Image Databases with Features Augmented with Singular Point Shapes for Enhanced Machine Learning

Nikolay Metodiev Sirakov * and Adam Bowden

Posted Date: 11 July 2024

doi: 10.20944/preprints202407.0874.v1

Keywords: Image; Vector Field; Features; Augmentatiion; Classification; Machine Learning

Article

Image Databases with Features Augmented with Singular Points Shapes to Enhance Machine Learning

Nikolay Metodiev Sirakov *,† **and Adam Bowden** †

Department of Mathematics, Texas A & M University-Commerce, Commerce, Tx; adam.bowden@tamuc.edu

* Correspondence: nikolay.sirakov@tamuc.edu; Tel.: 903 450 6212

† These authors contributed equally to this work.

Abstract: The main objective of this paper is to present a repository of image databases whose features are augmented with embedded vector field (VF) features. The repository is designed to provide the user with image databases which enhance machine learning (ML) classification. Also, six VFs are provided and the user may embed them into her/his own image database with the help of software named ELPAC. Three of the VFs generate real shaped singular points (SPs): springing, sinking and saddle. The other three VFs generate seven kinds of SPs which include the real shaped SPs and four complex shaped SPs: repealing and attracting (out and in) spirals; clockwise and counterclockwise orbits (centers). Further, the present work defines the locations of the SPs according to the image objects and the mappings between the SPs' shapes if separate VFs are embedded into the same image. Thus, the paper produces recommendations for the most appropriate VF selection to embed into an image database so that the augmented SPs shapes enhance the ML classification. Examples of images with embedded VFs are shown in the text for illustration and to support and validate the theoretical conclusions. Thus, the contributions of the paper are the derivation of the following findings: SPs location in an image; the mappings between the SPs of different VFs; the definition of an imprint of an image an of an image database in a VF. The advantage of classifying an image database with embedded VFs is that the new database enhances ML and improves the classification statistics if compared to the classification of the original image database with the same ML classifier.

Keywords: image; vector field; features; augmentation; machine learning; classification

1. Introduction

The number, the sizes, and the variety of image databases has flourished in the recent decade. These databases are generated to solve problems in medicine, science, security, and industry. The demand for increasing accuracy is high and important. For example, face image databases usually contain hundreds of thousands of images. Hence, a single percentage of error will lead to thousands of incorrectly processed samples. Therefore, embedding additional features into an image carries the potential of enhancing the capabilities of the classifier [1] to correctly identify unknown input samples. This holds because in medical images, like skin lesions, the features are cluttered and chaotic. Hence, embedding image related, additional, features make the classifier more focused, which improves the accuracy [2,3].

VFs are classical tools in hydro-, air- and electrodynamics but in the recent decades they attracted the attention of Computer Vision and Image Analysis scientists. In [4] the authors developed an Ambrosio-Tortorelli scalar field for the purpose of objects partitioning. The authors of [5] developed a method that applies the gradient vector field (GVF) of the solution of the Poisson partial differential equation to extract geometric features and objects. Another application of VFs is in facilitating the iterative matching of features through "consensus" between VFs [6]. The advantage provided by this approach is the reduced number of false matches. It holds because the VF is generated by a bank of directional morphological openings which considers multiple directions of contour and defines ellipses as special structuring elements [7], defining new adaptive filters for image segmentation.

This paper describes a repository of image databases where the geometric image features are augmented with VF features like SPs shapes, trajectories of vectors, and separatrices. In paper [3] we

 doi:10.20944/preprints202407.0874.v1

proved that the SPs shapes are invariant according to scaling, translation, and weak rotations. This property shows that the SP shapes are useful features to augment the image features and enhance the statistics of ML classifiers. We validated this advantage with a sparse representation wavelet classification (SRWC) and a sparse representation classification quaternion wavelet (SRCQW) method presented in paper [1] and with five convolutional neural networks (CNNs) in the papers [2,3].

For the purpose of validation we applied the above listed ML classifiers on the public image databases ISIC2018 [19], ISIC2020 [20], COIL100 [16], and YaleB [18]. These databases contain 2D images of different types and sizes. The first two image databases consist of skin lesion images, the third one consists of 100 objects recorded from 72 different angles, and the last database contains the images of 38 faces recorded under 64 different illumination conditions. Note that the ISIC2020 is the largest among the above listed databases and has 32,542 benign and 584 malignant images. In addition, 10,982 non-labeled benign and malignant images are available for testing.

To facilitate the embedding, learning and classification processes the original images were resized to 420×420 for the COIL100 database and to 192×168 for the Yale database. Further, from ISIC2018 were generated two image databases resized to 250×250 and 500×500 pixels, respectively. Additionally, two datasets were generated from ISIC2020 by resizing the original images to 256×192 and 332×590 pixels, respectively.

To augment the image features with SPs shapes we have developed six VFs. In [1,11] we designed three VFs $\nabla\hat{u}$, $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ whose SPs have real Eigenvalues and shapes formed by vectors whose main directions are determined by the Eigenvectors of the SPs. Further, in [3] we developed two VFs $v_{\hat{u}}$ and $v_{\hat{\phi}}$ which contain SPs with real and complex Eigenvalues, Eigenvectors, and shapes. In the present study we formulate the sixth VF $v_{\hat{\psi}}$ which also possess SPs with real and complex shapes.

We proved in [1,11] that the gradient VF (GVF) $\nabla\hat{u}$ generates two kinds of SPs' shapes, namely saddle and sinking. The two GVFs $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ generate three real shaped SPs': saddle, sinking and springing. Next, we proved in [3] that the VFs $v_{\hat{u}}$ and $v_{\hat{\phi}}$ may create seven SPs shapes which include the three real shaped SPs as well as attracting and repealing (in and out) spirals along with clockwise and counterclockwise orbits.

We embed the six VFs $\nabla\hat{u}, \nabla\hat{\phi}, \nabla\hat{\psi}, v_{\hat{u}}, v_{\hat{\phi}}$ and $v_{\hat{\psi}}$ with the help of Matlab tools, incorporated into the ELPAC software [12], which was initially developed for automatic image segmentation with an active contour guided by the VF $\nabla\hat{u}$. By embedding a VF into every image of a database we built up a new image database, where the image features are augmented with SPs shapes. Thus, by applying ELPAC on the four image databases ISIC2018 [19], ISIC2020 [20], COIL100 [16] and YaleB [18] we generated a set of new image databases where the above six VFs are embedded. These image databases are ready to use and have the advantage of enhancing the classification statistics compared to the classification statistics of the original database [1–3]. The first repository of image databases with embedded VFs is available with downloads at the URL given below. The web page also contains data to support the claim that embedding VF increases accuracy of classification:
https://www.tamuc.edu/projects/augmented-image-repository/?redirect=none.

On the same web page, we link to the EPLAC software [12] which could be implemented for embedding into any image any of the six VFs described in this paper. An additional advantage of images with embedded VF is for image databases augmentation when there is a shortage of samples or imbalanced classes. However, implementation of this idea requires a deep and rigorous study and support by experimental results. Hence, it will be a subject of study in our future works. Further, the above web page provides the SRWC [1] code which can be used, to classify original and image databases with embedded VFs, for the purpose of comparison with the users' own classifiers.

The contributions of this study are: 1) Definition of the SPs locations according to the image objects; 2) Generalization of the mappings between the SPs shapes if the six VFs are separately embedded into the same image: 3) Definition of the new type of image and image database named "imprint of an image and imprint of an image database in a VF". These contributions provide the opportunity for the

 doi:10.20944/preprints202407.0874.v1

user to choose the VF to be embedded into an image database in accordance to the particular database in order to increase classification statistics.

The rest of this paper is organized as follows: Sections 2 is composed of two subsections. The first one describes the six VFs used to augment the image features with SPs shapes, while the second one describes the SPs shapes, the mappings between them, and the SPs locations in an image; Section 3 consists of two main subsections which describe the original, the newly generated image database with embedded VFs, and the software tools available for image features augmentation and image databases classification. The paper ends with a discussion on the contributions and the advantages obtainable by classifying image databases with embedded VFs and/or imprints of image databases in VFs.

2. Vector Fields That Augment the Image Features

2.1. Definition of Vector Fields

The set of VFs for augmenting image features with SPs shapes can be characterized with the help of weighted Laurent polynomials [13], which distinguish 16 kinds (shapes) of SPs. Another way to describe the VFs is through the Eigenvalues of their Jacobian [1,14,15]. This approach defines 7 different kinds (shapes) of SPs. In our studies we adopted the latter classification and distinguish 7 different SPs shapes which we embed into images. Three of the SPs shapes are: springing, sinking and saddle. They correspond to the real Eigenvalues of the Jacobian of the gradient VFs (GVFs). The shapes of the three kinds of SPs are described in [1,11], and are illustrated in Figure 1.

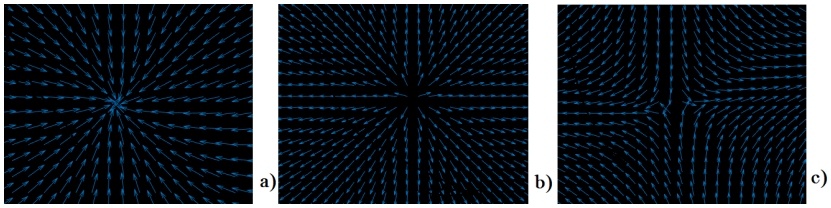


Figure 1. The SPs shapes are cropped from synthetic image where is embedded the VF: **(a)** $v_{\hat{u}}$ - sinking shaped SP; **(b)** $v_{\hat{u}}$ - springing shaped SP; **(C)** $\nabla\hat{\phi}$ - saddle shaped SP.

The remaining four SPs shapes are those corresponding to the complex Eigenvalues of the Jacobian of the VFs $v_{\hat{u}}$, $v_{\hat{\phi}}$ and $v_{\hat{\psi}}$. The SPs shapes are formed according to the directions of the vectors oriented toward the directions of the Eigenvectors and resemble attracting (in) and repealing (out) spirals, as well as clockwise and counterclockwise orbits [3]. The SPs with complex shapes are usually located in homogeneous regions. Examples of SPs with spiral out (repealing) and clockwise orbit shapes are presented in Figure 2.

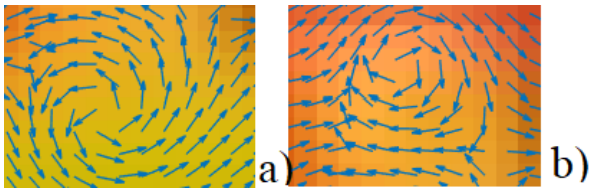


Figure 2. The SPs shapes are cropped from COIL100 image where is embedded the VF $v_{\hat{\phi}}$: **(a)** shows spiral out (repealing) shaped SP; **(b)** presents clockwise orbit SP.

To generate the seven kinds of SPs shapes described above, we have developed three GVFs with real Eigenvalues [1,11] and three non GVFs with real and complex Eigenvalues [3]. The six VFs are defined with the help of the solution $\hat{u}(x,y)$ of the Poisson Image equation [1,3,11], where $\partial\Omega$ denotes the image frame and $I(x,y)$ the image function:

$$\Delta u(x,y) = -\frac{|\nabla I(x,y)|^2}{1+I(x,y)}, u = I(x,y), (x,y) \in \partial\Omega. \quad (1)$$

 doi:10.20944/preprints202407.0874.v1

The developed GVFs are defined with the following equations [1,11]:

$$\nabla\hat{u} = u_x i + u_y j, \qquad \nabla\hat{\phi} = \hat{\phi}_x i + \hat{\phi}_y j, \qquad \nabla\hat{\psi} = \hat{\psi}_x i + \hat{\psi}_y j, \tag{2}$$

and they generate the sinking, springing and saddle shaped SPs. In Eqs. 2 the function

$$\hat{\phi} = \|\nabla\hat{u}\|^2, \qquad \hat{\psi} = \frac{-\Delta\hat{u}+1}{\|\nabla\hat{u}\|} = \frac{-\Delta\hat{u}+1}{\sqrt{\hat{\phi}}}. \tag{3}$$

The next three VFs are non-GVF and are developed by replacing the summations in Eqs. 2 with subtractions [3]:

$$v_{\hat{u}} = u_x i - u_y j, \quad v_{\hat{\phi}} = \hat{\phi}_x i - \hat{\phi}_y j, \quad v_{\hat{\psi}} = \hat{\psi}_x i - \hat{\psi}_y j. \tag{4}$$

The VFs defined with Eq. 4 generate the above three real shaped SPs and the following four complex shaped SPs, named attracting (in) and repealing (out) spirals, as well as clockwise and counterclockwise orbits (known also as centers) [14,15].

The ELPAC software was initially introduced in [12] with the capability of embedding $\nabla\hat{u}$ to evolve an active contour. The tools to embed $\nabla\hat{\phi}$, and $\nabla\hat{\psi}$ were incorporated to conduct the study in [1], while [3] extended ELPAC with tools to embed $v_{\hat{u}}, v_{\hat{\phi}}$ and most recently we added a tool to embed $v_{\hat{\psi}}$. Hence ELPAC embeds in an image any of the six VFs , as shown in Figures 4, 6, 7. Further, if we remove the objects from the image with embedded VF, we receive an image of the VF generated from the original image (Figure 4 parts (b), (d), (f), (h), (j), (l)). Thus, an image containing only the VF, generated on the original image, we call "imprint of the image in the VF". The imprints of the image from Figure 3, in the six VFs, are shown in Figure 4. Further, we define the notion "imprint of an image database in a VF" as the set of imprints of the images from the original database in the VF.

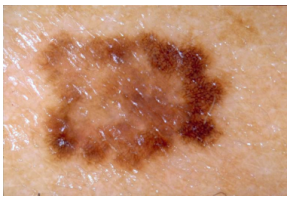


Figure 3. A malignant skin lesion image from [17].

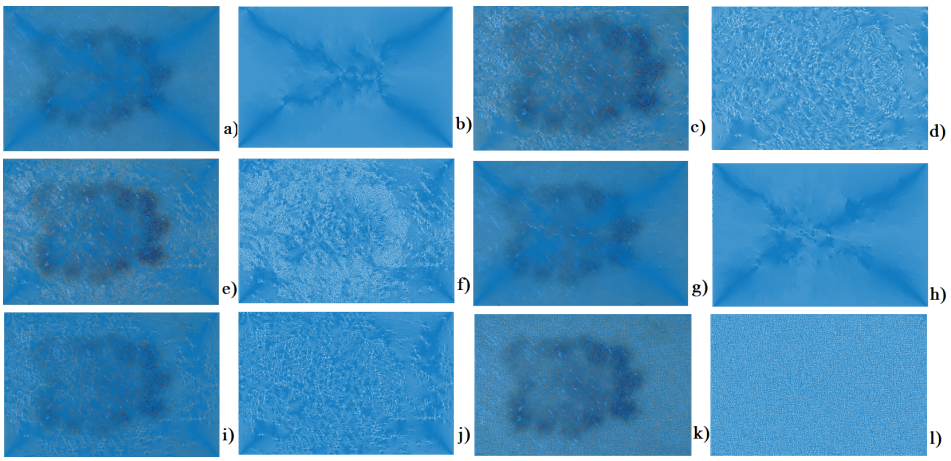


Figure 4. Parts **(a)**, **(c)**, **(e)**, **(g)**, **(i)**, and **(k)** show the six VFs $\nabla\hat{u}, \nabla\hat{\phi}, \nabla\hat{\psi}, v_{\hat{u}}, v_{\hat{\phi}}, v_{\hat{\psi}}$ embedded into the skin lesion image shown in Figure 3; The remaining parts **(b)**, **(d)**, **(f)**, **(h)**, **(j)**, and **(l)** show the imprints of the skin lesion in the six VFs respectively.

 doi:10.20944/preprints202407.0874.v1

2.2. Vector Fields SPs for Image Features Augmentation

In the present section we describe the SPs shapes created by each of the six VFs and list approximately where, according to the image objects, the SPs are located. Also, we determine the mappings between the SPs shapes if different VFs are separately embedded into an image. It helps define the similarities and differences between VFs components like SPs, separatrices [1] and architectures (skeletons [21] if embeded into the same image. This comparison facilitates the choice of the VF most appropriate for embedding into an image database to enhance ML and improve classification.

Recall that each of the VFs introduced, in subsection 2.1, is defined with the solution $\hat{u}(x, y)$ of Eq. 1, which we solve on every image $I(x, y)$ of an image database. This implies that, in every image, we can embed VF features like SPs shapes, edges of SPs shapes, trajectories of vectors, and separatrices [1]. The last entity is created by SPs shapes and trajectories that connect them. A SP edge is a shape composed by a string of SPs shapes generated in very close vicinity of an object's edge, as shown in Figure 5 (c) and (d). Here one may observe a string of sinking vectors, and in (d) a string of shrinking vectors. Note, the listed VF features are generated from the image geometric features, and the former set is a natural augmentation to the second set.

In [1,11] we proved that $\nabla\hat{u}$ generates sinking and saddle SPs, while $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ generate sinking, saddle, and springing SPs. It follows that $\nabla\hat{u}$ possess less variety and smaller number of SPs if compared with the other two VFs. Further, we showed in [1] that $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ have same number of SPs such that the springing SPs of $\nabla\hat{\phi}$ are mapped into the sinking SPs of the $\nabla\hat{\phi}$ VF and vice versa. At the same time the saddle SPs do not change shape, but their vectors have opposite directions in the two VFs. Furthermore, the saddle SPs may appear only in homogeneous regions, while the springing and sinking SPs are located on the boundary. Also, the trajectories and the separatrices of the two VFs have similar architectures, but their vectors have opposite directions [1]. The above-described SPs properties and mappings of the VFs $\nabla\hat{u}$, $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ can be observed in Figure 5.

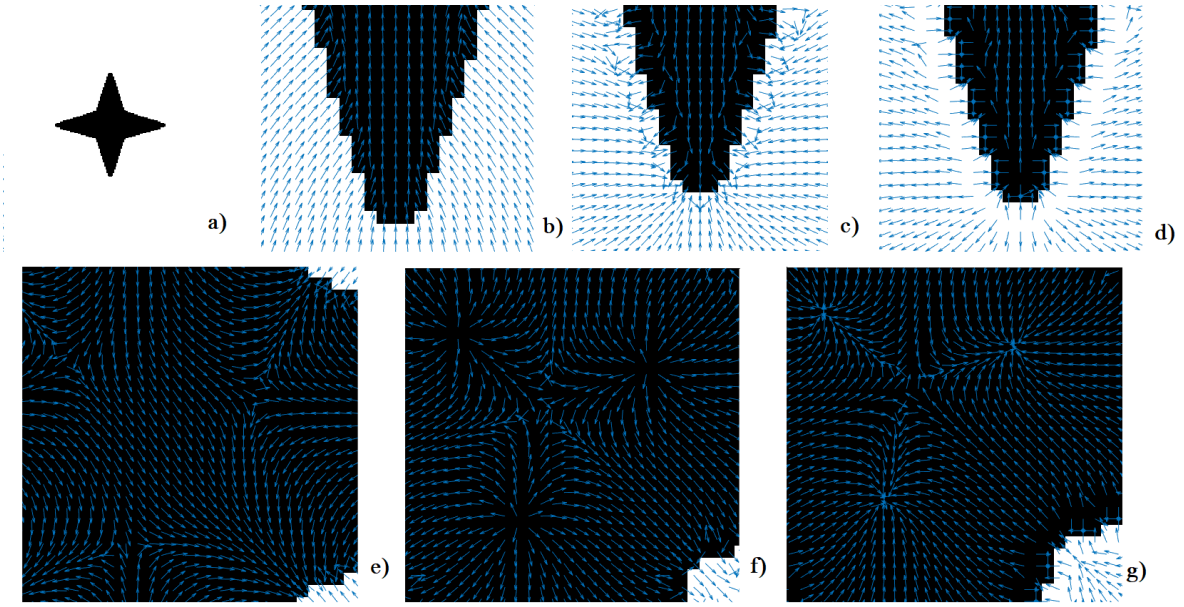


Figure 5. (a) Synthetic object. In the upper row are shown zooms of the lower branch, of the object in (a), with embedded VF; **(b)** $\nabla\hat{u}$; **(c)** $\nabla\hat{\phi}$; **(d)**$\nabla\hat{\psi}$. In the lower row are shown zooms of the core part, of the object in (a), with embedded VF: **(e)** $\nabla\hat{u}$; **(f)** $\nabla\hat{\phi}$; **(g)**$\nabla\hat{\psi}$.

 doi:10.20944/preprints202407.0874.v1

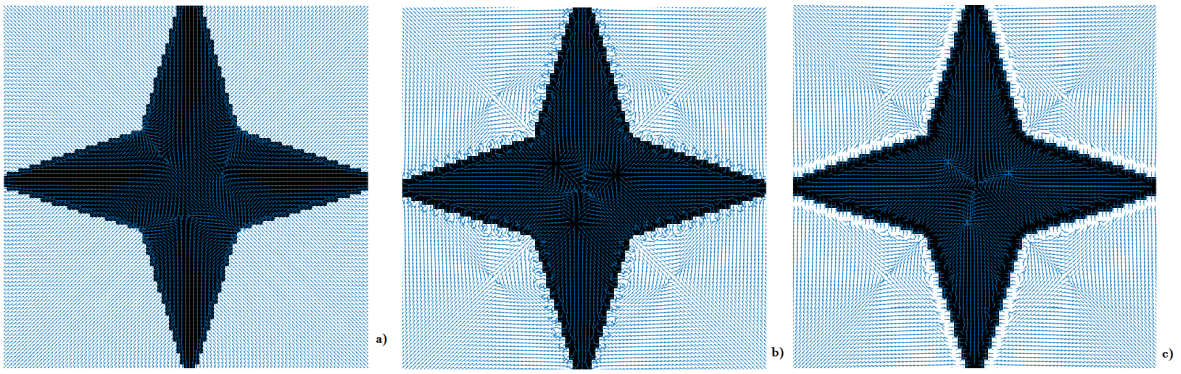


Figure 6. Overal view of the object in Figure 5 (a) with embedded VF: **(a)** $\nabla\hat{u}$; **(b)** $\nabla\hat{\phi}$; **(c)** $\nabla\hat{\psi}$.

Figures 5 (b) and 6 (a) show that the GVF $\nabla\hat{u}$ generates no SPs at convexity vertices, edges and external concavities but has sinking SPs at concavity corners Figures 5 (e). Further, $\nabla\hat{\phi}$ generates sinking SPs on the boundary convexity vertices and edges as shown in Figures 5 (c) and 6 (b) and no SP at the boundary concavities' corners (see Figures 5 (f)). Next, $\nabla\hat{\psi}$ generates sprinking SPs on the boundary convexity vertices and edges as shown in Figures 5 (d) and 6 (c) and no SP at the boundary concavities' corners (Figures 5 (g)). Further, $\nabla\hat{u}$ creates only saddle SP in the object's core homogeneous regions (see 5 (e)), while $\nabla\hat{\phi}$ generates springing SPs at the saddle SPs locations of $\nabla\hat{u}$, and a saddle SP between the springing (Figure 5 (f)). Also, the VF $\nabla\hat{\phi}$ generates saddle SPs at external concavities (Figure 6 (b)). The VF $\nabla\hat{\psi}$ preserves the saddle SPs of $\nabla\hat{\phi}$, swithing the vectors' directions, while replaces the springing with sinking SP (Figure 5 (f) and (g), Figure 6 (b) and (c)).

Note, Figure 6 provides an overall view of the 4-ray star from Figure 5 (a), where the three GVFs, with real shaped SPs, are embedded. One may observe that the objects' external concavity regions in Figure 6 (b) and (c) contain saddle SPs, but the vectors that create their transversal and hyperbolic trajectories have opposite directions. Once again one may observe that the objects with embedded VF $\nabla\hat{\phi}$ create edges of sinking SP, while $\nabla\hat{\psi}$ generates edges of springing SPs. The latter kind of SPs edges are better visually exhibited than the former edges of SPs.

We continue here after with the description of the properties and the locations of the SPs generated by the VFs with real and complex Eigenvalues. Recall that these VFs contain seven kinds of SPs: sinking, springing, saddle, spiral in (attracting), spiral out (repealing), and orbits with clockwise and counterclockwise directions. Examples of their shapes are shown in Figures 1, 2, 5, 7.

In [3] we proved that the CPs of the function $\hat{u}(x,y)$, which are the solution of Eq. 1, may map to any of the 7 kinds of SP shapes of $v_{\hat{u}}$. Further, in the same paper we proved that the CP of $\hat{u}$ and $\hat{\phi}$ map to the saddle SPs $v_{\hat{\phi}}$. Recently, we validated that the same holds for the CPs of $\hat{\psi}$ and the SPs of $v_{\hat{\psi}}$. Also, we validated that the SPs of $v_{\hat{\phi}}$ map to the SPs of $v_{\hat{\psi}}$ such that: springing SP maps to a sinking one and vice versa; saddle SPs do not change shape; spiral in map to spiral out and vice versa; and the clockwise orbits map to counterclockwise orbits and vice versa. These mappings imply that the VFs $v_{\hat{\phi}}$ and $v_{\hat{\psi}}$ possess same number of SPs and trajectory architectures, but the vectors that compose the two kinds of VFs architectures have opposite directions. Moreover, the positions of the SPs generated by the two VFs are in a close vicinity to each other, if projected on one and the same plane. Further, we note that the SPs edges of the VFs $v_{\hat{\phi}}$ and $v_{\hat{\psi}}$ are like each other as structures, but the vectors of these structures are opposite to each other. One may observe in Figure 7 parts (b), (c) that $v_{\hat{\phi}}$ possesses sinking SPs edges while $v_{\hat{\psi}}$ has springing SPs edges. Further $v_{\hat{\phi}}$ has saddle SPs points with coming in vertical transversals, while the horizontal are coming out as shown in part (e). On the other hand, the transversal of the saddle SPs of $v_{\hat{\phi}}$ at the same positions are build up by vectors with opposite directions as Figure 7 (f) shows.

 doi:10.20944/preprints202407.0874.v1

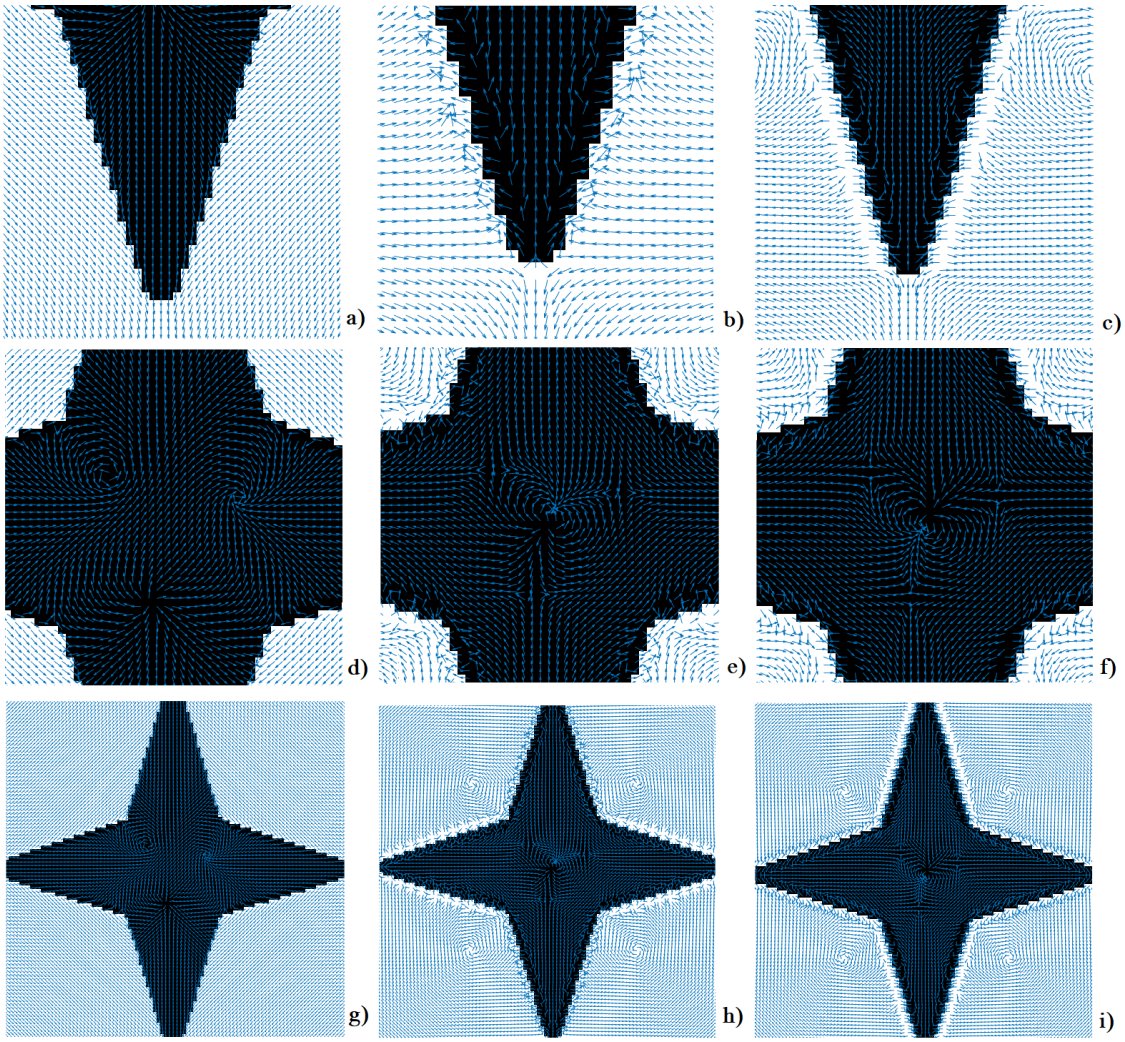


Figure 7. The object in Figure 5 (a) with embedded VF: $v_{\hat{u}}$ in the left column, parts **(a)**, **(d)** and **(g)**; $v_{\hat{\phi}}$ in the middle column, parts **(b)**, **(e)** and **(h)**; $v_{\hat{\psi}}$ in the right column, parts **(c)**, **(f)** and **(i)** .

Next, one may observe that the images with embedded VF $v_{\hat{u}}$ show no SPs located on the object exterior nor on its boundary. Hence, $v_{\hat{u}}$ does not create SPs edges, nor are SPs present at: objects boundary; on external concavities; inner convexities; nor at convexity vertices. However, any of the seven SPs may show up in the core part of the object. The above statements are validated in Figure 7 (a), (d), where we present a zoomed in portion of the lower ray and the core part of the four rays star from Figure 5 (a) with embedded VF $v_{\hat{u}}$. Note, that part (d) exhibits spiral out (the upper left SP), spiral in (the upper right SP), and an springing SP (the lower one). An overall view of the same object and VF is shown in (g) where the SPs are present only in the object core.

VF $v_{\hat{\phi}}$ generates spiral in SPs on the exterior concavities of the object (Figure 7 (h)). Also, saddle SPs are located at the convex vertices of the boundary as shown in part (b). Moreover, $v_{\hat{\phi}}$ generates edges of sinking SPs (Figure 7 (b)) and springing SPs (see the horizontal rays in part (h)). Further, the core interior of the object may contain several SP shapes from the entire set of seven kinds of SPs. As one may observe from Figure 7 (e), the core of the 4-ray star contains three saddle SPs, and a springing and sinking SPs between them. The upper two saddle SPs have outgoing transversal trajectories which go to the sinking SPs, while the incoming transversal of the lower saddle SP comes from the springing SP.

The last VF to consider is $v_{\hat{\psi}}$. Recall that, if embedded into an object, it has same number of SPs shapes and similar architecture, of its trajectories, as $v_{\hat{\phi}}$ does. Note, the vectors that build up the

 doi:10.20944/preprints202407.0874.v1

$v_{\hat{\psi}}$ trajectories and SPs have opposite directions to the vectors of $v_{\hat{\phi}}$. This could be observed in part Figure 7 (c) where at the convex vertex is located a saddle SP whose vectors are opposite to those of the saddle SP in part (b). Also, the four spiral out SPs in the external concavities of the star in part (i) are created with vectors whose directions are opposite to the directions of the vectors which created the spiral out SPs in the external concavities Figure 7 (h). Further, part (f) shows that the VF $v_{\hat{\psi}}$ generates five SPs in the core of the 4-rays star, as $v_{\hat{\phi}}$ does it in part (e). In (f) there are again three saddle SPs, but they have opposite vectors if compared with the saddle SPs in (e). The remaining two SP, in between the three saddles, are springing and sinking, but they switched positions, such that the sinking is below the springing as shown in part (f), while the former is above the latter in part (e) of Figure 7.

Following the above reasoning and observations we develop the diagram in Figure 8 to describe the mappings between the CPs of the functions $\hat{u}$, $\hat{\phi}$, $\hat{\psi}$ and the SPs of the six VFs if they are separately embedded in one and the same image. Note, the VFs having one to one correspondence have the same trajectory architectures, but vectors with opposite directions, as we described in the above paragraphs. Also, with the notion "partial" we denote that only part of the SPs of $\nabla\hat{u}$ map to the SPs of $\nabla\hat{\phi}$, because $\nabla\hat{u}$ has sinking SPs ($\hat{u}$ has maxima as proven in [1]) at concavity vertices (see Figure 5 (e)) while $\nabla\hat{\phi}$ does not have SPs at concavity vertices (see part (f)). Also, $\nabla\hat{\phi}$ has sinking SPs ($\hat{\phi}$ has maxima as proven in [1])) at convexity vertices (see Figure 5 (c)), while $\hat{u}$ does not (see part (b)).

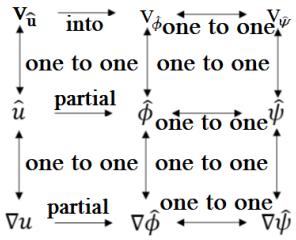


Figure 8. The mappings between the CPs of $\hat{u}$, $\hat{\phi}$, $\hat{\psi}$ and the SPs of the six VFs derived from the three functions, as well as the mappings between the SPs of the VFs.

Table 1 shows the distribution of the SPs of a VF across an image if the VF is embedded into the image. It is a guide designed to help the user select the VF to be embedded into an image database in order to provide the highest classification statistics. Also, sink denotes sinking SP(s), spring denotes springing SP(s), core denotes "core of an object", branches denote branches of objects, and edges denote "boundary edges".

Table 1. SPs location on image objects according to the embedded VF.

VF	SP/Location	SP/Location	SP/Location
$\nabla\hat{u}$	saddle / core	sinking/concavity corners	
$\nabla\hat{\phi}$,	saddle/core, branches, concavities	sink/convex vertices, edges	spring/core
$\nabla\hat{\psi}$	saddle/core, branches, concavities	sink/core	spring/edges, convex vertices
$v_{\hat{u}}$	saddle/core	sink/ core	spring/core;
$v_{\hat{u}}$	spiral (in and out)/core	orbits/homogenous regions	
$v_{\hat{\phi}}$, $v_{\hat{\phi}}$	saddle/core, convex vertices	sink/core, edges, branches	spring/core, edges, branches
$v_{\hat{\phi}}$, $v_{\hat{\phi}}$	spiral (in and out)/core, concavities, branches	orbits/homogenous regions	

 doi:10.20944/preprints202407.0874.v1

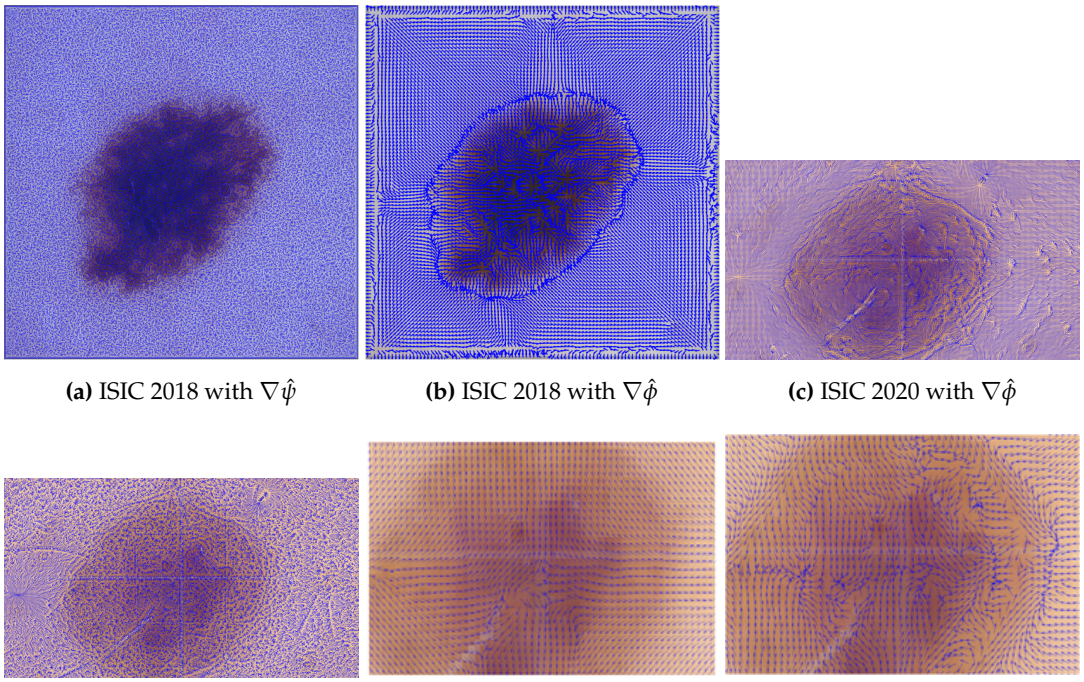
3. Repository of Image Databases with Embedded VFs

3.1. Original Datasets

In this section we detail the technical characteristics of four original datasets, into which we embedded the VFs defined above and used to obtain the results in [1–3]. Recently, we added to the repository the image database digit-MNIST [23] with embedded VF $v_{\hat{\psi}}$.

The ISIC 2018 [19] and 2020 [20] image databases contain skin lesion images along with malignant and benign labels for each skin lesion. For the original ISIC 2018 database, there are 3694 skin lesion images with dimensions ranging from 640x480 to 6670x4440 in JPG format. For the original ISIC 2020 database, there are 33126 skin lesion training images and 10982 skin lesion testing images with dimensions ranging from 640x480 to 6000x4000 in DCM format.

One primary use case for the ISIC2020 databases is to improve ML diagnosis of skin lesions either as malignant or benign. Thus, the database is broken down into two classes of images which are labeled by three dermatologists. ML is tasked with correctly labeling each skin lesion image into one of the two classes. Note that the training images are labeled by dermatologists, but the testing have no labels and classification is shown when results are uploaded to the database website [20]. Examples with embedded VFs from the ISIC skin lesion datasets are found in Figure 9.



(a) ISIC 2018 with $\nabla\hat{\psi}$ **(b)** ISIC 2018 with $\nabla\hat{\phi}$ **(c)** ISIC 2020 with $\nabla\hat{\phi}$

(d) ISIC 2020 with $\nabla\hat{\psi}$ **(e)** ISIC 2020 with $v_{\hat{u}}$ **(f)** ISIC 2020 with $v_{\hat{\phi}}$

Figure 9. Examples of ISIC 2018 and ISIC2020 images with embedded VFs.

The original COIL100 [16] database contains 7200 images. These images were produced by taking 100 common household objects and rotating each object 5° for a total of 72 rotations. This produced $72 \times 100 = 7200$ images in total, stored in PNG format. The images are sized with dimensions of 128x128 pixels.

One purpose of this database is to help train ML to recognize a variety of objects, and to recognize these objects regardless of the viewing angle. The different classes in this case are the 100 objects presented to the ML algorithm in 72 different rotations.

For embedding VFs to augment the image features of COIL100 database, we resized its images to 420x420. Examples from this database with embedded VFs are found in Figure 10.

The original Yale face database [18] contains 2414 face images taken under different lighting conditions. The original images are stored in the PNG format with dimensions $341x385$. Note that to

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 July 2024 doi:10.20944/preprints202407.0874.v1

conduct the experiments in [1] we have resized the original images to 192x168 and embedded the VFs again. Examples from this database are found in Figure 11.

The Yale face database allows ML to be trained on a set of face images under different lighting conditions in order to produce facial recognition. The classes in this case are each of 39 faces, with a picture of each taken with a different angle of lighting.

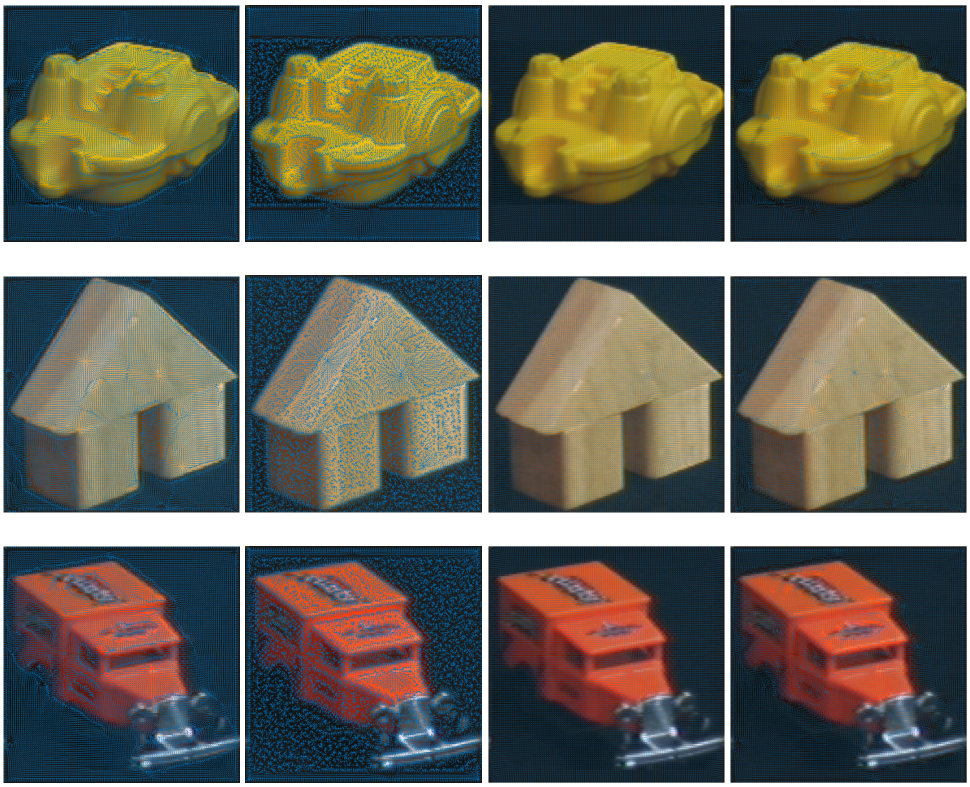
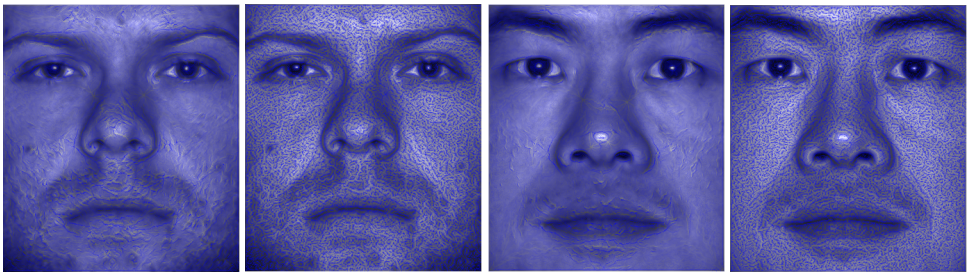


Figure 10. Examples of COIL100 Images with embedded, from left to right: $\nabla\hat{\phi}$, $\nabla\hat{\psi}$, $v_{\hat{u}}$, $v_{\hat{\phi}}$



(a) $\nabla\hat{\phi}$ **(b)** $\nabla\hat{\psi}$ **(c)** $\nabla\hat{\phi}$ **(d)** $\nabla\hat{\psi}$

Figure 11. Examples of YALE Face Database Images with embedded VFs.

3.2. Image Datasets with Augmented Image Features

Our new repository of image databases with embedded VFs, which augment the image features with SPs shapes, currently hosts several sets of image databases with embedded VFs. They are generated from the ISIC2018, ISIC2020, COIL100, and digit-MNIST databases with the help of the software ELPAC [12], which is included in the repository as well.

In particular, from the original ISIC2018 image database we have generated two ISIC2018 databases in PNG format with sizes 500x500 and 250x250. Then in each of them we embedded the VFs $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$. Further, we resized the ISIC2020 to 256x192, in PNG format, and embedded in

 doi:10.20944/preprints202407.0874.v1

every training image of the resized database the VF $\nabla\hat{\phi}$ or $\nabla\hat{\phi}$. Respectively, we named the new image databases as follows:

- ISIC2018$-500x500-\nabla\hat{\phi}$ and ISIC2018$-500x500-\nabla\hat{\psi}$
- ISIC2018$-250x250-\nabla\hat{\phi}$ and ISIC2018$-250x250-\nabla\hat{\psi}$
- ISIC2020$-train-\nabla\hat{\phi}$ and ISIC2020$-test-\nabla\hat{\phi}$
- ISIC2020$-train-\nabla\hat{\psi}$ and ISIC2020$-test-\nabla\hat{\psi}$.

In addition we made imprints of the ISIC2020 into the VFs $\nabla\hat{\phi}$, $\nabla\hat{\psi}$ and denoted them with $ISIC2020-imprint-\nabla\hat{\phi}$ and $ISIC2020-imprint-\nabla\hat{\psi}$. Note, in the repository we used the names "VF only" to facilitate the users who did not read the paper.

Also, we generated four image databases using the training and testing sets of ISIC2020 and embedding into them the VFs $v_{\hat{\phi}}$ and $v_{\hat{\psi}}$. Analogously, we named the datasets:

- ISIC2020$-train-v_{\hat{u}}$ and ISIC2020$-test-v_{\hat{u}}$
- ISIC2020$-train-v_{\hat{\phi}}$ and ISIC2020$-test-v_{\hat{\phi}}$

As well, in the resized COIL100 we embedded the VFs $\nabla\hat{\phi}$, $\nabla\hat{\psi}$, $v_{\hat{u}}$, $v_{\hat{\phi}}$, and $v_{\hat{\psi}}$ generating five new image databases whose image features are augmented with SPs shapes. According to the adopted titles convention we named the generated image databases with: $COIL100-\nabla\hat{\phi}$, $COIL100-\nabla\hat{\psi}$, $COIL100-v_{\hat{u}}$, $COIL100-v_{\hat{\phi}}$, and $COIL100-v_{\hat{\psi}}$.

To further validate the robustness of the idea that ML classification is boosted by embedding a VF into image database (which augments the image features with VF features), we added Gaussian noise to the original datasets COIL100 and ISIC2020. This Gaussian noise was generated using the built in MatLab function with a 0 mean and variance set to 0.001. The noisy images were embedded with VFs $v_{\hat{u}}$ and $v_{\hat{\phi}}$ respectively. Thus, we created the six new image datasets listed below:

- COIL100$-noise-v_{\hat{u}}$ and COIL100$-noise-v_{\hat{\phi}}$
- ISIC2020$-train-noise-v_{\hat{u}}$ and ISIC2020$-test-noise-v_{\hat{u}}$
- ISIC2020$-train-noise-v_{\hat{\phi}}$ and ISIC2020$-test-noise-v_{\hat{\phi}}$.

In [2] we applied our CNN and classified the image database COIL100$-noise-v_{\hat{u}}$ with added noise and embedded VF $v_{\hat{u}}$ with an accuracy of 92.97%. It is 2.9% higher than the accuracy 90.07% of classifying the original database COIL100 with the same CNN. The same held with ISIC2020$-test-noise-v_{\hat{u}}$ which was classified with an accuracy of 92.09% while the original image database ISIC2020 was classified with 86.67%.

On the other hand, ISIC2020$-test-noise-v_{\hat{\phi}}$ was classified with an accuracy of 85.26% which is 1.41% lower than the classification of the original ISIC2020. It holds because the skin lesion images contain many different structures and details. Hence adding noise and embedding VF features simply clutters the image and decreases the classification.

Recently we extended the repository adding the new image database digid-MNIST [23] with embedded VFs $\nabla\hat{\psi}$ and $v_{\hat{\psi}}$. The new image databases are named digit-MNIST-$\nabla\hat{\psi}$ and digit-MNIST-$v_{\hat{\psi}}$. The original digit-MNIST contains about $60,000$ images of size $28x28$, while the digit-MNIST-$\nabla\hat{\psi}$ and digit-MNIST-$v_{\hat{\psi}}$ conain images of sizes $400x400$.

To access the repository that contains the listed above image databases, ELPAC and SRWC software please use the URL:
https://www.tamuc.edu/projects/augmented-image-repository

Files are hosted as single ZIP files for each respective image database and software. Files stored in each respective image database ZIP are saved in the PNG format. New image databases with embedded VFs will be added in the future.

 doi:10.20944/preprints202407.0874.v1

3.3. Software

The image database repository contains links to the software tool ELPAC whose GUI is shown in Figure 12. This software provides sophisticated and mature capabilities for image segmentation based on the work in [12] and VFs embedding based on the works in [1–3]. Hence ELPAC incorporates the following features:

- Segmentation via an evolving contour directed by VF flow. To guide the active contour, with parameters customizeable under "Contour Size and Shape", a VF should be selected from the drop-down menu under "Vector Field Generation". The recommended choice is "Norm of Grad in Poisson Eq" ;
- Splitting and tracing contours around multiple objects for full image segmentation. Such options are available under "Splitting Options";
- Selecting any of the VFs from the "Vector Field Generation" drop-down menu. The six VFs described in section 2 are at the top of the list. The first 3 of them have real shaped SPs [1], while the next 3 have real and complex shaped SPs [3]. The selected VF will be embedded into the image file chosen using the Browse option.

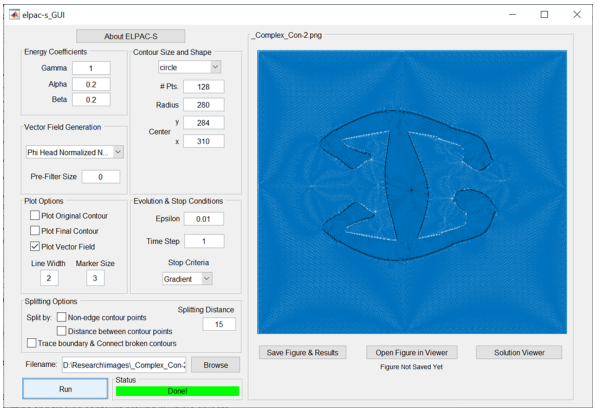


Figure 12. The GUI of ELPAC software [12]. The drop-down list below "Vector Field Generation" shows the list of VFs that could be embedded in an image.

In addition to the GUI for ELPAC, the repository provides code for batch VF embedding into an image database. Instructions for how to use the code and software are included in the repository. Furthermore, the code for ML classifier SRWC [1] is given as well.

4. Conclusions

The present paper describes our repository containing a total of twenty nine image databases built from embedding the VFs $\nabla\hat{\phi}, \nabla\hat{\psi}, v_{\hat{u}}, v_{\hat{\phi}}$ and $v_{\hat{\psi}}$ into four public image databases. Recall that six of the image databases are "injected" with a Gaussian noise and VF are embedded afterward. Also, four of the image databases are imprints of ISIC2020 into $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$. These image databases bring the advantage of enhancing classification statistics if compared with the classifications statistics of the original image databases, as shown in [1–3]. The advantage was validated by applying three ML methods, namely the sparse representation wavelet classification (SRWC) [1], the sparse representation classification with quaternions in wavelet domain (SRCQW) [1] and five CNNs [2].

Recall that the contributions of this paper, as formulated in the Introduction and validated through the text, are as follow:

1. Definition of the SPs locations according to the image objects, shown in Table 1
2. Generalization of the mappings between the SPs shapes, as shown in Figure 8, if the six VFs are separately embedded into the same image

 doi:10.20944/preprints202407.0874.v1

3. Definition of the new type of image and image database named "imprint of an image and imprint of an image database in a VF".

The contributions provide the user with the opportunities to choose the VF to be embedded into a particular image database to provide the advantage of increasing classification statistics if compared with the classification statistics of the original image database.

To understand the nature of the VFs and how they correlate to each other we determined in [1] the mappings between the CPs of $u, \hat{\phi}$ and $\hat{\psi}$ and the SPs of the GVFs $\nabla\hat{u}, \nabla\hat{\phi}, \nabla\hat{\psi}$. In [3] we investigated the mappings between the CPs of u and the SPs of $v_{\hat{u}}, v_{\hat{\phi}}$ as well as between the shapes of the SPs of the latter two VFs. The present paper contributes to the study of SP shapes mapping through the addition of the VF $v_{\hat{\psi}}$ into the diagram of SPs mappings.

Further the present paper introduced the notion "imprint of an image in a VF". The imprints of the malignant skin lesion from Figure 3 in the six VFs are shown in Figure 4 parts (b), (d), (f), (h), (j) and (l). One may observe that a single image creates different imprints in the different VFs. To validate the usefulness and the efficiency of the new notion we created the imprint of the ISIC2020 image database [20] in the VF $\nabla\hat{\phi}$ and named the new database $ISIC2020 - imprint - \nabla\hat{\phi}$. Then, we validated in [1] that classifying $ISIC2020 - imprint - \nabla\hat{\phi}$ gives an accuracy of 88.89%, while the accuracy of classifying the original image database is 84.21%.

From the above descriptions, derivations, observations, Figure 8 and Table 1 we derive information regarding the VFs' SPs shapes, the mappings between them, and the SPs locations in image objects. Hence, we conclude that the VF with the smallest number of SPs is $v_{\hat{u}}$, which generates SPs only in the core part of the interior (Figure 7 parts (a), (d) and (g)). The next is $\nabla\hat{u}$, which generates SPs in the core part of the interior and at the concavity corners (Figure 5 parts (b), (e) and Figure 6 (a)). Note, $v_{\hat{u}}$ may generate the whole set of seven types of SP shapes, while $\nabla\hat{u}$ generates only saddle and springing SPs as proven in [1,11].

The pair of VFs $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ generate the three kinds of real shaped SP (springing, singing and saddle) in the objects' exterior concavities, branches, boundary convexity vertices, and boundary edges. Therefore, $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ generate more SPs than $v_{\hat{u}}$ and $\nabla\hat{u}$ (Figures 5 (b) and (e), 6 (a), 7 (a), (d), and (g)). Recall that $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ have similar architectures created by the SPs and the trajectories connecting them, but the vectors that build these architectures are inverse to each other, as stated in section 2.2 and can be observed in Figures 5 c), f) and d), g), as well us 6 b) and c).

The last pair of VFs in our study is $v_{\hat{\phi}}$ and $v_{\hat{\psi}}$, and they generate almost the same number of SP as $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ in the object core, the exterior concavities, convexity vertices, branches, and boundary edges. Comparing Figures 5 and 6 with Figure 7 one may tell that the real Eigen valued VFs ($\nabla\hat{\phi}$ and $\nabla\hat{\psi}$) have springing and sinking SP at the convexities' vertices, and saddle SP in the external concave regions and the objects branches. On the other hand the complex Eigen valued VFs ($v_{\hat{\phi}}$ and $v_{\hat{\psi}}$) have saddle SPs on the convexities' vertices, spiral out SPs in the external concave regions and shrinking, sinking and spiral SPs at objects branches. Regarding the edges of SPs, those generated by $\nabla\hat{\phi}$ and $v_{\hat{\phi}}$ are similar. The same holds for $\nabla\hat{\psi}$ and $v_{\hat{\psi}}$. Finally, the VF architectures of $v_{\hat{\phi}}$ and $v_{\hat{\psi}}$ are alike to each other but have opposite vectors that create them.

The above analysis provides the knowledge to assist users in their choice of VF to be embedded into given image database and/or to make an imprint of the database into a certain VF in order to improve the classification statistics.

For the needs of paper [2] we embedded VFs $v_{\hat{u}}$ and $v_{\hat{\phi}}$ into the image databases ISIC2020 [20] and COIL100 [16]. Then, five contemporary CNN classifiers were applied to classify the original image databases and the image database with embedded VFs. The results show that the maximum accuracy of classifying the original databases ISIC2020 [20] and COIL100 [16] were 91.69% and 90.99 % respectively [2]. Classifying the databases with embedded $v_{\hat{u}}$ increased the maximum classification accuracies to 93.25% and 91.85% respectively [2]. However, embedding $v_{\hat{\phi}}$ into the original databases decreased the accuracies to 78.19% and 90.08% respectively.

 doi:10.20944/preprints202407.0874.v1

Please note that embedding $v_{\hat{\phi}}$ into the cluttered skin lesion images led to a drop of accuracy with about 13%, while embedding the same VF into COIL100, which has not a few details, led to a drop of 0.91%, but increased the precision from 70.46% to 71.57%.

Please note in the paper [2] for the image databases $COIL100-v_{\hat{u}}$, $ISIC2020-test-v_{\hat{u}}$ and $ISIC2020-test-v_{\hat{\phi}}$ we used slightly different notations, namely $v_{\hat{u}}(C)$, $v_{\hat{u}}(I)$ and $v_{\hat{\phi}}(I)$ respectively.

The advantage for the ML classification of image database with embedded VF is further validated in [1]. In this paper the VFs $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ where embedded into the ISIC2018 [19], ISIC2020 [20], COIL100 [16] and YaleB [18] creating ten new image databases, including the imprint of ISIC2020 in $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$. We classified the original image databases and their derivatives with embedded VFs by applying the software tools SRWC and SRCQW described in [1]. The accuracy of classifications of the VF embedded image databases increased except in the case of YaleB. For example, COIL100-$\nabla\hat{\phi}$ was classified by SRWC with an accuracy of 82.83%, while the original image database exhibited 81.29% [1]. Also, as mentioned above the accuracy of classification of $ISIC2020-imprint-\nabla\hat{\phi}$ by SRCQW was 88.89% while the original ISIC2020 was classified with 84.21%. Hence the ISIC2020 imprint classification showed an increase of 4.68%.

There was a drop in accuracy for the database YaleB with embedded $\nabla\hat{\phi}$ and $\nabla\hat{\psi}$ compared to the accuracy of the original YaleB because the objects' boundaries are not present in the images. Hence the SP edges, which play a role in the classification, were not generated by the VFs.

Our study continues by enlarging our repository with more image databases with embedded VFs and imprints of image databases into VFs. Also, we are developing VFs of the linear combinations of the functions $\hat{\phi}$ and $\hat{\psi}$ and will embed them into the presently used and other contemporary image databases. Further we will create an image database of SPs [22] to train ML classifiers and develop an approach to apply VFs to image data augmentation.

Acknowledgments: The authors would like to express their thanks to Mr. Jeremy Gamez, Chief Information Officer, and Mr. Jeff Faunce, Director, Center for IT Excellence at Texas AandM University-Commerce, The team from the Office of Marketing and Communications at Texas AandM University – Commerce for partially supporting this project through providing memory and webpage design.

Author Contributions: Conceptualization, Nikolay Metodiev Sirakov (NMS), Adam Bowden (AB); methodology, NMS; validation, NMS, AB; formal analysis, NMS, AB; data curation, AB; writing-NMS; review and editing-NMS, AB; supervision - NMS; project administration-NMS. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The link below provides access to a web page, where the reader may find the image datasets with embedded VFs ready to use and capable of enhancing the classification statistics of machine learning (ML) methods. Also, this page provides the software ELPAC [1,12] the reader may use to embed any of the above described six VFs into her/his image dataset(s) to boost the classification statistics of her/his classifier. The web page provides also a code for batch processing for embedding VF into an image database and code for sparce representation wavelets classification (SRWC) [1]. The user may apply the SRWC to classify original image dataset and the corresponding dataset with embedded VF. https://www.tamuc.edu/projects/augmented-image-repository.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ML	machine learning
VF	vector field
GVF	gradient vector field
SP	singular points
CP	critical points
NN	neural network
SRWC	sparce representation wevelets classification
SRCQW	sparce representation classification quaternions wevelets
CNN	convolutional NN
SL	skin lesion

 doi:10.20944/preprints202407.0874.v1

References

1. Sirakov, N.M., Bowden, A., Chen, M., Ngo, L.H., Luong, M. Embedding vector field into image features to enhance classification, *J of Computational and Applied Mathematics* **2024**, *V 441, 115685, ISSN 0377-0427, https://doi.org/10.1016/j.cam.2023.115685*.
2. Igbasanmi, O., Sirakov, N.M., Bowden, A. CNN for Efficient Objects Classification with Embedded Vector Fields, *In: García Márquez, F.P., Jamil, A., Ramirez, I.S., Eken, S., Hameed, A.A. (eds), Computing, Internet of Things and Data Analytics, pp. 297-309.* ICCIDA 2023. Studies in Computational Intelligence, vol 1145. Springer, Cham. *https://doi.org/10.1007/978-3-031-53717-2*, (2024)
3. Igbasanmy, O. D., Sirakov, N. M. On the Usefulness of the Vector Field Singular Points Shapes for Classification, *International Journal of Applied and Computational Mathematics, Springer, 10, 52. https://doi.org/10.1007/s40819-024-01679-8* , March (2024)
4. Tari, S., Genctav, M., From a non-local Ambrosio-Tortorelli phase field to a randomized part hierarchy tree, *J. of Mathematical Imaging and Vision, 49(1), 69–86*, (2014)
5. Li, B., Acton, S.: Automatic active model initialization via Poisson inverse gradient. *In: IEEE Trans. On Image Processing, vol. 17, 1406–1420*, (2008)
6. Ma, J., Ma, Y., Zhao, J., Tian, J. Image Feature Matching via Progressive Vector Field Consensus, *IEEE Signal Processing Letters, Vol. 22, No. 6*, June (2015).
7. Legaz-Aparicio, A.G., Verdu-Monedero, R., Angulo, J. Adaptive morphological filters based on a multiple orientation vector field dependent on image local features. *Journal of Computational and Applied Mathematics, 330, pp.965 - 981. 10.1016/j.cam.2017.05.001. hal-01688889*, (2018)
8. Leevy, Joffrey L and Khoshgoftaar, Taghi M and Bauder, Richard A and Seliya, Naeem. A survey on addressing high-class imbalance in big data, *Journal of Big Data* **2018**, *5, 1*, 1–30.
9. Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning, *J of Big Data-Springer* **2019**, *6, 60, https://doi.org/10.1186/s40537-019-0197-0*.
10. Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In Proc. of the AAAI Conference on Artificial Intelligence (AAAI), (2020).
11. Chen, M and Sirakov, N. M. Poisson Equation Solution and its Gradient Vector Field to Geometric Features Detection, In *Lecture Notes in Computer Science 11324* D. Fagan et al. Eds.; Springer Nature, 2018, pp. 36-48.
12. Bowden, A., Sirakov, M. N, Active Contour Directed by the Poisson Gradient Vector Field and Edge Tracking, *Journal of Mathematical Imaging and Vision – Springer, 63:665–680, 2021 https://rdcu.be/cflaI* , (2021)
13. Wei L., Eraldo R. Detecting Singular Patterns in 2-D Vector Fields Using Weighted Laurent Polynomial, *Pattern Recognition, 45(11), 3912-3925*, (2012)
14. Zhang, E., Mischaikow, K., Turk, G. Vector field design on surfaces, *ACM Trans. on Graphic, 25(4), 1294-1326*, (2006)
15. Sosinsky, A., Vector fields on the plane, *Retrieved from: http://ium.mccme.ru/postscript/s16/topology1-Lec7.pdf*, (2015)
16. Nene, S. A., Nayar, S. K., Murase, H., et al. Columbia object image library(coil-100), *Technical Report, CUCS-006-96*, (1996).
17. Argenziano, G., Soyer, H., De Giorgi, V.: Dermoscopy: A Tutorial. *Edra Medical Pub.*, New Media, Milan (2000)
18. Georghiades, A. S., Belhumeur, P. N., D. J. Kriegman, P. N. From few to many: Illumination cone models for face recognition under variable lighting and pose, *IEEE Trans. on PAMI, 23 (6) 643-515 660. doi:10.1109/34.927464*, (2001).
19. Codella, N., Rotemberg, V., Tschandl, P., Celebi, E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., Kittler, H., Halpern, A. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC) 2018; *https://arxiv.org/abs/1902.03368*, (2018).
20. International Skin Imaging Collaboration. SIIM-ISIC 2020 Challenge Dataset. Internat. Skin Imaging Collaboration. *https://doi.org/10.34970/2020-ds01*, (accessed May, 2023).
21. Siddiqi, K., Bouix, S., Tannenbaum, S. W. Hamilton–Jacobi skeletons, *Int. J. Comput. Vis. 48 (2002) 215*, em http://dx.doi.org/10.1023/A:1016376116653, (2002)

 doi:10.20944/preprints202407.0874.v1

22. Bina, T., Yib, L. CNN-based flow field feature visualization method. *Int. J. Performability Eng. 14(3), 434–444 . https://doi.org/10.23940/ijpe.18.03.p4.434444*, (2018).
23. digits-MNIST image database, *http://yann.lecun.com/exdb/mnist/*, last visited July (2024).