Comparing shape-based and pixel-based approaches for melanoma detection

Andréa Davila[1,2], Issam-Ali Moindjié[1,3], and Cédric Beaulac[1*]

[1] Département de mathématiques, UQAM, Montréal, Canada
[2]École des Mines de Saint-Étienne, Saint-Étienne, France
[3]LAMPS, UPVD, Perpignan, France
[*]Corresponding author, email: beaulac.cedric@uqam.ca

Abstract

In recent years, both the number and the size of image datasets have grown at an uncontrollable rate. This creates a serious challenge for the analysis of image databases, especially for researchers who may not have access to expensive and powerful supercomputers. In this research report, we study an alternative to pixel-based analysis: a functional representation of the shapes of objects within images. This representation offers several advantages. By being of much lower dimension, it greatly reduces the computational cost of subsequent analyses; it is also far more interpretable and can leverage the extensive set of tools already developed for the analysis of multivariate functional data. We investigate this shape-based approach in the context of classification using a real dataset, the HAM10000 dataset, and our results demonstrate a clear computational benefit with similar predictive power.

1 Introduction

Photos and images have been attracting an increasing amount of interest in the data analysis literature as this type of data structure becomes more accessible. Images are an important source of information in many fields, including transportation, where cities can use cameras to identify dangerous intersections, and health sciences, where images from various sources such as magnetic resonance imaging (MRI) or X-rays are used to predict diseases. In this report, we focus on the latter field; more precisely, we discuss the diagnosis of skin diseases using photographs of moles.

As cameras went digital, pixels became the atom of photos and images. This means that the most common approach to store an image is as a matrix, where an element (i, j) represents the color at pixel coordinates $[i, j]$. Unsurprisingly, most approaches for image analysis directly use this pixel representation, and modern models inspired by convolutional neural networks (CNNs; LeCun et al., 1989) have shown great prowess in learning complex patterns within images. However, as data dimensions increase in both quantity and resolution, these models can be computationally intensive to train. Additionally, pixel-based approaches are often difficult to interpret and are vulnerable to changes in resolution.

Hence, in this paper, we study the benefits of using a different representation for images in the context of supervised learning on real data. Shape-based image representation Moindjié, Beaulac, and Descary, 2025; Moindjié, Descary, and Beaulac, 2025 focuses on the geometric features of objects of interest within images. By capturing the contours of these objects and representing them using a Fourier basis expansion, we can dramatically reduce the dimensionality of images, thereby lowering the computational burden of the analysis. Moreover, this representation enhances interpretability by replacing pixel-level information with shape-related inference. Finally, once projected onto the same basis, images of different resolutions become directly comparable, effectively addressing the generalization issues of CNNs across varying image resolutions.

In the following research report, we study a popular benchmark dataset for image analysis as we compare pixel-based and shape-based approaches. We discuss computational cost and classification results.

In Section 2, we thoroughly introduce the dataset of interest as well as the preprocessing steps required for the subsequent analysis. Section 3 describes the nonlinear classification models designed for this comparison, as well as the tuning and hyperparameter selection process. Finally, results are discussed in Section 4, before concluding in Section 5.

2 Dataset

This paper focuses on *HAM10000 dataset (Human Against Machine)* (Tschandl et al., 2018) for the numerical experiments. The *HAM10000* dataset is a popular benchmark for medical image analysis (Guan et al., 2024). Its popularity can be explained by its high number of samples (10015), the diversity of skin lesions (7 types), and the quality of the labels, obtained via expert consensus.

We focus on the two most represented classes over the seven skin lesions: *benign* and *melanoma*. By restricting *HAM10000* to these classes, the sample number decreases to 7826 images, in which 1115 are *melanoma* and 6711 are *benign*:

- As its name indicates, the first one (*benign*) represents non-pathological moles. They appear in various cases and are generally symmetric.
- The second class (*melanoma*) represents a malignant mole, which is a type of skin cancer. Contrary to *benign* moles, they do not generally have a specific form.

Since these moles are used for cancer diagnostic-making, a classification algorithm that can distinguish these two classes is of great interest (see e.g. Das et al., 2021, Ali et al., 2022, Mehr and Ameri, 2022). However, this algorithm must be trained in a highly imbalanced class situation, as in real conditions, *benign* are overrepresented compared to *melanoma*. In the learning part, this class's imbalanced situation is considered by using a weighted loss (see Section 3.3, for details).

2.1 Preprocessing

Contrary to pixel-based methods, which can consider color variations, shape-based methods can only account for objects' forms in the image. Hence, to fairly compare these two approaches, we focus on the *silhouette* of the moles as explanatory variables.

Then, the inputs of the two approaches are derived from the segmented images, which encode the silhouette of the mole (see Figure 1). This section presents the operations aimed at obtaining the segmented images from the original ones.

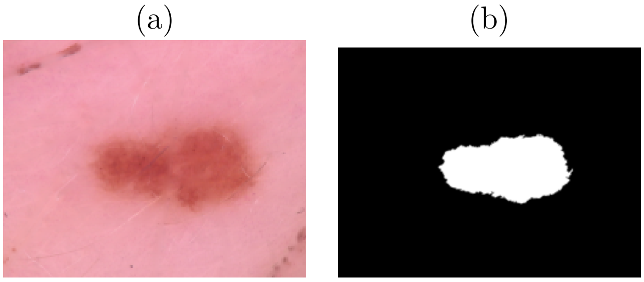


Figure 1: From the colored image (a) to the segmented image (b).

Since *HAM10000* is a multi-source dataset, we have to define pre-processing steps that are robust to the diversity of the images, relative to their initial publication. In addition to the diversity of the sources, there are also intrinsic challenges that are obstacles to decent segmentation of moles. Figure 2 presents some of them.

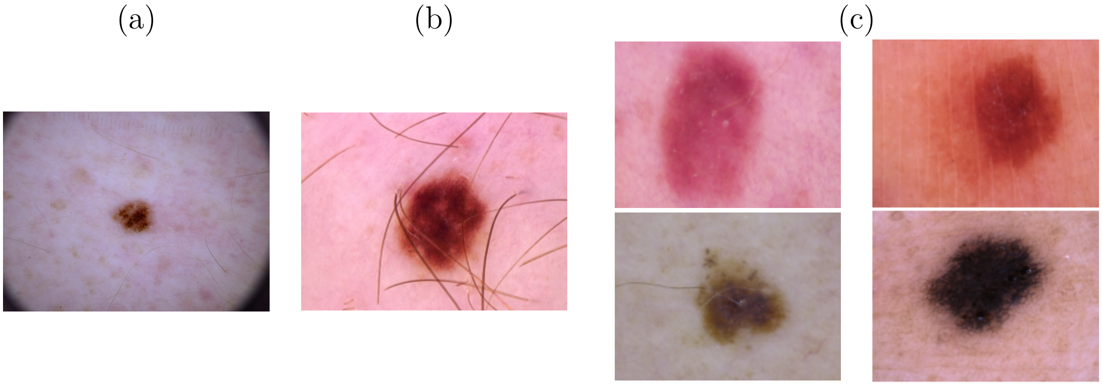


Figure 2: Challenges to the segmentation: (a): Circular frame from the lens used for the dermoscopy, (b): Presence of hairs, and (c): Variability in the tints of the moles

The *Circular frame* problem (Figure 2 a) is addressed using the elliptic region of interest (ROI) methodology, and, to address the remaining challenges, we propose a procedure relying on the following steps: hair removal and channel choice for segmentation. The next sections briefly present them. For additional details, we also provide their implementation on our open-access GitHub repository[1].

2.1.1 Hair removal

In Figure 2 (b), the presence of hairs is presented as a major nuisance. This, because hairs can mimic the edges of the lesion, leading to poor border detection of the mole's contour. To remove hairs from images, we use the methodology proposed in Gencer, 2025, which relies on the *top hat* mathematical morphological operation and an in-painting method. Briefly, the top hat transform is used to extract the hairs in the image, which are then removed. Then, in-painting extrapolates the value of missing pixels.

[1] https://github.com/Advla/Internship_Shape-based_pixel-based_DeepL_approach-to-mole-classification

Specifically, four main steps compose the hair removal pipeline (see Figure 3): (1) apply a top-hat transform on the red channel of the image, (2) apply a Gaussian blur to the resulting image, (3) apply the Otsu thresholding method (Otsu, 1979) to detect and remove hairs,(4) use internal in-painting to fill the resulting gaps (Telea, 2004).

Figure 3 presents the pipeline on a *HAM10000* image. Moreover, it is worth noting that this method has a low computational cost as it relies on basic and computationally efficient operations.

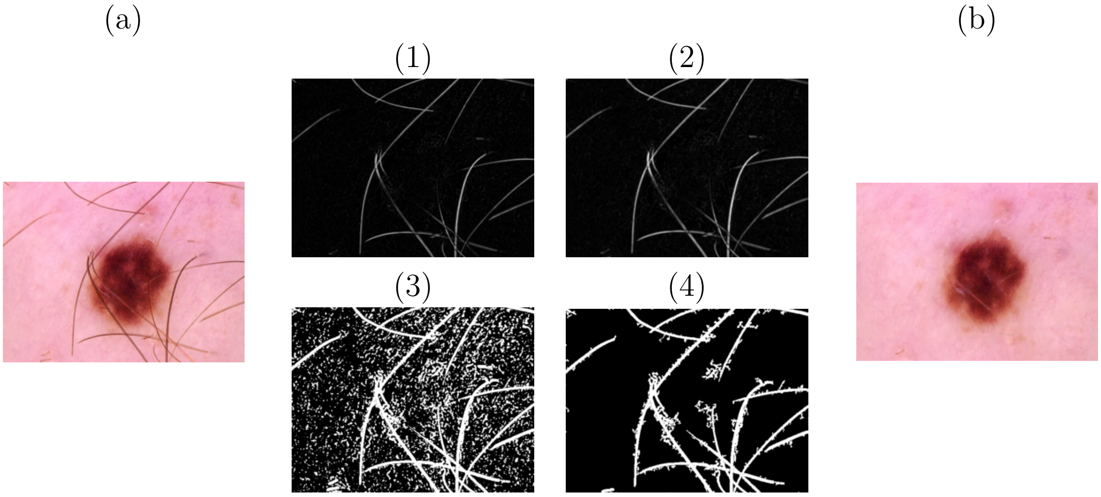


Figure 3: Illustration of the four hair removal steps (Gencer, 2025) on the original image (a): the resulting image is (b).

2.1.2 Channel choice for segmentation

Segmentation aims to obtain a binarized image (or mask) where a pixel is set to one if it belongs to the mole and zero otherwise (see Figure 1). A classic approach is to define a threshold α or automatically determine one (Otsu, 1979): if a pixel's gray-scale value exceeds α, we replace it with 1, and if not, we replace it with 0.

As seen in Figure 2 (c), the data exhibits a unique variability of tints and consequently using a known linear combinations of the Red, Green, and Blue (RGB) channels to create a gray-scale representation could be suboptimal. As an alternative, we propose a method that learns the coefficient of the linear combination of RGB channels to create a unique gray-scale representation specific to this problem. We use Principal Component Analysis (PCA) on the RGB channels, and we choose the first principal component such that this new channel captures the maximum variance for each image. Figure 4 presents an illustration of the segmentation pipeline.

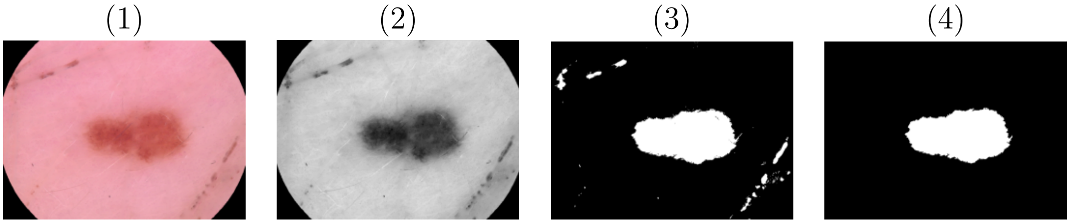


Figure 4: Segmentation pipeline: (1) image after the hair removal step, (2) the "PCA" channel with min-max normalization, (3) segmented images using Otsu's threshold, (4) segmented images after artifacts cleaning.

It also shows that some additional operations are needed to clean the mask of remaining artifacts (from step (3) to (4)), such as area calculus and additional morphological mathematical operations. For the sake of conciseness, these remaining details are provided on our open-access GitHub repository [2].

2.2 Contour extraction

The input of the pixel-based methods is the mask obtained in the previous step. For shape-based methods, they take as input the contour of the main object in the image, which also necessitates the use of masks; the marching Squares algorithm (Maple, 2003) is performed on the masks to obtain the contour. The next sections provide more details on the models used for comparing the two paradigms.

3 Models

This part presents the chosen methods for evaluating the ability of each framework (pixel and shape) to classify the moles. Deep-learning methods are used as they demonstrate remarkable progress in image classification tasks.

- For the pixel-based paradigm, we use a convolutional neural network method (CNN). It learns features automatically from the image, using a kernel optimization of a feedforward neural network (LeCun et al., 1989).
- For the shape-based paradigm, we use a multilayer perception method (MLP): a feedforward neural network of fully connected neurons with nonlinear activation functions (see Popescu et al., 2009 for details).

To ensure a fair comparison with the shape-based MLP, we designed the CNN with a complexity order comparable to the MLP. The goal is to assess the quality of the information contained in the pixel representation versus the shape representation, without the bias introduced by massive, pre-trained deep learning architectures.

[2] https://github.com/Advla/Internship_Shape-based_pixel-based_DeepL_approach-to-mole-classification

3.1 Pixel-based: CNN for classification

The convolutional neural network automatically extracts features from the pixel representation of images, where the features aim to be meaningful descriptors for classification. Then, the inputs of the CNN are the full-size segmentation masks produced by the previous steps. Figure 5 presents the global procedure.

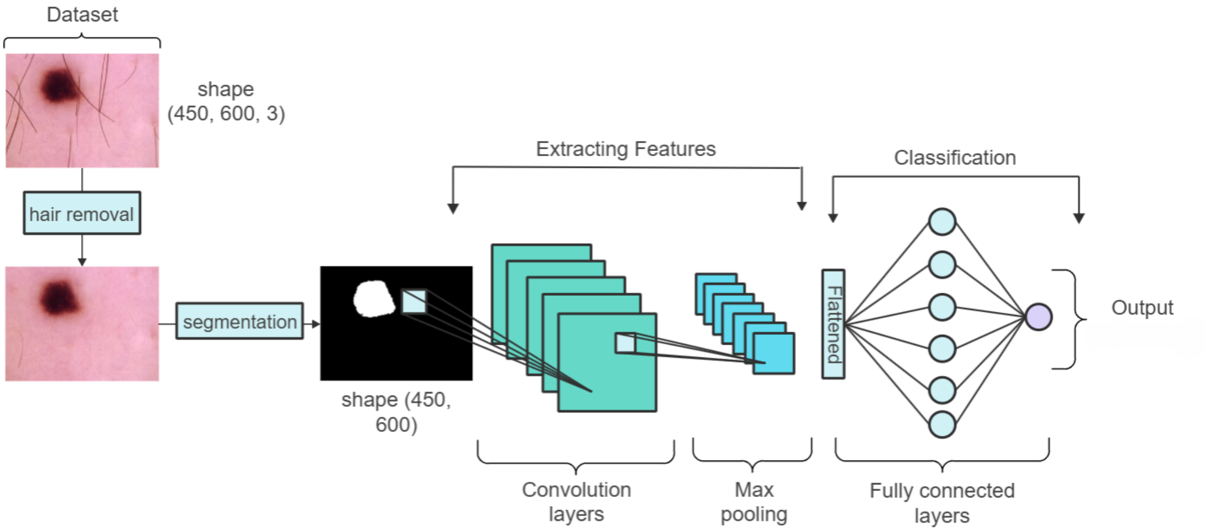


Figure 5: Complete data analysis pipeline, including preprocessing, for the pixel-based approach.

3.2 Shape-based: MLP for classification

In the shape-based paradigm, we use Fourier coefficients to approximate the contour's shapes. To remove translation, rotation, and re-parametrization deformations from the detected contours, the work of Moindjié, Beaulac, and Descary, 2025 is performed as a preprocessing step. The obtained coefficients are descriptors of the *shape* of the contours. These coefficients are the shape features in MLPs. The deformation estimates, such as translation and rotations are also added to the MLP model as features. Figure 6 illustrates the global MLP pipeline, and Appendix A provides details on the input formats.

3.3 Selected spaces of hyperparameters

For each method, we perform qualitative tests followed by a Bayesian Search (with 100 iterations), to define appropriate hyperparameter spaces for each architecture (CNN and MLP). These preliminary results help us select the hyperparameter space, where the main characteristics are the following:

- *L_2- Regularization:* 10^{-4}–10^{-2} (log scale)
- *Dropout rates:* 0.1–0.4 (log-scale)
- *Learning rates:* 5×10^{-4}–2×10^{-3} (log scale)

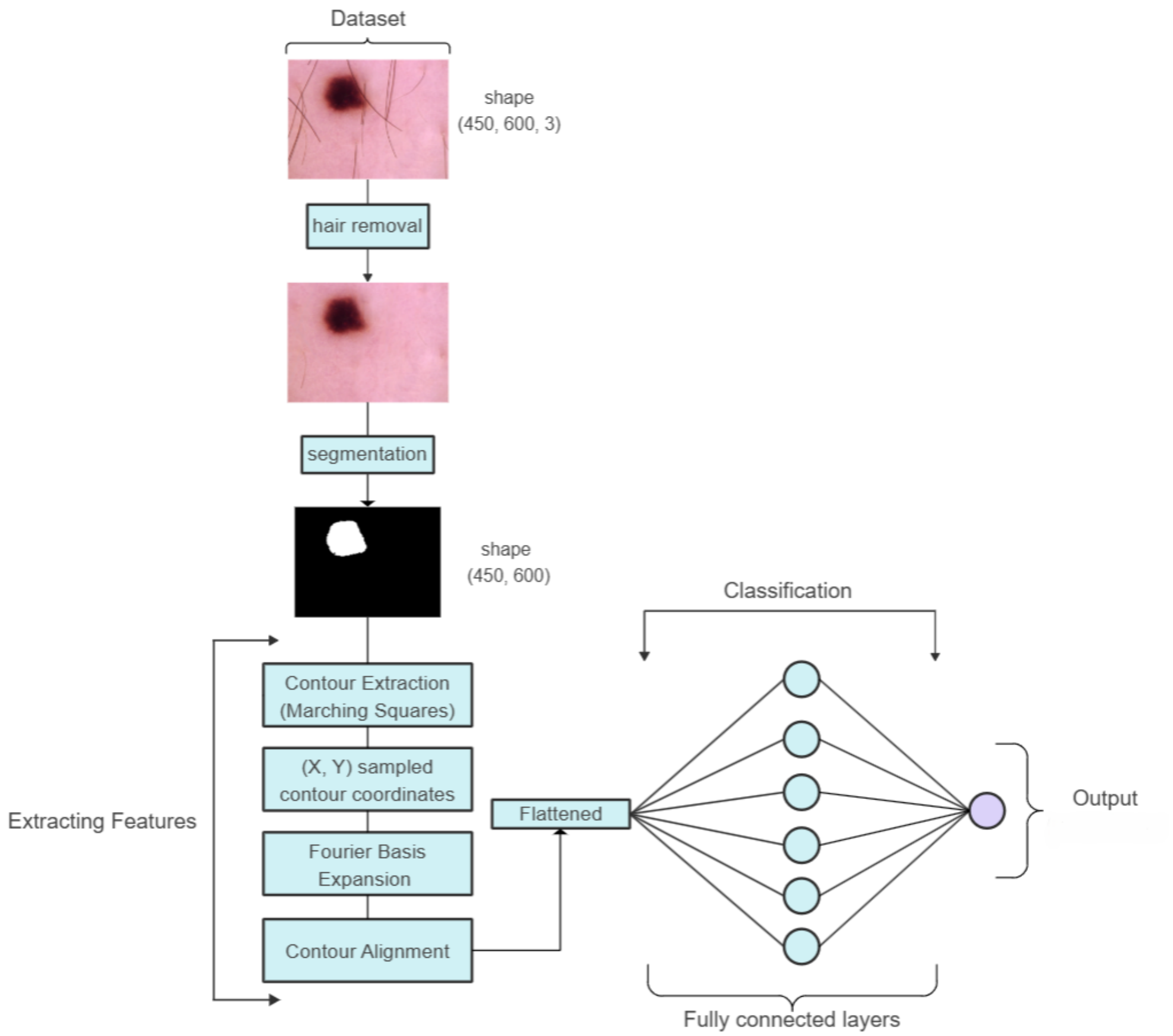


Figure 6: Complete data analysis pipeline, including preprocessing, for the shape-based approach.

- *Depth:*

MLP	*Dense depth:*	1–3 layers with units $\{16, 32, 64, 128\}$
CNN	*Dense depth:*	1–3 layers with units $\{32, 64, 128\}$
	Convolutional depth:	2–4 layers with filter sizes $\{16, 32, 64\}$, kernel size = (3, 3)

In both models (MLP and CNN), we use ReLU activations, He-normal initialization, and batch normalizations. The estimation of the models is performed by *Adam* (Kingma and Ba, 2014) on a weighted binary cross-entropy loss and a batch size of 32, using Area Under the Curve (AUC) as the evaluation metric.

The training was conducted on *Google Colab* using an NVIDIA A100 GPU with 80 GB of VRAM. The complete process, including both training and evaluation, required a total runtime of 17 hours and 19 minutes.

The MLP architecture had a modest memory footprint of 0.7 GB, whereas the CNN consumed as much as 66.7 GB of VRAM, despite both models being trained with a batch size of 32. Full-resolution images were used for the experiments (450 × 600).

3.4 Tuning procedure and evaluation metrics

We evaluated our feature extractors using a nested stratified cross-validation with $m = 20$ outer folds (See Figure 7). The principle is quite simple, we devided the data set into a training set and a test set $m = 20$ different times and each time we performed cross-validation on the trainings set to tune the model. For each of the $m = 20$ outer test fold, the remaining folds were used for hyperparameter tuning via an inner cross-validation with $l = 10$ randomly sampled configurations from a reduced search space.

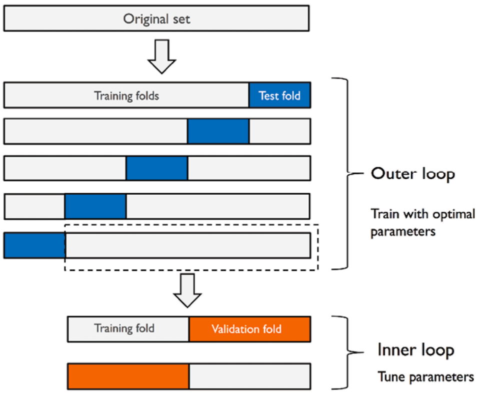


Figure 7: Nested Cross-Validation Chart.

The best model according to the inner cross-validation was then retrained on the entire training set and evaluated on the held-out test set. The resulting test metrics across $m = 20$ folds are presented in the next section.

4 Results

Table 1 presents a summary of the obtained results, and Figure 8 shows the resulting AUCs. We also present the result of the *naive* model, which predicts the majority class (*benign*) for all instances. If a model achieves a lesser performance than the naive one, then it is less efficient than a random classification procedure. As these are standard metrics, we left their definition to the appendix.

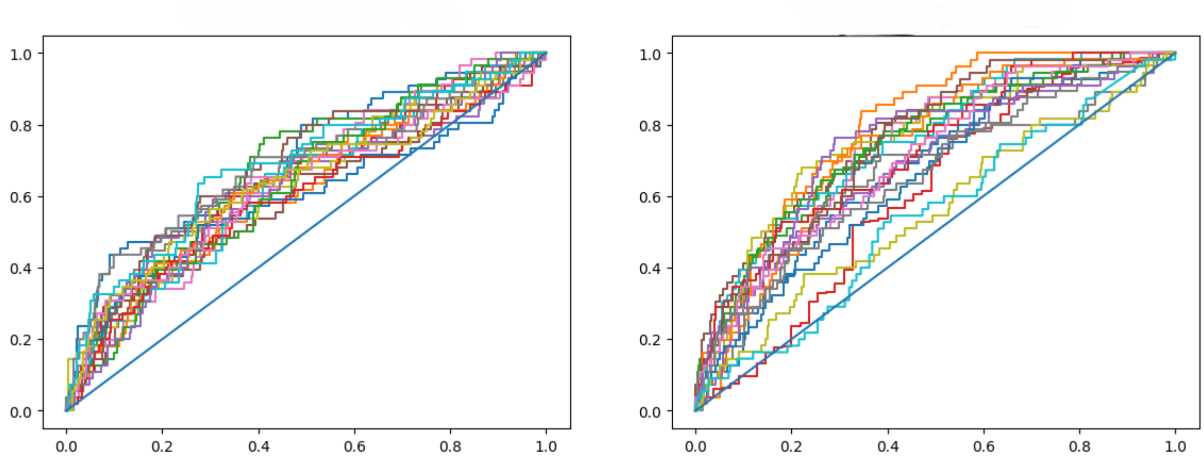
The obtained results show that MLP and CNN are both more efficient than the naive model. They also show that the pixel-based method (CNN) achieves, on average, a slightly higher performance than MLP in terms of AUC. However, compared to MLP, CNN has a much higher AUC standard deviation, which makes it less stable in terms of prediction quality. Moreover, in terms of computational cost, MLPs are much more competitive than CNNs, as **they have nearly** 5.5 **times less training time** (including data standardization) than CNNs, **while using almost** 95 **times less VRAM** (0.7GB for MLPs vs. 66.7GB for CNNs).

4.1 Statistical significance of the results

Since the results are quite close for the two models, we perform some tests to assess the statistical significance of the performance gaps. Specifically, we construct confidence intervals for the difference metrics (AUC, Balanced Accuracy, F1) across all folds using bootstrapping (Efron, 1982).

Model	AUC	Bal.Acc.	F1	Train time (s)	VRAM (GB)
MLP	0.640 ± 0.031	0.572 ± 0.036	0.837 ± 0.020	31.7 ± 6.92	0.7
CNN	0.697 ± 0.069	0.544 ± 0.049	0.814 ± 0.016	174 ± 82.9	66.7
Naive	0.5	0.5	0.791	0	0

Table 1: Stratified-Nested-CV results overview: $m \pm \sigma$, with m denotes the mean and σ the standard deviation



(a) ROC Curves per fold - MLP

(b) ROC Curves per fold - CNN

Figure 8: Comparison of ROC curves per fold for MLP and CNN architectures.

For estimating the empirical distributions of the difference vectors, we bootstrap them 10000 times, which allows us to determine the 2.5% and 97.5% quantiles of each metric. Differences for which the interval did not include 0 were considered statistically significant, whereas intervals containing 0 indicated no significant difference. Table 2 presents the results for each metric.

Metric	95% CI (Bootstrap)	Decision
AUC	$(-0.092, -0.017)$	Significant$^{(*)}$
Balanced Accuracy	$(-0.003, +0.057)$	Not significant
F1 Score	$(-0.003, +0.057)$	Not significant

Table 2: Statistical comparison of MLP vs CNN using 95% bootstrap confidence intervals. $(*)$: *CNN performs significantly better than MLP.*

The obtained results show that the observed difference in terms of AUC is significant, with CNN outperforming the MLP. For the remaining metrics (Balanced Accuracy and F1 Score), the confidence intervals include zero, meaning that there is no evidence of a significant performance difference between the two models for these metrics.

5 Conclusion

We presented a numerical comparison of discrete and continuous representations of images in a supervised classification context. While the discrete paradigm based on pixel representations of the image data has been largely studied (LeCun et al., 1989), the presented continuous representation is a novel methodology (Moindjié, Beaulac, and Descary, 2025); it relies on statistical shape analysis (Dryden and Mardia, 2016) and functional data analysis (Ramsay and Silverman, 2005) to account for the continuous nature of the main object in the images. This work aimed to compare these two approaches in a real data application context.

To do so, we used the *HAM10000* datasets (Tschandl et al., 2018): a popular medical image dataset of diverse skin lesions. Specifically, the focus was on the two most represented classes: *benign* (non-pathological) and *melanoma* (pathological) moles. Using neural network algorithms for classification, CNN for pixels and MLP for continuous shapes, our study demonstrated that the continuous approach gives overall competitive results compared to pixel-based methodologies, while needing fewer computational resources; the computational time was divided by 5, and the memory footprint ratio was of about 95: MLP needed 0.7 GB and CNN 66.7 GB for the training. The parsimonious representation of continuous-based methods explains these findings: it needs a negligible number of Fourier coefficients (202) compared to the number of pixels (450×600).

In this report, we focused on the silhouette of the moles and, therefore, the influence of color variations was ignored. Our motivation was to provide a fair comparison between the two approaches, since the continuous approach exclusively relies on the object's silhouette in the images. Future works will investigate the integration of the colors in the shape-based approach, as colors in images may be strongly informative in the classification task (see e.g., Gowda and Yuan, 2018, Funt and Zhu, 2018). This will enable additional numerical experiments to study the limitations and the benefits of the two paradigms on various supervised and unsupervised problems.

Acknowledgements

The authors would like to acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada and STATQAM. The authors would like to thank Mélanie Raymond for her insightfull suggestions.

References

Ali, K., Shaikh, Z. A., Khan, A. A., & Laghari, A. A. (2022). Multiclass skin cancer classification using efficientnets–a first step towards preventing skin cancer. *Neuroscience Informatics*, *2*(4), 100034.

Das, K., Cockerell, C. J., Patil, A., Pietkiewicz, P., Giulini, M., Grabbe, S., & Goldust, M. (2021). Machine learning and its application in skin cancer. *International Journal of Environmental Research and Public Health*, *18*(24), 13409.

Dryden, I. L., & Mardia, K. V. (2016). *Statistical shape analysis: With applications in r*. John Wiley & Sons.

Efron, B. (1982). *Bootstrap methods: Another look at the jackknife*. Springer.

Funt, B., & Zhu, L. (2018). Does colour really matter? evaluation via object classification. *Color and Imaging Conference*, *26*, 268–271.

Gencer, K. (2025). Enhanced skin lesion classification through hair artifact removal and transfer learning models. *Black Sea Journal of Engineering and Science*, *8*, 1007–1021. https://doi.org/10.34248/bsengineering.1665621

Gowda, S. N., & Yuan, C. (2018). Colornet: Investigating the importance of color spaces for image classification. *Asian conference on computer vision*, 581–596.

Guan, H., Yap, P.-T., Bozoki, A., & Liu, M. (2024). Federated learning for medical image analysis: A survey. *Pattern recognition*, *151*, 110424.

Keras Team. (2025). *Classification metrics (auc)* [Accessed: 2025-09-15]. https://keras.io/api/metrics/classification_metrics/

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551. https://doi.org/10.1162/neco.1989.1.4.541

Maple, C. (2003). Geometric design and space planning using the marching squares and marching cube algorithms. *2003 international conference on geometric modeling and graphics, 2003. Proceedings*, 90–95.

Mehr, R. A., & Ameri, A. (2022). Skin cancer detection based on deep learning. *Journal of biomedical physics & engineering*, *12*(6), 559.

Moindjié, I.-A., Beaulac, C., & Descary, M.-H. (2025). A functional approach for curve alignment and shape analysis. *arXiv preprint arXiv:2503.05632*.

Moindjié, I.-A., Descary, M.-H., & Beaulac, C. (2025). Statistical analysis of multivariate planar curves and applications to x-ray classification. *arXiv preprint arXiv:2508.11780*.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Popescu, M.-C., Balas, V. E., Perescu-Popescu, L., & Mastorakis, N. (2009). Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, *8*(7), 579–588.

Ramsay, J. O., & Silverman, B. W. (2005). *Functional data analysis* (2nd). Springer.

Telea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, *9*(1), 25–36. https://doi.org/10.1080/10867651.2004.10487596

Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, *5*(1), 1–9.

A The shape inputs in MLP

With the border extracted, we rely on the work of Moindjié, Beaulac, and Descary, 2025 to extract the following information of the contours:

- $\rho \in \mathbb{R}^+$: scaling factor
- $T \in \mathbb{R}^2$: translation vector
- $\theta \in [0, 2\pi]$: rotation angle
- $\delta \in [0, 1]$: starting point of the parametrization function.

In their approach, they consider the contours as a planar curve $X : [0,1] \to \mathbb{R}^2$ which relates a latent shape $\tilde{X}$ by

$$X(t) = \rho \, O_\theta \, \tilde{X} \circ \gamma_\delta(t) + T \quad t \in [0,1]$$

where $t \in [0,1]$, O_θ is the rotation matrix of angle θ and $\gamma_\delta(t) = \text{mod}(t - \delta, 1)$, where mod is the modulo function.

For obtaining parameters θ and δ, a template μ is used. The estimation of these parameters relies on the following optimization problem:

$$(\hat{\theta}, \hat{\delta}) = \arg\min_{\theta,\delta} \; \|X - O_\theta \mu\|_f^2$$

where $\|\cdot\|_f$ is an adequate functional norm.

A.0.1 Fourier Representation and Numerical Optimization

To make the minimization tractable, both X^* and μ are projected on a truncated Fourier basis of dimension M:

$$X^*(t) = \sum_{k=1}^{M} \alpha_k \psi_k(t), \qquad \mu(t) = \sum_{k=1}^{M} u_k \psi_k(t) \quad \alpha_k, u_k \in \mathbb{R}^2$$

The coefficients are arranged into matrices $\alpha, u \in \mathbb{R}^{M \times 2}$.

The reparametrization γ_δ acts through a block-diagonal orthogonal matrix $\beta(\delta)$, while the rotation O_θ acts globally. The alignment criterion becomes

$$(\hat{\theta}, \hat{\delta}) = \arg\min_{\theta,\delta} \; \|\alpha - O_\theta u \beta(\delta)\|_F^2$$

where $\|\cdot\|_F$ is the matrix Frobenius norm:

$$\|A\|_F^2 = \sum_{i,j} A_{ij}^2$$

Algorithm

1. Grid search over $\delta \in [0, 1]$ (e.g. step size 0.01).
2. For each δ, solve for the optimal rotation $\hat{\theta}_\delta$.
3. Select $(\hat{\theta}, \hat{\delta})$ minimizing the Frobenius error.

A.0.2 Pipeline used and Features retained

The contours extracted from the images are not the shape variables themselves, as they are latent variables that we need to estimate. For doing so, we apply the approach proposed in Moindjié, Beaulac, and Descary, 2025. It aims to align every contour on an arbitrary reference: a mole with a well-defined boundary.

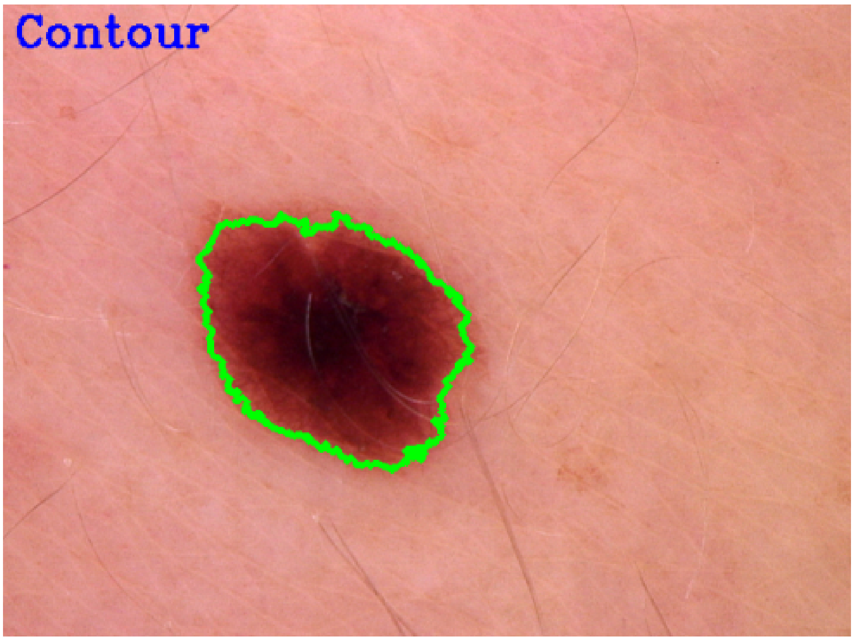


Figure 9: Reference we aligned other contours on - ISIC_0024803

For this purpose, we need to set a number M of Fourier functions for the functional approximation of the two coordinates of the contour (see Appendix A for details). Here, it is set to $\mathbf{M} = \mathbf{101}$ Fourier functions, where the first one is a constant function.

The features in the MLP are the following :

- $\mathbf{2M} = \mathbf{202}$ aligned Fourier coefficients (float):, which are the 101 coefficients for $x-$coordinate of the contours and 101 coefficients of $y-$coordinate of the contours.
- ρ (float) scaling factor: which are the norm of centred contour.
- θ (float): rotation angle (in radians) estimated to align the contour with the reference.
- δ (float): the reparametrization coefficient, which defines the aligned starting point of the contour with the reference.
- $\min_{\theta,\delta} \|\alpha - O_\theta u \beta(\delta)\|_F^2$ (float): which is the Frobenius distance between the aligned contour and the reference.

B Performance metrics

Let us briefly defined the metrics used in Section 4. Given the following notation:

- True positive := $\mathbf{TP}$,
- True negative := $\mathbf{TN}$,
- False positive := $\mathbf{FP}$,
- False negative := $\mathbf{FN}$,
- True Positive Rate (Recall) := $\mathbf{TPR} = \frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FN}}$,
- True Negative Rate (Specificity) := $\mathbf{TNR} = \frac{\mathbf{TN}}{\mathbf{FP}+\mathbf{TN}}$,
- False Positive Rate := $\mathbf{FPR} = \frac{\mathbf{FP}}{\mathbf{FP}+\mathbf{TN}}$,

we define the following metrics:

- Area Under Curve (AUC) := $\int_0^1 \mathbf{TPR}(\mathbf{FPR})d(\mathbf{FPR})$, computed using the trapezoidal rule (following Keras implementation ; Keras Team, 2025),
- Balanced accuracy := $\frac{1}{2}\left(\frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FN}} + \frac{\mathbf{TN}}{\mathbf{TN}+\mathbf{FP}}\right)$,
- F1 := $\frac{2\mathbf{TP}}{2\mathbf{TP}+\mathbf{FP}+\mathbf{FN}}$.