreover, in Section IV we will use common similarity indexes to assess the quality of the segmentation. These indexes are defined directly in ImgQL and shown in Specification 5. They provide numeric support in the form of values of several commonly used similarity indexes that allow for an objective comparison with expert ground truth.

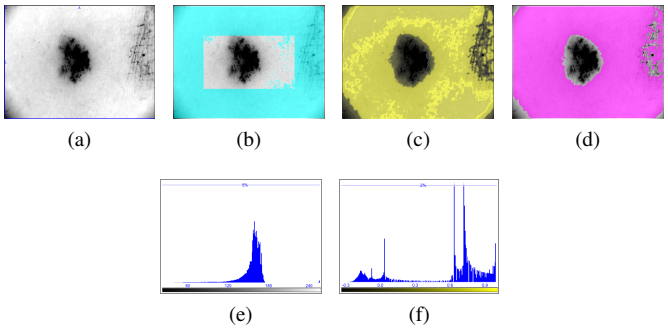

Fig. 3: The texture analysis operator of VoxLogicA. Fig. 3a Original image of nevus (image ISIC-0000010 of the *ISIC Gallery* set) and related histogram of voxel intensity in Fig. 3e. Sample Φ of background voxels shown in cyan (Fig. 3b). Cross-correlation score (in yellow) for surrounding of each voxel w.r.t. histogram of background sample Φ. The higher the score the higher the intensity of the yellow colour of voxels (Fig. 3c). The distribution of these cross-correlation scores is shown in Fig. 3f. Voxels with a cross-correlation score of more than 0.05 are shown in pink in Fig. 3d.

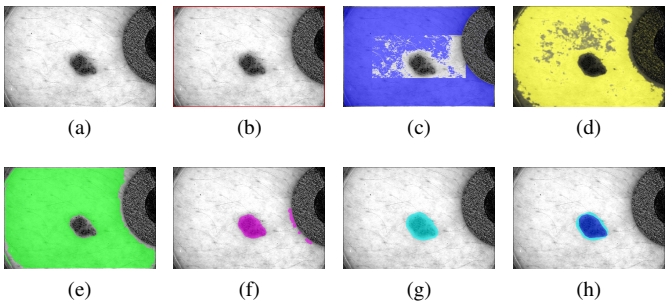

Fig. 4: Illustration of segmentation procedure in Specification 1 of image ISIC_0008294 (also shown in Fig 2 (e)). Figure (a) shows the nevus intensities (greyscale); Figures (b) to (h) are each associated with a specific formula, as indicated below, with the exception of (d) where the score is shown as a varying intensity of yellow: (b) blackBorder (red); (c) bgSample (blue); (d) bgSimScore (yellow); (e) bgSim (green); (f) preSeg (magenta); (g) nevSegV0 (cyan); (h) nevSegV0 (cyan) and ground truth (blue).

Note that the VoxLogicA procedure provided in the sequel does not require any particular preprocessing of the images as provided by the ISIC 2016 dataset, except for format conversion for the test data (from jpg format to png format) and colour conversion for the ground truth data (from grayscale to RGB). So we do not use any pre-processing transformations that, for example, remove hairs or black corners or borders of the image or that increase contrast or normalise the size of the images.

ImgQL Specification 1: Nevus segmentation specification

```
1  import "stdlib.imgql"
2  // Part a: Get images and intensities
3  load groundTruth = "$INPUTDIR/$NAME_seg_RGB.png"
4  load nevus = "$INPUTDIR/$NAME.png"
5  let nevusImgIntens = intensity(nevus)
6  let groundIntens = intensity(groundTruth)
7  // Part b: Define auxiliary operators
8  let similarTo(a,rad) =
     crossCorrelation(rad,nevusImgIntens,
     nevusImgIntens,a,min(nevusImgIntens),
     max(nevusImgIntens),15)
9  let imgPerc = percentiles(nevusImgIntens, !blackBorder,
     0.5)
10 let brightest(p) = imgPerc >. p
11 // Part c: Obtain sample of skin
12 let almostBlack = nevusImgIntens <. 40.0
13 let blackBorder = grow(distleq(relDist(5),border),
     almostBlack)
14 let bgSampleWidth = relDist(200)
15 let bgSample = (distleq(bgSampleWidth, blackBorder) &
     (!blackBorder) & (!patch)) | (brightest(0.7) &
     (!patchBright))
16 let onlySkinOrNevus = (! patch) & (! patchBright) & (!
     blackBorder)
17 let bgSimScore = similarTo(bgSample, relDist(5))
18 let bgSim = (bgSimScore >. 0.05) & onlySkinOrNevus
19 // Part d: Preliminary nevus segmentation
20 let preSeg1 = smoothen(((!border) S (bgSimScore <.
     0.25)) & onlySkinOrNevus & (! brightest(0.7)),
     relDist(10))
21 let preSeg2 = preSeg1 & !touch(preSeg1, border |
     blackBorder)
22 let preSeg = ifB(volume(touch(preSeg1, border |
     blackBorder) & !distleq(relDist(300), border))
     .=.0,preSeg2, preSeg1)
23 let nevSeg = smoothen(maxvol(preSeg), relDist(3))
24 let nevSegSmooth = smoothen(maxvol(preSeg), relDist(10))
25 let nevSeg1 = maxvol(nevSeg & nevSegSmooth)
26 // Part e: Extend pre-seg with similar nevus points
27 let nevSimScore = similarTo(nevSeg1, relDist(15))
28 let nevSim = nevSimScore >. 0.2
29 let nevSegV0 = grow(nevSeg1,
     distleq(relDist(10),nevSim)) & !patch & !patchBright
```

C. Explanation of the full procedure

1) Segmentation procedure: The nevus segmentation procedure consists of five parts as indicated in Specification 1. We describe the procedure in more detail below making reference to Specification 1.

Part a: After importing the standard library (stdlib.imgql), containing derived `VoxLogicA` operator definitions, and loading the image with ground truth and the related nevus image (lines 3-4), two abbreviations are introduced: nevusImgIntens for the nevus image and groundIntens for the ground truth image (lines 5-6). These (grayscale) images associate to each voxel its intensity (luminosity).

Part b: In line 8 a similarity operator is defined with parameters a and rad; a defines the sample area (denoted by Φ in Section II) and rad defines the radius around each voxel x for the construction of the local histogram of x. In line 9 a grayscale image is defined where the standard percentile of intensity is associated to each voxel. This makes it possible to reason about the brightest points in the image using the function defined in line 10.

Part c: The (intermediate) results of the segmentation procedure defined in the remaining part of the specification are illustrated in Fig. 4. Lines 12-13 specify the characteristics of voxels that are part of the black corners that can be observed in many images (in a similar way as shown in Fig. 2a and Fig 2b). Such voxels should not be considered in the sample of the skin texture. In line 12 voxels are specified that are almost black, i.e. having an intensity below 40. Then (line 13) only those almost black voxels are considered from which the border can be reached exclusively 'passing by' further almost black voxels, exploiting the grow operator.

In lines 14-15 a sample (bgSample) of the skin around the nevus is specified, namely a sample of voxels that are most likely part of the healthy skin without (or with very few) voxels that are part of the nevus. This sample consists of voxels that are at most at relative distance 200 (bgSampleWidth) from the black border (lines 12-13). In line 17 the similarity score of each voxel in the image w.r.t. the sample is computed using the similarTo operator and saved as a grayscale image. Line 18 characterises all voxels that have a cross correlation score larger than 0.05. This line has been inserted only for illustration purposes here to highlight the voxels with skin texture; in Figure 4e we show this set of points (bgSim).

Part d: A preliminary segmentation is specified in line 20, where we look for voxels that are not part of the border and that are surrounded by voxels with a cross correlation score of less than 0.25, a relatively small correlation score. The idea is that, at the border of the nevus and the healthy skin, the histograms of the area around those voxels represent in part the skin and in part the nevus, which have in general rather different intensity distributions. The cross correlation of such histograms with the sample area of the skin can therefore be expected to be quite small. The exact value of the threshold has been established in an empirical way; it is the value that gives *on average* good results for the subsets of the *training* dataset used in the third phase (see Section III-A). For optimal results on individual images this threshold value may differ slightly.

Of course, this pre-segmentation should exclude areas close to the black border and in patches. The latter are used in some images to indicate the position of nevus with little contrast (see for example Fig. 2e and Fig. 2f).

There may also be other small darker areas on the skin that are not part of the nevus. Therefore, in line 23, the preliminary segmentation is refined by taking only the largest volume (maxvol) smoothening the specified area removing small noise and irregularities at the edge. Occasionally, there may be darker areas that are not nevi but rather darker shadows in the area of the borders. This is due to the way the nevus are illuminated. These areas are removed in line 21 and 22.

In line 24 the same procedure of line 23 is repeated with a larger smoothening factor. This is used to exclude possible thin protrusions attached to the segmented nevus that are originating from thin hairs or shadows. In line 25 the intersection of these intermediate results is taken to preserve the more detailed edge of the nevus and at the same time to exclude some larger protrusions (i.e. several hairs grouped together). nevSeg1 provides a first nevus segmentation.

Part e: In the last part (lines 27 - 29) of the segmentation

procedure we extend the segmentation obtained so far with points that are sufficiently similar to the nevus by growing the area with these points, or nearby points, that are not part of a patch.

ImgQL Specification 2: Generating model checking results and similarity scores

```
1 let manualSeg = groundIntens >. 0
2 save "$OUTPUTDIR/$NAME_nevSegV0.png" nevSegV0
3 save "$OUTPUTDIR/nevSegV0.nii.gz" nevSegV0
4 print "DICE V0" dice(nevSegV0,manualSeg)
```

2) Comparing segmentation with ground truth: In Specification 2 the manual segmentation performed by domain experts (the 'ground truth') is defined as a predicate that is satisfied by voxels in the image of the ground truth where the intensity of the voxel is positive (line 1). In fact, manualSeg is a black and white image of the same size as the image of the nevus where the area indicated by the expert is white (intensity 255) and the rest black. The resulting segmentation (but also other intermediate results as those shown in Fig 4) can be saved in .png format or in the NIfTI (.nii) format. The latter format is used by various viewers used in medical imaging. We used the free viewer MRIcron[12]. The operator dice compares (line 4) the segmentation defined by nevSegV0 with the groundTruth giving as result a similarity score as defined in Specification 5. Further details on these scores are provided in the next section.

3) Scaling distance: Specification 1 uses the relDist operator defined in Specification 3. The ISIC 2016 datasets contain images of very different sizes. The relDist operator is introduced to scale the distance appropriately, with respect to a reference image. The size of the reference image is defined as the length of its perimeter, i.e. the number of voxels on its border. The reference image has a width of 1022 voxels and a height of 767 voxels. The perimeter of the image being analysed can be found as the volume (i.e. number of voxels) that form the border (i.e. one voxel wide edge) of the image. The property border is a built-in operator of ImgQL. The scaling of the distance is the fraction between the length of the perimeter of the image under analysis and that of the reference image.

4) Dealing with patches: Specification 1 also uses the predicate patch that is satisfied by voxels that are part of a patch. Patches are defined in Specification 4. Lines 1-3 define three quantitive images (matrixes) projecting the intensity of the blue, red and green part of the rgb-vector for each voxel of the image. Lines 6-10 define blue, red and green patches, respectively. These also cover intermediate hues such as yellow and orange. However, it is not enough to define the colour ranges of patches because nevi or skin may have occasionally colours in those ranges too (see for example Fig. 2a and Fig. 2b). Using further knowledge about the relative spatial position of patches (they are at the border of the image), their relative size (covering not more than 40 percent of an image) and their compactness (their Polsby-Popper measure of compactness of a shape[13], ppM, is at least 0.5), the specification patch is given in line 14. ifB is the boolean if-then-else construct of VoxLogicA. The definition of ppM is shown in Specification 5 (lines 6-8) in the next section. Occasionally also white or very bright patches are used. These are defined in much the same way, but starting from their brightness rather than the specific colour (lines 16-18).

ImgQL Specification 3: Relative distances

```
1 let refImgPerimeter = 2 .*. (1022 .+. 767)
2 let imgSizeFactor = (volume(border) ./. refImgPerimeter)
3 let relDist(x) = (imgSizeFactor .*. x)
```

ImgQL Specification 4: Patches

```
1  let bNev = blue(nevus)
2  let rNev = red(nevus)
3  let gNev = green(nevus)
4  //— blue, red and green patches —
5  let patchBlue = distleq(relDist(5),(bNev > (rNev +. 30))
    &
6  (bNev > (gNev)) & (bNev >. 150))
7  let patchRed = distleq(relDist(5),(rNev > (bNev +. 100))
    &
8  (rNev > (gNev +. 20))) & (rNev >. 130)
9  let patchGreen = distleq(relDist(5),(gNev > (rNev +.
    20)) &
10 (gNev > bNev) & (gNev >. 100))
11 let patchPart(x,y) = ifB(volume(x) .<. (y .*.
    volume(tt)),x,ff)
12 let patchSample = patchPart(patchBlue,0.4) |
    patchPart(patchRed,0.4) | patchPart(patchGreen,0.4)
13 let patchAtBorder =
    touch(smoothen(patchSample,relDist(10)),
    distleq(relDist(20),border))
14 let patch = ifB(ppM(patchAtBorder) .>. 0.5,
    patchAtBorder, ff)
15 //— bright patches —-
16 let patchYW = touch(smoothen(brightest(0.75),
    relDist(20)), distleq(relDist(20),border))
17 let patchSampleYW = patchPart(patchYW,0.4)
18 let patchBright = ifB((ppM(patchYW) .>. 0.3),
    patchSampleYW, ff)
```

IV. Results

In this section we assess the quality of the segmentation results for nevSegV0 of Specification 1. A standard way to do this is by using common similarity scores for comparing our segmentation with the manual segmentation (ground truth) by domain experts for each image of the ISIC datasets we used. We address two questions. The first question concerns the feasibility of the segmentation procedure proposed in the previous section. The second question is how our results are positioned with respect to the ISIC Challenge 2016. Before presenting the results, we recall the main similarity indexes in this field.

A. Similarity indexes

In the literature on medical imaging several indexes are used to compare similarity between two segmentations of the same image, in particular similarity between the manual and automatic segmentation. Commonly used similarity measures

[12]https://www.nitrc.org/projects/mricron

[13]Also known as "Isoperimetric quotient".

are the Dice index, the Jaccard index and the accuracy index. These coefficients give a result between 0 (no similarity) and 1 (perfect similarity). Further similarity measures are the sensitivity (fraction of true positives, i.e. fraction of pixels that the segmentation and the ground truth have in common) and specificity (fraction of true negatives, i.e. fraction of pixels that are not identified by the segmentation and are also not part of the ground truth). For example a Dice index of around 0.9 is considered as indicating very good similarity. The Dice index (D) and the Jaccard index (J) are related: $J = D/(2 - D)$. In general, the Dice index is the most relevant and mostly used in the literature; in the sequel, we will use the Dice index as the main similarity index. In Specification 5 these common similarity indexes are defined in ImgQL so that they can be calculated for each image segmentation during the analysis. Their definitions should be self-explanatory recalling that the operator `volume(x)` gives the number of voxels that satisfy property `x`. It must be noted though, that no unique 'gold standard' for comparison exists because also manual expert markings have a considerable level of variability. For instance, in [17] it was found that the average Jaccard index of agreement between 3 pairs of clinicians that each segmented a subset of 100 images was 0.786.

B. Feasibility of the VoxLogicA approach

The first question concerns the feasibility to obtain, using our approach, segmentations of nevi of sufficiently good quality, in other words, with a Dice similarity score of around 0.9 with respect to ground truth. To that purpose, we have run the `nevSegV0` specification on all the images of the ISIC 2016 *test* set; note that no image from that data-set has been used for the development of the specification. In Table I we show the similarity scores (Dice) we obtained for the images in the ISIC 2016 *test* set and for those in the *training* set. The table shows that for almost half of the images in the *test* set we obtain a Dice score of 0.9 or higher. For 70% of them we have a very good Dice score of more than 0.8.

So this shows that with our method we can, in principle, reach the required accuracy at least for a considerable number of individual images. Note also that the number of images for which we completely miss the nevus is very low; only four cases which amounts to only 1% of the total number. The fact that we do not reach a very high score for *all* images should not be a surprise, because, as we anticipated in Section III, the current version of the specification was not designed to be able to deal with all types of images. In particular our approach currently focuses on the feasibility of contouring single compact nevi with a fair amount of healthy skin around it and with a reasonably good contrast between skin and nevus. As shown in Table I, similar results have been obtained for the *training* set.

C. Further results on the training and test datasets

In Table II the *mean* values for the various indexes are shown for the segmentation with the `nevSegV0` specification for the ISIC 2016 *training* and *test* sets. The table also shows

	test set		training set	
nevSegV0	number	fraction (of 379)	number	fraction (of 900)
Dice > 0.9	175	0.46	398	0.44
Dice > 0.8	265	0.70	632	0.70
Dice > 0.7	307	0.81	742	0.82
Dice < 0.5	36	0.09	68	0.08
Dice $= 0$	4	0.01	6	0.01

TABLE I: Dice score distribution of the nevus segmentation method `nevSegV0` for the ISIC 2016 *test* and *training set*.

ImgQL Specification 5: Similarity indexes and the Polsby-Popper measure

```
let dice(x,y) = (2 .*. volume(x & y)) ./. (volume(x) .+.
 volume(y))
let jaccard(x,y) = dice(x,y) ./. (2 .-. dice(x,y))
let sensitivity(x,y) = volume(x & y) ./. (volume(x & y)
 .+. volume((!x) & (y)))
let specificity(x,y) = volume((!x) & (!y)) ./.
 (volume((!x) & (!y)) .+. volume((x) & (!y)))
let accuracy(x,y) = (volume(x & y) .+. volume((!x) &
 (!y))) ./. (volume(x & y) .+. volume((!x) & (!y)) .+.
 volume(x & (!y)) .+. volume((!x) & (y)))
let square(x) = x .*. x
let iboundary(x) = near(interior(x)) & !(interior(x))
let ppM(x) = (volume(x) .*. 4 .*. 3.14) ./.
 (square(volume(iboundary(x))))
```

the average scores, for the *test* set, of the best performing segmentation approach that participated in the ISIC 2016 Challenge [27].

As was expected, our results show lower average values than the best one, but that should not be a surprise as we know that the current version of our segmentation procedure is not designed to deal with all types of images. Nevertheless, it confirms that we reached promising results. In particular when considering the Jaccard scores of the results of other teams that participated to the 2016 Challenge, shown in the Challenge leaderboard[14]. The Jaccard scores for the 28 participating teams range from 0.468 to 0.843 on the Challenge 2016 *test* set. Our Jaccard score of 0.717 would position our results between rank 15 and 16 (out of 28). Some of the best performing segmentation approaches that participated to the 2016 Challenge on segmentation are based on machine learning approaches (see also [27], [36], [38]).

The values for sensitivity (SE) and specificity (SP) we obtained with our specification are also very interesting. As can be observed in Table II, for `nevSegV0` the specificity score (fraction of true negatives) is very good, and in line with that of the best performing team. The sensitivity score (fraction of true positives) is lower than the best score. This seems to indicate that when we find the nevus, we do not always manage to identify the *entire* nevus in an image. This is again not a big surprise since we focused on compact and connected nevi that are not covering a very large part of the image. Visual inspection of the images in the training set for which we obtain relatively low similarity scores confirms that these images indeed show these problematic features. Fig. 5

[14]https://challenge.isic-archive.com/leaderboards/2016

	Accuracy	Dice	Jaccard	SE	SP
V0: Mean *Training* set	0.902	0.818	0.726	0.810	0.965
V0: Mean *Test* set	0.899	0.809	0.717	0.802	0.960
2016 Best Mean *Test* set	0.953	0.91	0.843	0.91	0.965

TABLE II: Average similarity scores of nevSegV0 for images of the ISIC 2016 *training* set (all 900 images) and *test* set (all 379 images). 2016 Best Mean *Test* set gives the average scores of the best team that participated to the ISIC 2016 Challenge [27].

illustrates some examples of segmentation using nevSegV0.

D. Computational performance

As previously mentioned, the analysed images show very high variability in size (e.g. in *test* set the image size varies from a minimum of 389,156 pixel to a maximum of 12,212,224 pixel). Such variability has an effect on the duration of the analysis. As shown in Fig. 6 the correlation between the execution time and the image size is linear. This is also confirmed by the analysis on the average time nevSegV0 needs to analyse a single pixel (execution time over image size) which is 0.006 ms per pixel with a standard deviation almost equal to zero (0.001). More in general, for the analysis of the *test* set (379 images) nevSegV0 took in total less than 30 minutes with an average duration of 4 seconds per image.

For the analysis we have used VoxLogicA version 0.6.0_osx-x64 on an AMD Ryzen 7 2700 Eight-Core Processor with 32GB of memory. For the purpose of reproduction, the results for the datasets *first10*, *first50*, *training* and *test* are available in a git-repository[15]. The other datasets are available from ISIC.

E. Explainability, replicability and exchangeability

The illustration of the intermediate results of the segmentation procedure in Fig. 4 shows that with the spatial model-checking approach we have been using in this paper, each step

[15] https://github.com/brocciagi/Spatial-Model-Checking-for-Nevus-Segmentation

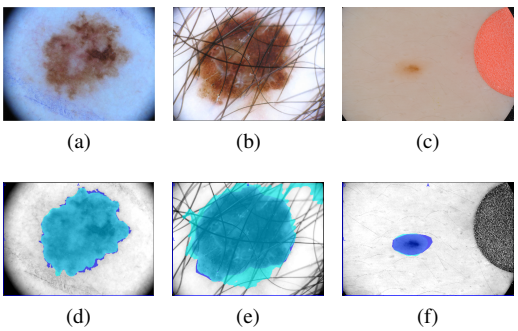

Fig. 5: Images and their segmentation (cyan) and ground truth (blue): ISIC_0000002 (5a) resp. (5d), ISIC_0000043 (5b) resp. (5e) and ISIC_0004309 (5c) resp. (5f).

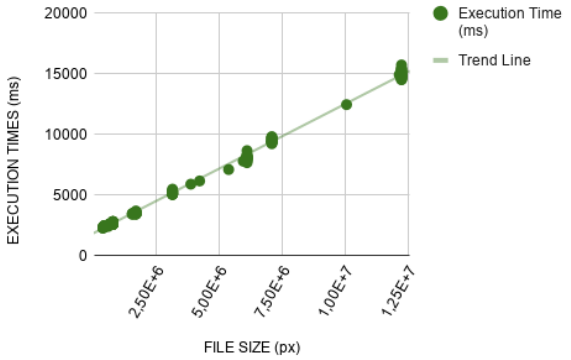

Fig. 6: Execution times (in milliseconds) as the image size (in pixels) varies needed for nevSegV0 to produce the final segmentation.

in the segmentation procedure is 'explainable', meaning that one can understand why a particular result is produced, based on an unambiguous formal spatial logic specification. This is one of the advantages of our approach. This approach also enables one to further improve the segmentation procedure, to discuss or *exchange* a particular part of the procedure with domain experts and to compare different segmentation procedures specified in ImgQL or use them to document a particular analysis of a patient's dermoscopic image.

Moreover, the resulting segmentations are amenable to more detailed analysis of the nevus area itself. For example one may extract further features such as a measure of the regularity of its shape or texture, size and so on that may be relevant to the diagnosis of nevi. This will be part of our future work. Note also that the segmentation method can be calibrated to specific individual images to obtain more precise results by tuning the thresholds. This would lead to a semi-automatic form of segmentation. Furthermore, this could also be a place where machine learning, semi-automatic procedures and our logic-based approach are bridged, as ML could be used to calibrate the numeric parameters, without affecting either explainability of the approach, or the possibility of a human re-calibrating them in order to fine-tune the results.

V. Conclusions and Future Work

We have shown how spatial model checking techniques and the related tool VoxLogicA can be used efficiently for the segmentation of nevi. Nevus segmentation based on dermoscopic images is an important part of many automatic procedures to diagnose malign skin tumours such as Melanoma. To the best of our knowledge, this is the first time that spatial model checking is applied to this specific domain. Spatial model-checkers use high-level, often domain oriented, logic languages to specify spatial properties. In this paper we have presented a segmentation method combining spatial operators inspired by the notion of closure spaces with more domain oriented operators such as a texture similarity operator. This first, rather simple method shows that an accuracy can

be obtained that is in line with the state-of-the-art in nevus segmentation, i.e. a Dice score above 0.9. It has also been shown that it obtains this high accuracy in 46% of 379 images of the ISIC 2016 Challenge *test* set of dermoscopic images that is publicly available. An advantage of this spatial model-checking method is that the segmentation procedure is explainable and high-level. This makes the method amenable to further improvements by inspection of the intermediate results, exchange and discussion of the method specifications between domain experts, conservation of the method for the purpose of documentation of the analysis and independent replication by other experts.

The results we obtained so far are very promising and future work is envisioned to increase the class of images for which accurate segmentation can be obtained in a similar spirit as we have shown how one can deal with the presence of patches or the presence of other artifacts in the images that are due to the way the images have been produced. The enormous inhomogeneity in this type of images, both for what concerns the nevi and the images themselves, remains a great challenge.

References

[1] Aiello, M., Pratt-Hartmann, I., Benthem, van, J.: Handbook of Spatial Logics. Springer (2007)

[2] Banci Buonamici, F., Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Spatial logics and model checking for medical imaging. Int. J. Softw. Tools Technol. Transf. 22(2), 195–217 (2020), https://doi.org/10.1007/s10009-019-00511-9

[3] Belmonte, G., Ciancia, V., Latella, D., Massink, M.: VoxLogicA: a Spatial Model Checker for Declarative Image Analysis (Extended Version). ArXiv e-prints (Nov 2018), https://arxiv.org/abs/1811.05677

[4] Belmonte, G., Ciancia, V., Latella, D., Massink, M., Biondi, M., De Otto, G., Nardone, V., Rubino, G., Vanzi, E., Banci Buonamici, F.: A topological method for automatic segmentation of glioblastoma in MR FLAIR for radiotherapy - ESMRMB 2017, 34th annual scientific meeting. Magnetic Resonance Materials in Physics, Biology and Medicine 30(S1), 437 (oct 2017), https://doi.org/10.1007/s10334-017-0634-z

[5] Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Innovating medical image analysis via spatial logics. In: ter Beek, M.H., Fantechi, A., Semini, L. (eds.) From Software Engineering to Formal Methods and Tools, and Back - Essays Dedicated to Stefania Gnesi on the Occasion of Her 65th Birthday. Lecture Notes in Computer Science, vol. 11865, pp. 85–109. Springer (2019), https://doi.org/10.1007/978-3-030-30985-5_7

[6] Belmonte, G., Ciancia, V., Latella, D., Massink, M.: Voxlogica: A spatial model checker for declarative image analysis. In: Vojnar, T., Zhang, L. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 25th International Conference, TACAS 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11427, pp. 281–298. Springer (2019), https://doi.org/10.1007/978-3-030-17462-0_16

[7] Brock, K.: Image processing in radiation therapy. CRC Press (2014)

[8] Castellano, G., Bonilha, L., Li, L., Cendes, F.: Texture analysis of medical images. Clinical Radiology 59(12), 1061–1069 (dec 2004)

[9] Chen, C., Da Ponte, J., Fox, M.: Fractal feature analysis and classification in medical imaging. IEEE Transactions on Medical Imaging 8(2), 133–142 (jun 1989)

[10] Chetelat, G., Baron, J.: Early diagnosis of alzheimer's disease: contribution of structural neuroimaging. NeuroImage 18(2), 525–541 (2003)

[11] Ciancia, V., Gilmore, S., Latella, D., Loreti, M., Massink, M.: Data verification for collective adaptive systems: Spatial model-checking of vehicle location data. In: Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW. pp. 32–37. IEEE Computer Society (2014)

[12] Ciancia, V., Grilletti, G., Latella, D., Loreti, M., Massink, M.: An experimental spatio-temporal model checker. In: Software Engineering and Formal Methods - SEFM 2015 Collocated Workshops. Lecture Notes in Computer Science, vol. 9509, pp. 297–311. Springer (2015)

[13] Ciancia, V., Latella, D., Loreti, M., Massink, M.: Specifying and verifying properties of space. In: Theoretical Computer Science - 8th IFIP TC 1/WG 2.2 International Conference, TCS 2014, Rome, Italy, September 1-3, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8705, pp. 222–235. Springer (2014)

[14] Ciancia, V., Latella, D., Loreti, M., Massink, M.: Model Checking Spatial Logics for Closure Spaces. Logical Methods in Computer Science Volume 12, Issue 4 (Oct 2016), http://lmcs.episciences.org/2067

[15] Ciancia, V., Latella, D., Massink, M., Paškauskas, R.: Exploring spatio-temporal properties of bike-sharing systems. In: 2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASO Workshops 2015, Cambridge, MA, USA, September 21-25, 2015. pp. 74–79. IEEE Computer Society (2015), https://doi.org/10.1109/SASOW.2015.17

[16] Ciancia, V., Latella, D., Massink, M., Paškauskas, R., Vandin, A.: A tool-chain for statistical spatio-temporal model checking of bike sharing systems. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9952, pp. 657–673 (2016), https://doi.org/10.1007/978-3-319-47166-2_46

[17] Codella, N.C.F., Nguyen, Q.B., Pankanti, S., Gutman, D., Helba, B., Halpern, A., Smith, J.R.: Deep learning ensembles for melanoma recognition in dermoscopy images. IBM Journal of Research and Development 61(45) (2017), http://www.research.ibm.com/journal/, special Issue on Deep Learning

[18] Davnall, F., Yip, C., Ljungqvist, G., Selmi, M., Ng, F., Sanghera, B., Ganeshan, B., Miles, K.A., Cook, G.J., Goh, V.: Assessment of tumor heterogeneity: an emerging imaging tool for clinical practice? Insights into Imaging 3(6), 573–589 (oct 2012)

[19] Doi, K.: Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. Comput. Med. Imaging Graph. 31(4-5), 198–211 (2007)

[20] Fitzpatrick, T.B.: The Validity and Practicality of Sun-Reactive Skin Types I Through VI. Archives of Dermatology 124(6), 869–871 (06 1988), https://doi.org/10.1001/archderm.1988.01670060015008

[21] Forsea, A., del Marmol, V., de Vries, E., Bailey, E., Geller, A.: Melanoma incidence and mortality in europe: new estimates, persistent disparities. British Journal of Dermatology 167(5), 1124–1130 (2012), https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2133.2012.11125.x

[22] Galton, A.: The mereotopology of discrete space. In: Freksa, C., David, M. (eds.) Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science, Lecture Notes in Computer Science, vol. 1661, pp. 251–266. Springer Berlin Heidelberg (1999), http://dx.doi.org/10.1007/3-540-48384-5_17

[23] Galton, A.: A generalized topological view of motion in discrete space. Theor. Comput. Sci. 305(1-3), 111–134 (2003), https://doi.org/10.1016/S0304-3975(02)00701-6

[24] Galton, A.: Discrete mereotopology. In: Calosi, C., Graziani, P. (eds.) Mereology and the Sciences: Parts and Wholes in the Contemporary Scientific Context, pp. 293–321. Springer International Publishing (2014), https://doi.org/10.1007/978-3-319-05356-1_11

[25] Gordillo, N., Montseny, E., Sobrevilla, E.: State of the art survey on MRI brain tumor segmentation. Magn. Reson. Imaging. 31(8), 1426–1438 (2013)

[26] Grosu, R., Smolka, S., Corradini, F., Wasilewska, A., Entcheva, E., Bartocci, E.: Learning and detecting emergent behavior in networks of cardiac myocytes. Commun. ACM 52(3), 97–105 (2009)

[27] Gutman, D., Codella, N.C.F., Celebi, M.E., Helba, B., Marchetti, M.A., Mishra, N.K., Halpern, A.: Skin lesion analysis toward melanoma detection: A challenge at the international symposium on biomedical imaging (ISBI) 2016, hosted by the international skin imaging collaboration (ISIC). CoRR abs/1605.01397 (2016), http://arxiv.org/abs/1605.01397

[28] Haghighi, I., Jones, A., Kong, Z., Bartocci, E., Grosu, R., Belta, C.: Spatel: A novel spatial-temporal logic and its applications to networked systems. In: Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control. pp. 189–198. HSCC '15, ACM, New York, NY, USA (2015)

[29] Han, F., Wang, H., Zhang, G., Han, H., Song, B., Li, L., Moore, W., Lu, H., Zhao, H., Liang, Z.: Texture feature analysis for computer-aided diagnosis on pulmonary nodules. Journal of Digital Imaging 28(1), 99–115 (aug 2014)
[30] Heinonen, T., Arola, T., Kalliokoski, A., Dastidar, P., Rossi, M., Soimakallio, S., Hyttinen, J., Eskola, H.: Computer aided diagnosis tool for the segmentation and texture analysis of medical images. In: IFMBE Proceedings, pp. 274–276. Springer Science (2009)
[31] Kassner, A., Thornhill, R.: Texture analysis: A review of neurologic MR imaging applications. Am. J. Neuroradiol. 31(5), 809–816 (2010)
[32] Lopes, R., Ayache, A., Makni, N., Puech, P., Villers, A., Mordon, S., Betrouni, N.: Prostate cancer characterization on MR images using fractal features. Med. Phys. 38(1), 83 (2011)
[33] Nenzi, L., Bortolussi, L., Ciancia, V., Loreti, M., Massink, M.: Qualitative and Quantitative Monitoring of Spatio-Temporal Properties with SSTL. Logical Methods in Computer Science 14(4), 1–38 (2018), DOI 10.23638/LMCS-14(4:2)2018. Published on line: 23 Oct. 2018. ISSN: 1860-5974
[34] Randell, D.A., Landini, G., Galton, A.: Discrete mereotopology for spatial reasoning in automated histological image analysis. IEEE Trans. Pattern Anal. Mach. Intell. 35(3), 568–581 (2013), https://doi.org/10.1109/TPAMI.2012.128
[35] Rodriguez Gutierrez, D., Awwad, A., Meijer, L., Manita, M., Jaspan, T., Dineen, R., Grundy, R., Auer, D.: Metrics and textural features of MRI diffusion to improve classification of pediatric posterior fossa tumors. American Journal of Neuroradiology 35(5), 1009–1015 (dec 2013)
[36] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR abs/1505.04597 (2015), http://arxiv.org/abs/1505.04597
[37] Sharma, N., Ray, A., Sharma, S., Shukla, K., Pradhan, S., Aggarwal, L.: Segmentation and classification of medical images using texture-primitive features: Application of BAM-type artificial neural network. J Med Phys 33(3), 119 (2008)
[38] Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence 39(04), 640–651 (apr 2017)
[39] Srinivasan, G., Shobha, G.: Statistical texture analysis. In: Proceedings of World Accademy of Science, Engineering and Technology. vol. 36, pp. 1264–1269 (dec 2012)
[40] Tijms, B., Series, P., Willshaw, D., Lawrie, S.: Similarity-based extraction of individual networks from gray matter MRI scans. Cerebral Cortex 22(7), 1530–1541 (aug 2011)
[41] Toosy, A.: Diffusion tensor imaging detects corticospinal tract involvement at multiple levels in amyotrophic lateral sclerosis. J. Neurol. Neurosurg. Psychiatry 74(9), 1250–1257 (2003)
[42] Woods, B., Clymer, B., Kurc, T., Heverhagen, J., Stevens, R., A., O., Bulan, O., Knopp, M.: Malignant-lesion segmentation using 4d co-occurrence texture analysis applied to dynamic contrast-enhanced magnetic resonance breast image data. J. Magn. Reson. Imaging 25(3), 495–501 (2007)