

Fixed Point Layers for Geodesic Morphological Operations

Santiago Velasco-Forero, Ayoub Rhim, Jesús Angulo

▶ **To cite this version:**

Santiago Velasco-Forero, Ayoub Rhim, Jesús Angulo. Fixed Point Layers for Geodesic Morphological Operations. British Machine Vision Conference, Nov 2022, London, United Kingdom. hal-03805667

HAL Id: hal-03805667
https://hal.science/hal-03805667

Submitted on 10 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FIXED POINT LAYERS FOR GEODESIC MORPHOLOGICAL OPERATIONS

Santiago Velasco-Forero, Jesús Angulo
Center for Mathematical Morphology
Mines ParisTech, PSL University
Fontainebleau, France

Ayoub Rhim
Ecole Nationale des Ponts et Chaussées
Champs-sur-Marne, France

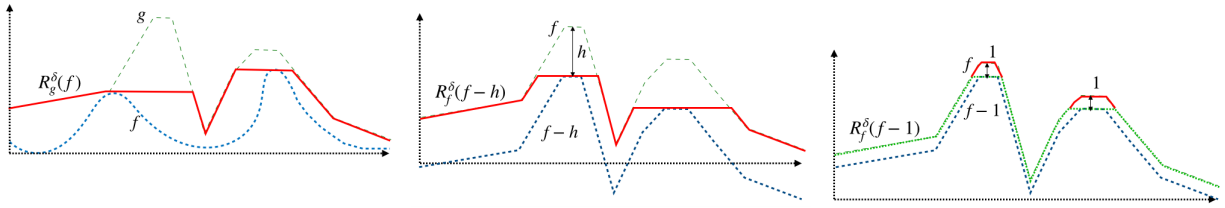
ABSTRACT

Morphological reconstruction is a contour-preserved geodesic transformation that is useful in many fields of image processing. On the other hand, deep learning methods achieved state-of-the-art performance in almost all computer vision tasks. This paper proposes new deep learning layers based on fixed-point morphological reconstruction operations. First, we show that they can be implemented in modern deep learning frameworks and analyse how they affect the learning process of gradient-based methods. Because of the Jacobian properties and the constraint nature of the morphological operators, our layers provide interpretability in both the output and the gradient flow. As examples of application, we consider the use of combining our layers and CNNs to a) improve the performance in the prediction of geometric attributes of objects on images, b) improve the robustness against additive random noise perturbation. Additionally, we study the case of *only one noise level and only one database* during training to analyse the generalisation capacity of the proposed layer.

1 Introduction

In most of the image understanding tasks such as recognition and segmentation, the best performance nowadays is obtained by deep Convolutional Neural Networks (CNNs). Most of these networks are multi-layer neural networks which consist of neurons with trainable weights locally shared (convolutions). However, noisy images have a negative impact on the training of CNNs and decrease the classification performance of these networks [17]. Noise removal from the images is an important issue in the image processing which is often done as a preprocessing step. In this paper, we deal with morphological operators by reconstruction, which are contour preserved operators [14, 11]. In general, morphological geodesic operators have shown their advantages when the detection of precise details is required, as in the case of many remote sensing applications [4], seeded image segmentation [8] and image dehazing among others [12]. We explore the use of these operators for deep learning (DL) architectures. To the extent of the author's knowledge this is the first paper dealing with geodesic operators in the context of deep learning. Previous work used geodesic transformation as a preprocessing task [2], Opening/Closing layers [10], Nonflat Dilation/Erosion layers [15], Learning Dilation for Adaptive Pooling [3], and morphological scale-spaces for scale equivariant neural networks [13]. The rest of the paper is organised as follows: we review geodesic morphological operators in Section 2. How to implement a geodesic morphological operation as layers for CNNs and the interpretation of its Jacobian in term of fixed points[1] and basin attraction of the transformation is presented in Section 3. Section 4 presents the experimental results in three different settings and Section 5 concludes the paper.

[1]Let X be any space and f a map of X, into X. A point $x \in X$ is called a *fixed point* for f if $x = f(x)$. The set of all fixed points of f, called the *fixed set* is denoted by $\texttt{Fix}(f)$



(a) Reconstruction of mask g from a marker f

(b) HMAX transform

(c) RMAX transform. The illustrated example use $\epsilon = 1$, but in practical implementation it can be a small number.

Figure 1: Reconstruction by dilation (1) in a mask g from a marker f. Note the geodesic morphological operations produce flat zones, and the difference with g allows to extract peaks. The peaks that are extracted are determined by the marker. For the HMAX transformation, peaks with contrast difference less than h are eliminated.

2 Geodesic Morphological Operations

2.1 Introduction

In this section, we firstly provide an overview of fixed-point geodesic morphological operations which achieve the so-called reconstruction from a function under a constraining of a second function. Then, we explain how these operations can be included as layers in a neural network, so that backpropagation can be performed. Finally, we present the main property of the Jacobian, that allows both a geometric interpretation and a strong robustness through noise. The main ingredient of morphological geodesic transformations is the *geodesic dilation*. Let us consider two numerical functions $f, g \in \mathcal{F}(\Omega, \mathbb{R})$, the set of functions mapping from space of points Ω to $\mathbb{R}$, the set of different possible values of the image. Let f, g be such that $f \leq g$, f is called in [14] the *marker* and g the *mask*. The geodesic dilation of size one of f with respect to g is denoted by $\delta_g^{(1)}(f)$ and is defined as the point-wise minimum between g and the elementary dilation $\delta_{\mathtt{SE}}$ in a given local neighbourhood SE (a.k.a. *structuring element*)[2],

$$\forall x \in \Omega, \quad \delta^{(1)}(f,g)(x) := \delta_g^{(1)}(f)(x) := \delta_{\mathtt{SE}}(f)(x) \wedge g(x) \tag{1}$$

where $\wedge$ denotes the minimum coordinate-wise operation. Following [14], the *reconstruction by dilation* of a mask g from a marker f is defined as the geodesic dilation of f with respect to g iterated until stability and is denoted by $R_g^\delta(f)$:

$$\forall x \in \Omega, \quad R^\delta(f,g)(x) := R_g^\delta(f(x)) := \underbrace{\delta_g^{(1)} \circ \ldots \circ \delta_g^{(1)}}_{k \text{ times}}(f(x)) \tag{2}$$

where k is such that $\delta_g^{(k)}(f) = \delta_g^{(k+1)}(f)$. The reconstruction by dilation extracts the *domes* or *peaks* of the mask which are marked by the marker. This is illustrated in Fig. 1(a). The *h-maxima* transform provides a filter to select signal maxima using a contrast criterion [14],

$$\forall x \in \Omega, \quad \mathtt{HMAX}_h(f)(x) = R_f^\delta(f(x) - h) \tag{3}$$

where $h \in \mathbb{R}$ is a parameter. $\mathtt{HMAX}_h$ transformation cuts local maxima overall the image given a parameter h. Finally, a common way to detect peaks [14], is the *regional maxima transformation* defined by

$$\forall x \in \Omega, \quad \mathtt{RMAX}(f)(x) = f(x) - R_f^\delta(f(x) - \epsilon). \tag{4}$$

Some examples of (3) and (4) are shown in Fig. 1(b) and (c).

2.2 Geodesic Operations for CNNs

In this section, we define layers for CNNs based on morphological reconstruction. We show they can be implemented as part of CNN architectures, and how the backpropagation can be achieved during the neural network optimisation. To make the presentation of our results easier, we consider SE a flat structuring element and we consider functions having length $n \in \mathbb{N}^+$. However, for grey-scale images in 2D or 3D the implementation is equivalent by considering the connectivity induced by the structuring elements SEs, and n the total number of pixels.

[2] A *dilation* is a particular case of order statistics filters, returning the maximum value within a moving window denoted by SE.

2.2.1 Interpretation of Jacobian matrix

For a multivariate, vector-valued function $\tau : \mathbb{R}^n \mapsto \mathbb{R}^n$, the Jacobian is a $n \times n$ matrix denoted by $\mathbf{J}_\tau$, containing all first order partial derivatives of the transformation τ. The row i of the Jacobian corresponds to the gradient of the i-th component of the output vector. It tells how the variation of each input variable affects the variation of the i-th component of the output of τ. We are interested in giving an interpretation for the Jacobian of reconstruction operation (2), because this is the fundamental element to understand the evolution process in gradient-based learning methods [7, 5]. The geodesic reconstruction (2) is the composition of two operations, so below we describe the Jacobian of each of them. Firstly, for the minimum-wise operation,

$$\mathbf{J}_{f\wedge g}(f(x)) = \begin{cases} 1 \text{ in } (i,i) & \text{if } f(x_i) \leq g(x_i) \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Secondly, for the elementary dilation,

$$\mathbf{J}_{\delta_{\mathtt{SE}}}(f(x)) = \begin{cases} 1 \text{ in } (i,j) & \text{for } x_j = \arg\max \delta_{\mathtt{SE}}(f(x_i)) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

In (5), the Jacobian is not null is pixels satisfying the condition *less or equal than*. In (6), the Jacobian indicates from which pixel comes the maximum values that the dilation has locally. We should note that in implementation by auto-differentiation in DL modules as Tensorflow or Pytorch, the Jacobian in (5) and (6) will have values different from zero *only* for the first element equal to the $\wedge$ or $\arg\max$ instead of the complete equivalence class. That is the same as local pooling by maximum is implemented nowadays. By using the chaining rule for the composition of functions, the Jacobian the geodesic dilation in (1) is

$$\mathbf{J}_{\delta^{(1)}(f,g)}(f(x)) = \mathbf{J}_{f\wedge g}(\delta_{\mathtt{SE}}(f(x)))\mathbf{J}_{\delta_{\mathtt{SE}}}(f(x)) = \begin{cases} 1 \text{ in } (i,j) & \text{if } x_j = \arg\max \delta_{\mathtt{SE}}(f(x_i)) \\ & \quad \text{and } \delta_{\mathtt{SE}}(f(x_i)) \leq g(x_i) \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

To compute the Jacobian of (2), one should consider operation in convergence, that is, when it is idempotent. Firstly, the Jacobian with respect to the marker f has zero value, in values that come from dilation, creating flat areas that can be associated with a local maximum in f. Specifically using the concept of *basin of attraction*[3], the Jacobian of (2) with respect to f is determined by

$$\mathbf{J}_{R^\delta(f,g)}(f(x)) = \begin{cases} 1 \text{ in } (i,i) & \text{if } f(x_i) = R^\delta(f,g)(x_i) \\ 1 \text{ in } (i,j) & \text{if } x_j \in \mathtt{BA}_{x_i}(\delta_g^{(1)}(f)) \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

and equivalent with respect to the mask g is

$$\mathbf{J}_{R^\delta(f,g)}(g(x)) = \begin{cases} 1 \text{ in } (i,i) & \text{if } g(x_i) = R^\delta(f,g)(x_i) \\ 1 \text{ in } (i,j) & \text{if } x_j \in \mathtt{BA}_{x_i}(\delta_g^{(1)}(f)) \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

We highlight that the basin of attraction in both (8) and (9) are flat zones, i.e., $x_j \in \mathtt{BA}_{x_i}(\delta_g^{(1)}(f)) \Rightarrow R^\delta(f,g)(x_j) = R^\delta(f,g)(x_i)$. For the marker, (8), these flat zones are related to some local maxima in f, and in (9) are related to some local minima of g. In other words, the flow of the gradient in the reconstruction layer depends on the membership of the basin of attraction, that is, for x_j from which x_i comes the value of the reconstruction, and if this value comes from f or g. Additionally, the number of ones in the i-th row, correspond to the cardinality of the basin of attraction of x_i. An illustrative example is given in Fig. 2. Thus, if one uses the geodesic reconstruction in a DL architecture, the backpropagation of the gradient passes through some maximum of f and some minimum of g, and the gradient value will be proportional to the cardinality of the basin of attraction. As a final observation, the number of zero values in $\mathbf{J}_{R^\delta_g(f)}(f(x)) + \mathbf{J}_{R^\delta_g(f)}(g(x))$ is equal to n. Similarly, for the h-maxima transform (3), where both marker and mask depend on f, the Jacobian indicates the position x_i from where each x_j has taken the value in convergence, i.e., $\mathbf{J}_{\mathtt{HMAX}_h(f)}(f(x)) = 1$ in (i,j) if $x_j \in \mathtt{BA}_{x_i}(\delta^{(1)}_{f-h}(f))$, and 0 otherwise.

[3]The *basin of attraction* of a fixed point x_α for f, denoted by $\mathtt{BA}_{x_\alpha}(f)$, is the interval $[a,b]$ if for all $x_0 \in [a,b] \Rightarrow \lim_{k\to+\infty} \underbrace{(f \circ f \circ \cdots \circ f)}_{k \text{ times}}(x_0) = f^{(k)}(x_0) = f(x_\alpha)$, where k is such that $f^{(k)}(x) = f^{(k+1)}(x)$ for all x.

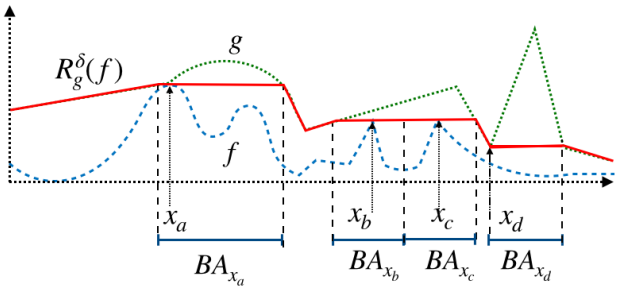


Figure 2: Basins of attraction with cardinality greater than one. $\texttt{BA}_{x_a}$, $\texttt{BA}_{x_b}$ and $\texttt{BA}_{x_c}$ contribute to the Jacobian with respect to the mask f in (8) and are associated with a local maxima of f. The $\texttt{BA}_{x_d}$ contributes to the Jacobian with respect to the marker g (9), and is associated with a local minimum of g.

2.2.2 Implementation

Firstly, we note that the dilation by a flat structuring element can be implemented as a MaxPooling layer without strides. Secondly, the main difficulty is the stability test in (2). This is equivalent to finding the fixed point of the operator (1). We would like to note that [6] and [1] seek to construct layers that are fixed points for a set of training data, which is fundamental in various optimization problems. However, in our case, the morphological reconstruction can be implemented by means of a control flow function [18] as it is shown in supplementary material for the case of 2D images.

3 Experiments

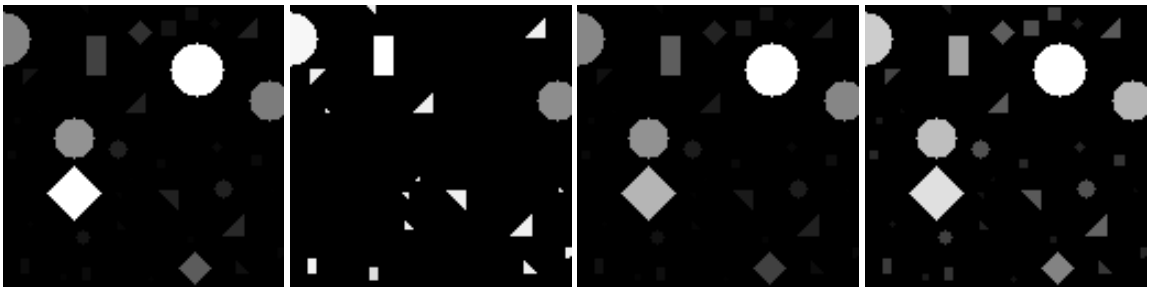
This section seeks to illustrate the benefits of using reconstruction operators in the context of CNNs. One thinks that the geometrical understanding of the effect of the reconstruction layer allows the design of architectures that are better adapted to specific problems. In this way two examples are presented: First, the use of the reconstruction operator (2) in the case of networks that learn geometric attributes about objects; second, the use of HMAX (3) and RMAX (4) to train a denoising layer which is robust to different levels and types of noise.

3.1 Learning geometric attributes on simple objects

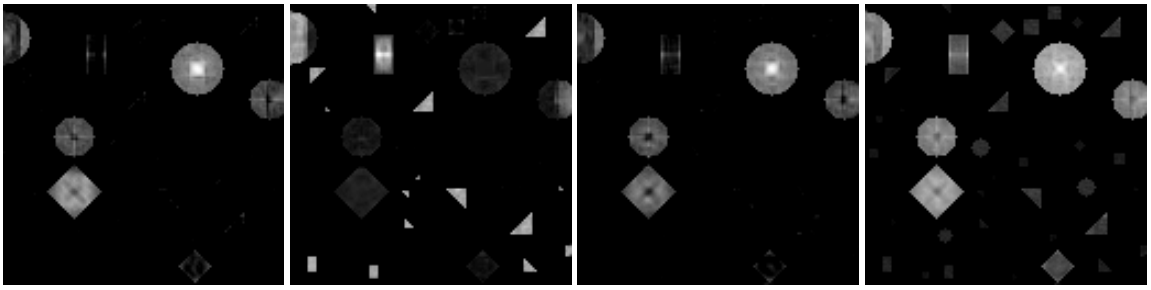
In this experiment, CNNs are trained in order to learn geometric attributes on images composed of simple shapes as it is shown in Fig 3. Each example is a random image with no overlapping objects with random size following an uniform distribution between $[3, 20]$ pixels (the image size is 128×128 pixels). As a manner of example, we consider the following geometrical attributes: Area, Perimeter, Area of Bounding-Box and Eccentricity[4]. 1024 random images are generated for training and 128 for validation. Two models composed of three convolutional layers of kernel size 5×5 with 24 filters with Relu activation functions are trained [5] to predict the value of a geometric attribute, with the difference for the model denoted as $\texttt{CNN}_R$ a reconstruction by dilation (2) is included of the last layer with the input image used as mask. This has two benefits: First, the result of the prediction is homogeneous within each of the objects due to the reconstruction process (Fig. 2). Second, according to the analysis of the Jacobian matrix, only the local maxima of the prediction will have an effect during learning, which should simplify the task. One can note that CNN easily predicts a value for shape in the marker of $\texttt{CNN}_R$, Fig. 3(i)-(l) than a constant value per shape in CNN Fig. 3(e)-(h). Note that each simple shape forms a basin of attraction with size equal to the area of shape. A quantitative comparison is given in the Table 1 using the average of the loss function in ten random replicates of the experiment. Finally, including geodesic operators has a cost in training time per epoch (batch size of 24 images) in a single GPU from 2.1 seconds for CNN to 5.2 seconds for $\texttt{CNN}_R$. In inference, the increase in computational cost is less important from 35.5ms to 39.9ms on average for images of 128×128 pixels.

[4]Eccentricity of the ellipse that has the same second-moments as the object. The eccentricity is the ratio of the focal distance (distance between focal points) over the major axis length. The value is in the interval [0, 1). When it is 0, the ellipse becomes a circle (i.e., isotropic object).

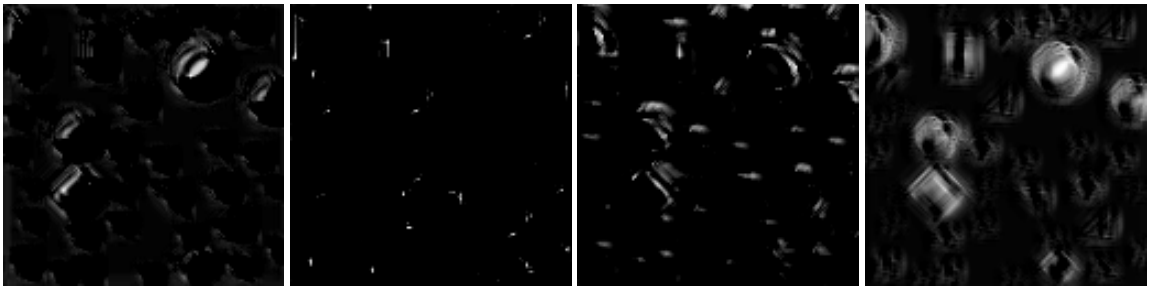
[5]The mean squared error is used as loss function, Adam optimiser, learning rate of 0.001, learning rate schedule by a factor of 0.1 with a patience of five epochs, and an early stopping with patience of ten epochs.



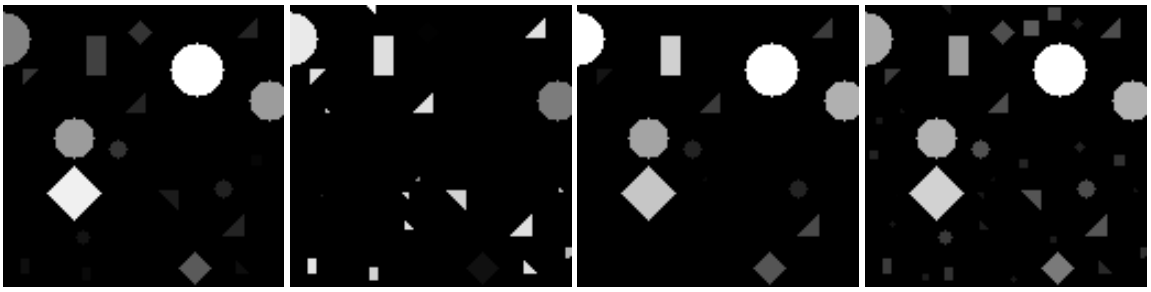
(a) Bounding box area (b) Eccentricity (c) Area (d) Perimeter



(e) Prediction CNN (f) Prediction CNN (g) Prediction CNN (h) Prediction CNN



(i) Marker of (m) (j) Marker of (n) (k) Marker of (i) (l) Marker of (j)



(m) Prediction $\texttt{CNN}_R$ (n) Prediction $\texttt{CNN}_R$ (o) Prediction $\texttt{CNN}_R$ (p) Prediction $\texttt{CNN}_R$

Figure 3: a) Example of a random image containing geometrical shapes as Circles, Squares, Triangles and Rectangles. b) Bounding box Area c) Eccentricity d) Area e) Perimeter. f) Example of prediction for the attribute perimeter e) for a CNN in f) and the proposed $\texttt{CNN}_R$ in g). Both trained models in f) and g) have the same number of parameters. $\texttt{CNN}_R$ homogenises the results of the CNN inside each object, drastically reducing the validation loss in Table. 1

Table 1: Quantitative comparison of Experiment 3.1. The average MSE over ten repetitions in the validation set is reported. CNN and $\texttt{CNN}_R$ models have the same number of parameters.

Attribute	CNN	$\texttt{CNN}_R$	Improvement
Area	0.001084	0.000546	49.61%
Perimeter	0.000683	0.000248	64.36%
Bounding Box Area	0.000504	0.000474	6.08%
Eccentricity	0.003715	0.000301	91.87%

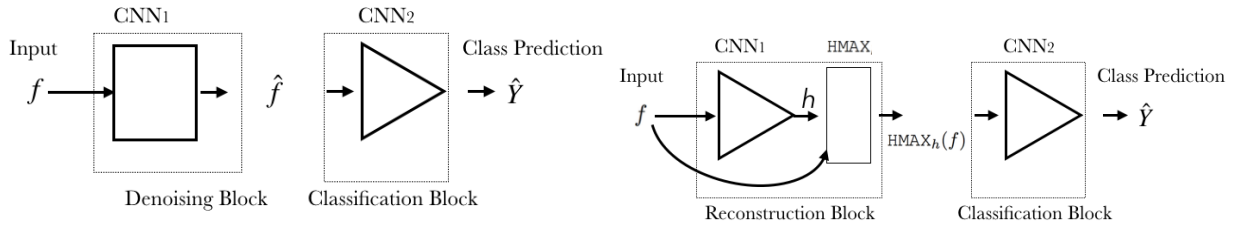


Figure 4: Left: $\texttt{CNN} - \texttt{CNN}$ model for denoising. The first block is trained to learn how to denoise an input image. The second is trained for classification. Right: Proposed model is composed of two blocks. Reconstruction block: it is composed of a CNN which computes from an input f a single real value h. The output is $\texttt{HMAX}_h(f)$. Classification block: A block of supervised classification. In both models, the second block is trained for classification without updating the denoising block.

3.2 Denoising (Only one noise level and only one database)

Let us consider an image perturbed by additive positive random noise. This noise implies the creation of local maximum over the image. These peaks contain strong information about the noise distribution. The HMAX transform in (3) gives more importance in the gradient to fixed points, thus using information about noise itself. This allows you to learn the noise directly from the peaks. Accordingly, our experiments aim to show that it is possible to train a CNN to learn the parameter h following the noise strength. At the test stage, we analyse the capacity of generalisation for a wide range of dataset and noise levels. Our proposed model has two components (Fig. 4): 1) a reconstruction block, which for an input image f estimates a value of h with a CNN and computes the $\texttt{HMAX}_h(f)$ transformation on an input image; 2) A classification block, using training a CNN using as input the $\texttt{HMAX}_h(f)$. At first impression one might think that it is enough to train the network in the supervised case, that is, using categorical cross entropy. However, you should train a large range of noise levels. To avoid this issue, we train on *only one noise level and only one database* (as a manner of example, we have used the MNIST database). In all the following experiments, the ADAM optimiser has been used during 50 epochs, with an initial learning rate of 0.001 and reducing it by a factor of 0.1 after ten epochs without improving on the validation loss. Models and reproductive code are available in the authors repository[6]. In the training stage of the reconstruction block (Fig. 4), we propose to minimise the following loss function, $\texttt{Loss}(f, \hat{f}) = ||f - \hat{f}||_2^2 + \alpha||\texttt{RMAX}(f)f - \texttt{RMAX}(\hat{f})\hat{f}||_2^2$ denoting $\hat{f} = \texttt{HMAX}_h(f)$. The first term is an *attached to data*, the second one looks for local large maxima to be preserved after HMAX transformation (See Fig. 5 for some examples) and α is a hyperparameter to trade-off the effect of each term ($\alpha = 0.75$ in our experiments). The pertinence of our proposition can be observed in Figure 7 when we have used the reconstruction block trained only in MNIST to estimate the value of h on different image databases at different noise levels (which have not been shown in the training phase). The estimated value of h follows the noise level, in pretty different datasets including CIFAR10, 91IMAGES[16] and BSD300 [9]. We highlight that for colour images the reported h is the average of the estimation channelwise. For a quantitative comparison, we illustrated the impact of the HMAX layer in a classification network, and show that it can provide a better robustness through noise than a classic CNN classification network. We have also compared with a CNN model trained to remove noise followed by a classification block denoted as $\texttt{CNN} - \texttt{CNN}$ model. For results shown in Fig. 6, denoising and reconstruction block has been pretrained on MNIST, and only the classification block is updated by a categorical cross-entropy loss function. In Fig. 6, we have also included a training with data augmentation by the type of noise with σ between 0 and 0.2 for fair comparison. In all the explored scenarios, the proposed model is more robust than classical and data augmentation approaches. In (a-c) cases in Fig. 6, our proposed reconstruction block is more robust than a denoising block.

[6]https://github.com/Jacobiano/GeodesicMorphological

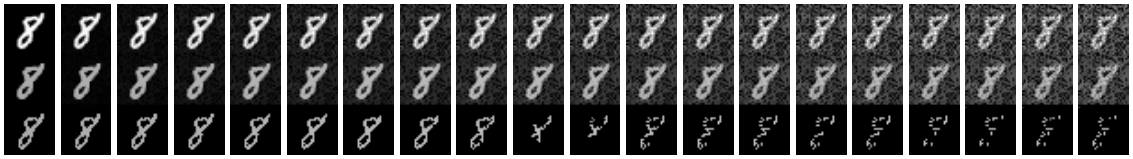
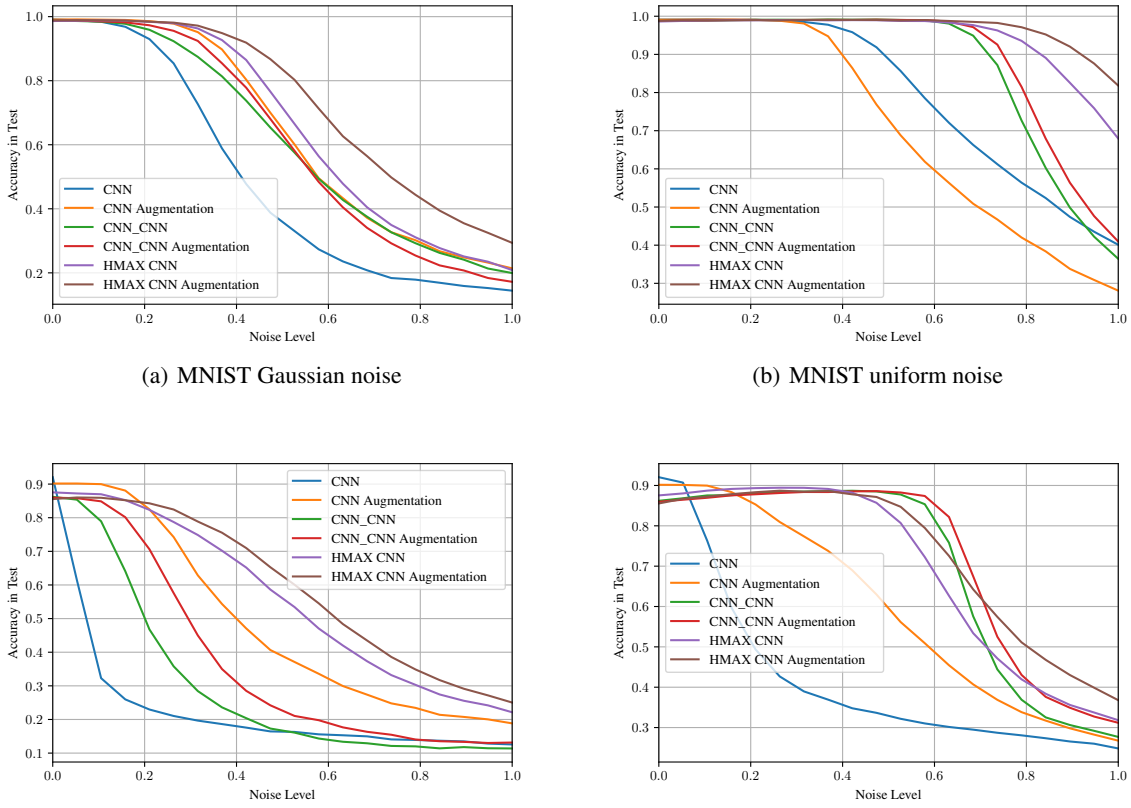


Figure 5: First Row: Example of images with level of uniform noise between zero to one where it has been used in Fig. 6(b). Second Row: Output of proposed reconstruction block $\texttt{HMAX}_h(\cdot)$ where h is estimated by $\texttt{CNN}_1$ (Fig. 4). Third Row: $\texttt{RMAX}(\texttt{HMAX}_h)$ used in the proposed loss function.



(a) MNIST Gaussian noise

(b) MNIST uniform noise

(c) Fashion MNIST Gaussian noise

(d) Fashion MNIST uniform noise

Figure 6: Classification accuracy for MNIST and Fashion MNIST with additive Gaussian and Uniform noise with $\mu = 0$ and $\sigma \in \{0., 0.05, \ldots, 1\}$. The reconstruction block (Fig.4)(b) has been trained *only* on MNIST with additive noise distributed as an absolute value zero-mean Gaussian with $\sigma = 0.1$. For fair comparison, models training with augmentation by additive random Gaussian noise at $\mu = 0$ and σ between 0 and 0.2 and a CNN-based model as in (Fig.4)(a) have been included.

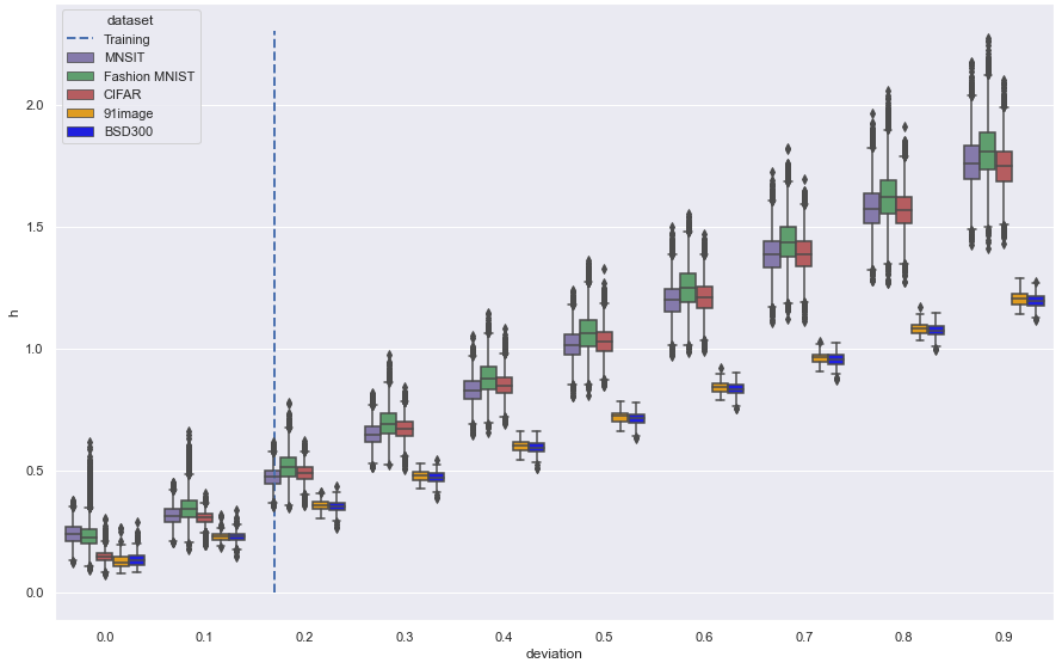


Figure 7: Predicted h values for same type of noise on dataset MNIST, Fashion MNIST, CIFAR10, 91IMAGES[16] and BSD300 [9] for $\sigma \in \{0, 0.1, \ldots, 0.9\}$. The proposed reconstruction block network succeeded in predicting the noise strength for several datasets despite having been trained *only* at the level $\sigma = 0.2$ and *only* on MNIST (dotted line).

4 Conclusion

In this work, a fixed-point geodesic morphological layer is proposed in the context of CNNs. By analysing its Jacobian matrix, we showed that our proposition has the strong property of focusing on local extrema during training. Additionally, we have compared its robustness for the application to the classification of noisy images in the scenario where the layer is trained in *only* one dataset and *only* one noise level. We hope that our problem formulation, our approach and our comparison will encourage future works on this research topic.

Acknowledgement

This work was granted access to the HPC resources of IDRIS under the allocation AD 011012212R1. This work has been supported by Fondation Mathématique Jacques Hadamard (FMJH) under the PGMO-IRSDI 2019 program.

References

[1] M. Blondel, Q. Berthet, M. Cuturi, R. Frostig, S. Hoyer, F. Llinares-López, F. Pedregosa, and J.-P. Vert. Efficient and modular implicit differentiation. *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2022.

[2] E. Decencière, S. Velasco-Forero, F. Min, J. Chen, H. Burdin, G. Gauthier, B. Laÿ, T. Bornschloegl, and T. Baldeweck. Dealing with topological information within a fully convolutional neural network. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 462–471. Springer, 2018.

[3] G. Franchi, A. Fehri, and A. Yao. Deep morphological networks. *Pattern Recognition*, 102:107246, 2020.

[4] P. Ghamisi, E. Maggiori, S. Li, R. Souza, Y. Tarablaka, G. Moser, A. De Giorgi, L. Fang, Y. Chen, M. Chi, et al. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, markov random fields, segmentation, sparse representation, and deep learning. *IEEE Geoscience and Remote Sensing Magazine*, 6(3):10–43, 2018.

[5] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[6] Y. Jeon, M. Lee, and J. Y. Choi. Differentiable forward and backward fixed-point iteration layers. *IEEE Access*, 9:18383–18392, 2021.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] T. Lei, X. Jia, T. Liu, S. Liu, H. Meng, and A. K. Nandi. Adaptive morphological reconstruction for seeded image segmentation. *IEEE Transactions on Image Processing*, 28(11):5510–5523, 2019.

[9] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[10] R. Mondal, M. S. Dey, and B. Chanda. Image restoration by learning morphological opening-closing network. *Mathematical Morphology-Theory and Applications*, 4(1):87–107, 2020.

[11] L. Najman and H. Talbot. *Mathematical morphology: from theory to applications*. John Wiley & Sons, 2013.

[12] S. Salazar-Colores, E. Cabal-Yepez, J. M. Ramos-Arreguin, G. Botella, L. M. Ledesma-Carrillo, and S. Ledesma. A fast image dehazing algorithm using morphological reconstruction. *IEEE Transactions on Image Processing*, 28(5):2357–2366, 2018.

[13] M. Sangalli, S. Blusseau, S. Velasco-Forero, and J. Angulo. Scale equivariant neural networks with morphological scale-spaces. In *International Conference on Discrete Geometry and Mathematical Morphology*, pages 483–495. Springer, 2021.

[14] P. Soille. *Morphological image analysis: principles and applications*. Springer Science & Business Media, 2013.

[15] S. Velasco-Forero, R. Pagès, and J. Angulo. Learnable empirical mode decomposition based on mathematical morphology. *SIAM Journal on Imaging Sciences*, 15(1):23–44, 2022.

[16] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.

[17] J. Yim and K.-A. Sohn. Enhancing the performance of convolutional neural networks on quality degraded datasets. In *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2017.

[18] Y. Yu, M. Abadi, P. Barham, E. Brevdo, M. Burrows, A. Davis, J. Dean, S. Ghemawat, T. Harley, P. Hawkins, et al. Dynamic control flow in large-scale machine learning. In *Proceedings of the Thirteenth EuroSys Conference*, pages 1–15, 2018.

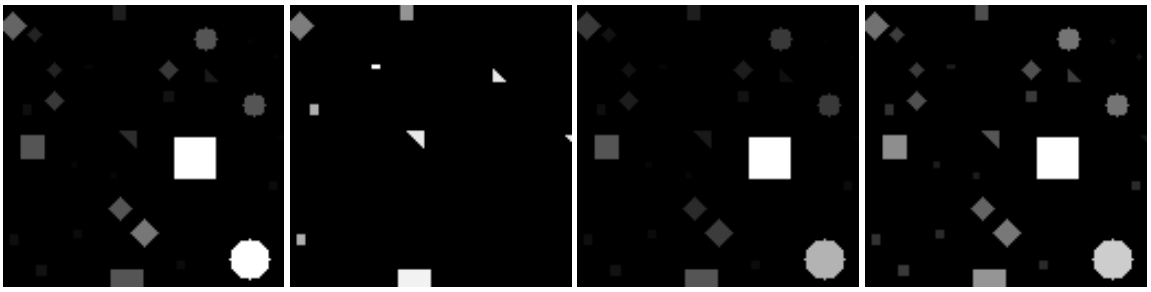
A Tensorflow Implementation

```
def condition_equal(last,new,image):
    return tf.math.logical_not(
    tf.reduce_all(tf.math.equal(last, new)
    ))
def update_dilation(last,new,mask):
    return [new, geodesic_dilation_step
    ([new, mask]), mask]
def geodesic_dilation_step(X):
    return Minimum()([MaxPooling2D(
    pool_size=(3, 3),strides=(1,1),
    padding='same')(X[0]),X[1]])
def geodesic_dilation(X,steps=None):
    rec = X[0]
    _,rec,_=tf.while_loop(condition_equal,
                          update_dilation,
                          [X[0], rec, X[1]],
                  maximum_iterations=steps)
    return rec
```

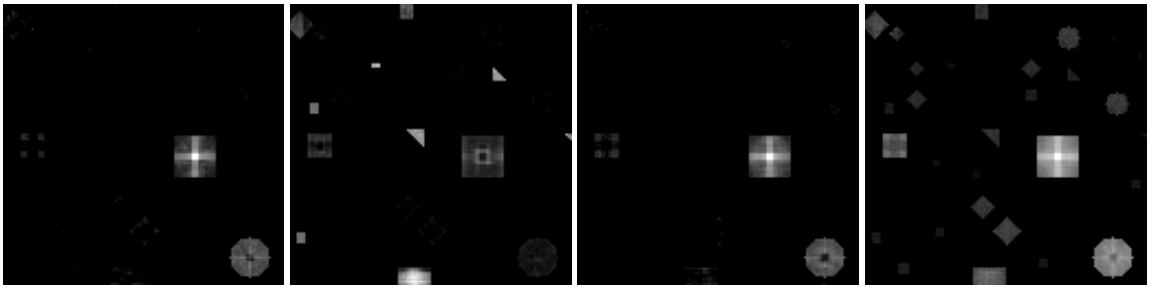
Figure 8: Geodesic Reconstruction implementation via `while_loop` function

B Complementary Examples

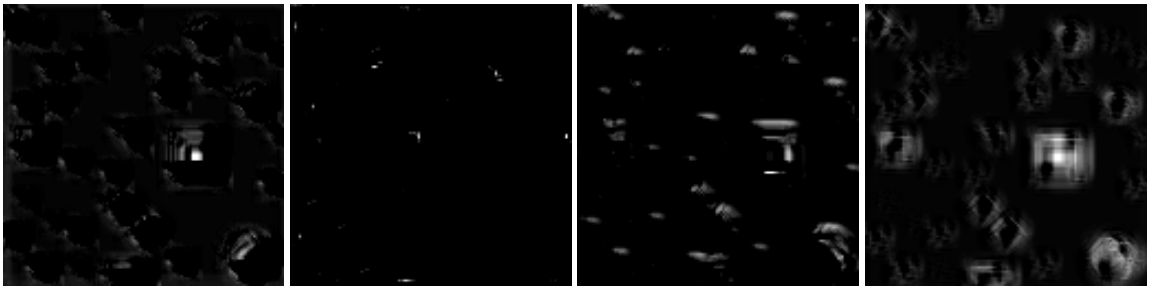
Another example of the prediction for the different models considered is given in Figure 9. The training curve for the CNN model and the proposed model CNN_R is presented for six repetitions of the experiment Figure 10. One can clearly notice that the inclusion of reconstruction facilitates convergence in fewer epochs and at a lower value.



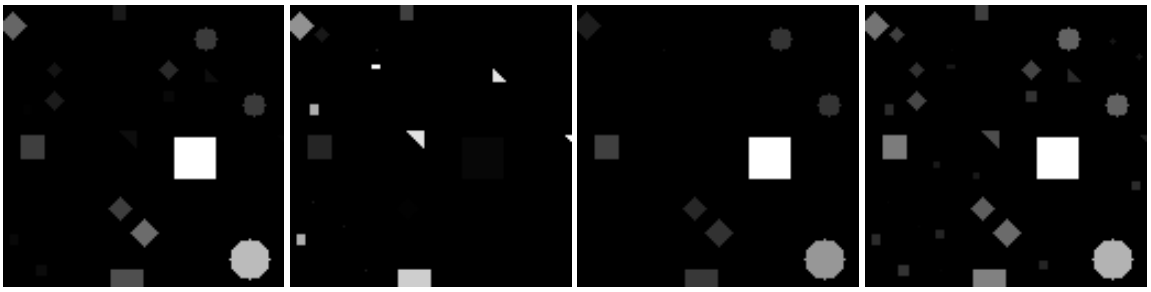
(a) Bounding box area (b) Eccentricity (c) Area (d) Perimeter



(e) Prediction CNN (f) Prediction CNN (g) Prediction CNN (h) Prediction CNN



(i) Marker of (m) (j) Marker of (n) (k) Marker of (i) (l) Marker of (j)



(m) Prediction $\texttt{CNN}_R$ (n) Prediction $\texttt{CNN}_R$ (o) Prediction $\texttt{CNN}_R$ (p) Prediction $\texttt{CNN}_R$

Figure 9: a) Example of a random image containing geometrical shapes as Circles, Squares, Triangles and Rectangles. b) Bounding box Area c) Eccentricity d) Area e) Perimeter. f) Example of prediction for the attribute perimeter e) for a CNN in f) and the proposed $\texttt{CNN}_R$ in g). Both trained models in f) and g) have the same number of parameters. $\texttt{CNN}_R$ homogenises the results of the CNN inside each object, drastically reducing the validation loss in Fig. 10

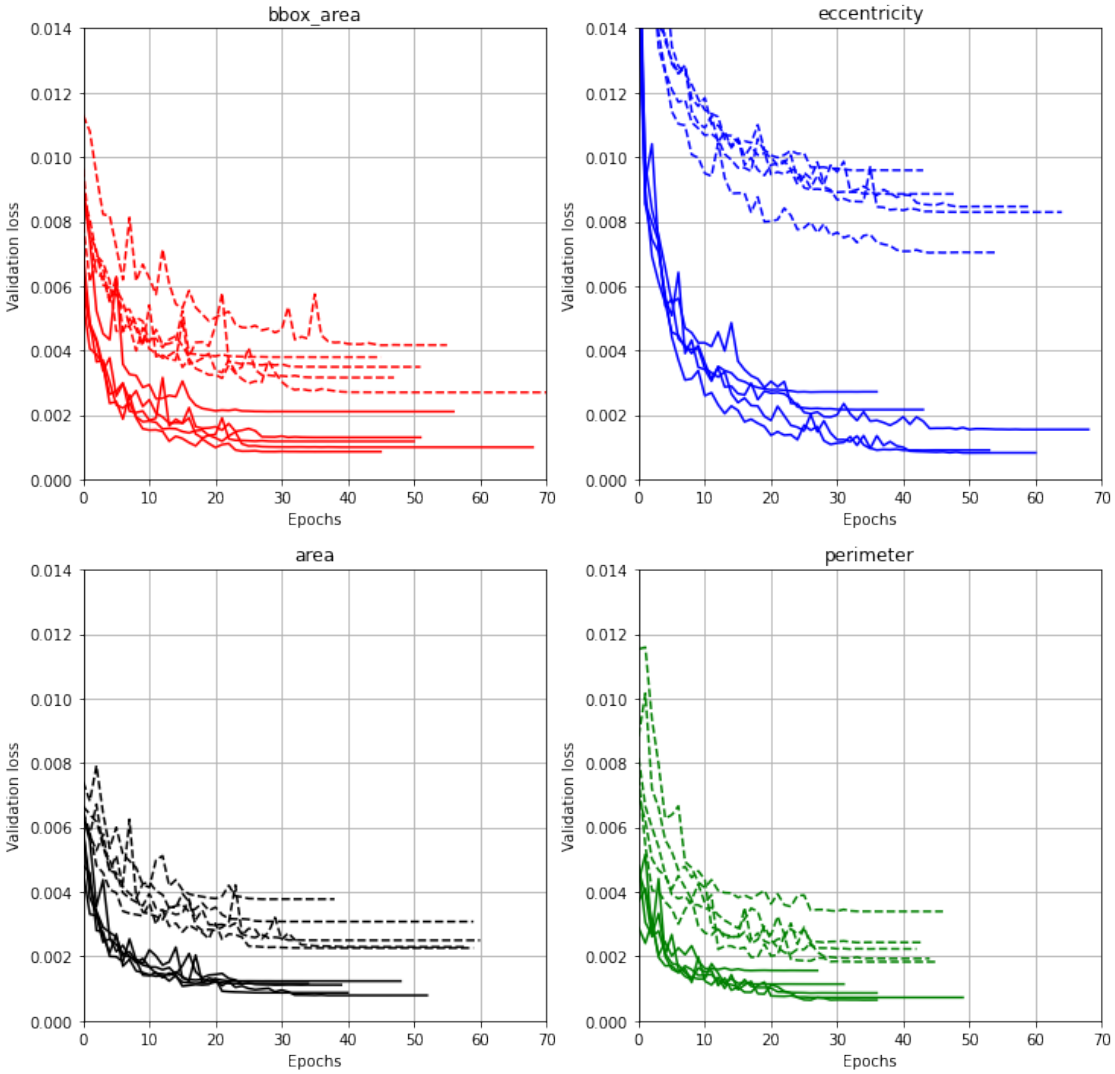


Figure 10: Several repetitions of the training protocol by varying the random initialisation. Dotted lines correspond to traditional CNN, and plain lines to CNN_R, i.e, with a reconstruction as the last layer. Note that CNN and CNN_R models have the same number of parameters. In the explored scenarios, the inclusion of the geodesic reconstruction helps to converge to a model with lower validation loss.