Hairy Ground Truth Enhancement for Semantic Segmentation

Sophie Fischer   Irina Voiculescu
sophie.fischer@cs.ox.ac.uk   irina@cs.ox.ac.uk
Oxford University Department of Computer Science

Abstract

Semantic segmentation is a key task within applications of machine learning for medical imaging, requiring large amounts of medical scans annotated by clinicians. The high cost of data annotation means that models need to make the most of all available ground truth masks; yet many models consider two false positive (or false negative) pixel predictions as 'equally wrong' regardless of the individual pixels' relative position to the ground truth mask. These methods also have no sense of whether a pixel is solitary or belongs to a contiguous group.

We propose the Hairy transform, a novel method to enhance ground truths using 3D 'hairs' to represent each pixel's position relative to objects in the ground truth. We illustrate its effectiveness using a mainstream model and loss function on a commonly used cardiac MRI dataset, as well as a set of synthetic data constructed to highlight the effect of the method during training. The overall improvement in segmentation results comes at the small cost of a one-off pre-processing step, and can easily be integrated into any standard machine learning model. Rather than looking to make minute improvements for mostly correct 'standard' masks we instead show how this method helps improve robustness against catastrophic failures for edge cases.

1. Introduction

Semantic segmentation (the process of assigning each pixel in an image with a binary label) has a wide range of applications within medical imaging, including creating 3D reconstructions [4], real-time ultrasound analysis [10], and tumour classification [8]. Due to its complexity, semantic segmentation requires significant computing resources and large amounts of labelled data; something which is limited due to the high demand on clinician time [11]. These challenges mean that every step of semantic segmentation needs to be closely scrutinised, to see where improvements can be made without additional data or other resources.

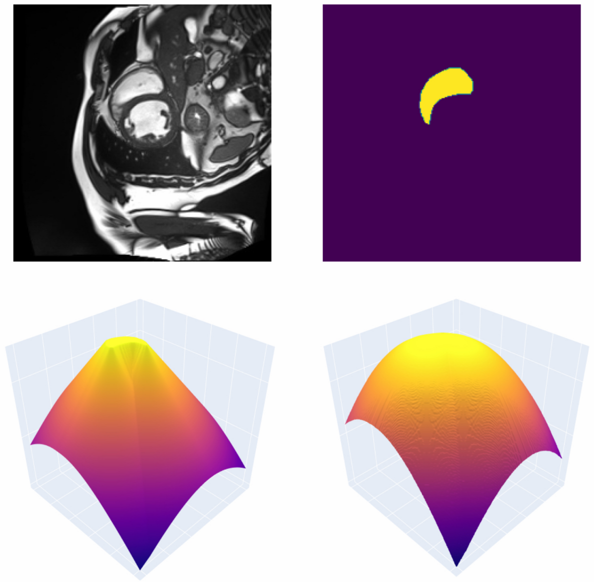


Figure 1. **Top Left:** An example cardiac MRI. **Top Right:** The corresponding segmentation for the right ventricle. **Bottom Left:** A 2.5D visualisation of a standard distance transform applied to the segmentation. **Bottom Right:** A 2.5D visualisation of our proposed 'Hairy' transform applied to the segmentation.

A unique advantage of medical imaging datasets is that throughout a given dataset the biological feature being segmented (such as a particular organ, bone or network of blood vessels) not only remains the same, but many of its geometric and topological properties do as well. An example of this can be seen in Fig. 1, where the right ventricle in a cardiac MRI is always a genus 0 (single) shape, and often slightly curved and elliptical.

Our contribution provides a method of enhancing ground truth images prior to training by treating the mask as a vector field (or 'hairy' surface), which can then be combined with the predicted pixel values. By considering the sum of these vectors we can not only establish a heuristic for clustering, but also for where the clustering is with respect to the mask. An even distribution of pixels around the mask results in the hairs effectively cancelling each other out, whereas large clusters (especially those further away from the mask) will remain.

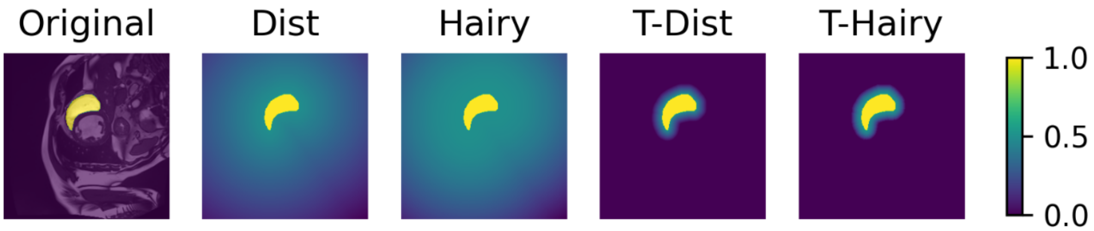


Figure 2. Examples of the resulting values from four transforms on data from the ACDC dataset (Sec. 5.2) with the right ventricle highlighted. **Left to Right:** The original ground truth (overlaying the scan), a Distance Transform, a Hairy Transform, a Truncated Distance Transform, and a Truncated Hairy Transform.

2. Background

Loss Functions

Loss functions are used to assign a real number to each machine segmentation against the corresponding ground truth. This value represents how well the algorithm has modelled the dataset at a given stage of training, and is therefore used to refine future learning. As the loss function is applied against every machine segmentation it must be chosen carefully based on the task and data used. Within semantic segmentation, the ways in which loss functions consider predictions can be grouped into four categories: distribution based, region based, boundary based, and compounded [13].

The first two of the categories, distribution-based and region based, contain many of the most popular loss functions for semantic segmentation, including Cross-Entropy Loss, Dice Loss, and Tversky Loss (Sec. 5). In these functions, the location of the pixel relative to the mask is irrelevant to the result of the function (aside from whether or not the pixel is in the mask). This has the benefit of being quicker to calculate, but potentially hinders the learning of the model.

Image Processing

Image processing is routinely used in semantic segmentation for multiple reasons. Basic transforms of the individual raw images (such as rotations and reflections) get used for generic dataset augmentation, whereas techniques such as contrast or smoothing filters get used to improve the suitability of a raw image. Processing techniques are particularly important for applications in medical imaging, where they can help address the issue of small datasets and low quality images respectively.

Whilst many processing techniques are aimed at the original raw data, in this work we focus on processing techniques applied to segmentation masks. These techniques can be used to directly affect the results of loss functions; regardless of the choice of function, the resulting value is generally dependent on the ground truth and its corresponding machine segmentation, rather than the original image.

When considering image pre-processing for the inputs to a loss function, we can choose to apply the processing to either the ground truth or the machine segmentation (or both). Each comes with its own advantages and disadvantages:

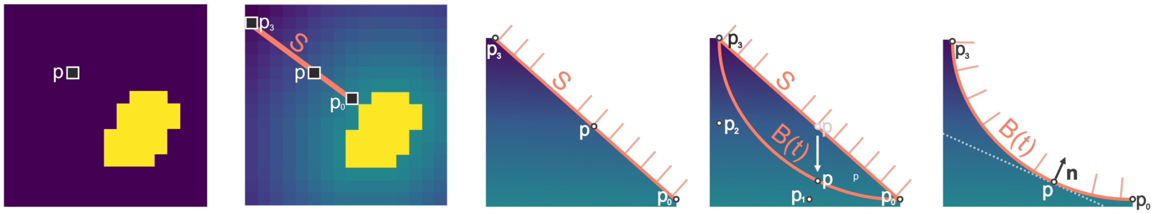
- **Ground Truth** Processing only needs to be applied once, but cannot provide insight into a specific machine segmentation [9, 18].
- **Machine Segmentation** Processing must be reapplied for each segmentation, but provides insight specific to each segmentation [21].

For our study we transform the ground truth images, a procedure which has only a small impact on the training time on the model. Uniquely however, the method we propose is still able to provide additional insight into the distribution of pixels in the machine segmentation.

Existing Methods

Distance transforms (Fig 2) are an existing method of transforming a binary mask: each pixel is assigned a value relating to the distance away from a ground truth object. Truncated distance transforms were used by Audebert et al. to help improve accuracy on the boundary of predictions [1]. Another use of distance transforms was within the deep learning architecture `ResUNet` [6]. This model achieved semantic segmentation of aerial images by using a multi-task learning algorithm, which trained a network simultaneously on multiple instances of the ground truth, including distance transform.

Another family of processing methods are **polar coordinate transforms**, which convert the image from Cartesian to polar coordinates (or spherical coordinates in 3D). This is normally applied to both the scan and the segmentation, using a given point of origin (such as the centre of the image). One theory is that these coordinates can be beneficial to segmenting elliptical features, such as tumours or lesions, or in scenarios when the original area is spherical such as retinal imaging [2, 16, 20].



(a) (b) (c) (d)

Figure 3. Fitting a Bézier curve within the Hairy Transform: **a)** The ground truth, considering pixel p **b)** Identifying the closest mask boundary pixel p_0 and corresponding image boundary pixel p_3, such that p, p_0, and p_3 are collinear on line segment S **c)** Fitting control points p_1 and p_2 to determine the curve $B(t)$ **d)** Calculating the new value for p, as well as the curve normal n_p

3. Method

We compare two methods of transforming a ground truth prior to use in a machine learning model:

- **Distance Transform.** The conventional distance transform f_{d_2} results in a value representing the shortest distance from the pixel to the mask boundary.
- **Hairy Transform.** The novel Hairy transform f_h incorporates aspects of distance transforms, whilst also adding a heuristic for position relative to the mask.

This section describes both f_{d_2} and f_h in detail. For both methods we also consider truncated versions of the transforms, referred to as $f_{t\text{-}d_2}$ for the truncated distance transform and $f_{t\text{-}h}$ for the truncated Hairy transform.

Key Definitions

In this work, we consider the segmentation **image** P as a complete 2D set of pixels with height $h{+}1$ and width $w{+}1$. Each pixel p has co-ordinates $p_x{\in}[0,...,w]$ and $p_y{\in}[0,...,h]$ as well as value $p_v{\in}[0,1]$. The **mask** C is then a subset of P, such that:

$$C = \{\, p \in P \mid p_v = 1 \,\} \tag{1}$$

The function $\texttt{nbrhd}(p)$ returns the set of adjacent pixels in P, including those diagonal to p. The **boundary** of P is defined as:

$$P_b = \{\, p \mid p_x = 0 \vee p_x = w \vee p_y = 0 \vee p_y = h \,\} \tag{2}$$

The **boundary** of C is defined as

$$C_b = \{\, p \notin C \mid \exists\, p_i \in \texttt{nbhd}(p) \text{ s.t. } p_i \in C,\ p \notin P_b \} \tag{3}$$

i.e. for all $p \in C_b$ where p does not lie on the boundary of P, there exists some adjacent pixel to p in C.

Truncation

Within a distance transform, we can introduce a maximum scalar p_{MAX}. This truncation value is a function of C, where for a given mask C:

$$p_{MAX} = \frac{\sqrt{|C|}}{\texttt{aspect}(C)} \tag{4}$$

This has the effect of localising the transform within a set distance around the mask, and is frequently used alongside distance transforms in semantic segmentation tasks [5, 12].

The Distance Transform

For two pixels p and q, the result of a standard distance transform is defined by a chosen distance metric. In this work we solely use the Euclidean distance d_2:

$$d_2(p,q) = \sqrt{\,(p_x - q_x)^2 + (p_y - q_y)^2} \tag{5}$$

For an image P with corresponding mask C, the distance transform $f_{d_2}(P)$ affects only the values p_v for $p \in P$. Each pixel p is considered independently:

$$f_{d_2}(p) = \left\{ \begin{array}{ll} 0 & \text{for } p \in C \\ \textbf{min}(\, d_2(p,b) \mid b \in C_b \,) & \text{for } p \notin C \end{array} \right\} \tag{6}$$

The transformed value of p_v is then:

$$p_v = f_{d_2}(p) \tag{7}$$

For the truncated version of f_{d_2} we simply have:

$$f_{t\text{-}d_2}(p) = \textbf{min}(\, f_{d_2}(p),\, p_{MAX} \,) \tag{8}$$

The Hairy Transform

The high-level concept of the Hairy transform is to assign each pixel in the ground truth a vector (or 'hair'), transforming the mask into a vector field. By replacing pixel values with multi-dimensional vectors we can embed information about the pixel position with regards to the relevant mask. The Hairy transform can be broken down into three stages:

1. A distance transform
2. A transform to a Bézier curve
3. The calculation of surface normals

Distance Transform

This is identical to the transform applied earlier, with truncation applied here if being used.

Bézier Transform

We now define a function `bezier` that takes a point p and transformed image $f(P)$ as input, and returns a three dimensional vector.

For our given point p, we have three associated values: p_x, p_y and p_v. We can consider p_v as a equivalent z coordinate, transforming P into a $h \times y$ field of three dimensional points.
We now wish to construct a three dimensional Bézier curve intersecting p. A single Bézier curve of degree n requires $n+1$ control points, where the first and last control points are interpolated. We use a cubic Bézier as it gives control of the gradient at both ends of the curve, without adding significant complexity. This then requires four control points to be assigned.

In our transform we want to only interpolate points in P_b and C_b. This sets a convex hull around these points, meaning that all Bézier curves constructed lie 'under' this hull. In instances where C is concave or disconnected we also look to interpolate local maxima, however for now we just consider the simpler cases.

Control Points. For each point $p \notin P_b, C_b$, we begin by assigning the first and last control points p_0 and p_3. The first point p_0 can be considered as the closest point to p on the boundary C_b:

$$p_0 \leftarrow \{\, p_0 \in C_b \;\mid\; d(p_0,p) < d(q,p) \quad \forall q \in C_b \,\} \tag{9}$$

Assuming $C \neq \emptyset$ and C is not concave, a single closest point must exist. The other interpolated point p_3 is then chosen as a point from P_b such that p_0, p, p_3 are collinear. It must also be closer to p than p_0, such that the line segment S from p_0 to p_3 does not intersect C:

$$\begin{aligned} p_3 \leftarrow \{\, p_3 \in P_b \;\mid d(p_0,p) + d(p,p_3) = d(p_0,p_3) \\ \wedge \;\; d(p_3,p) < d(q,p) \quad \forall q \in P_b \,\} \end{aligned} \tag{10}$$

An illustration of this can be seen in Fig. 3 parts **a** and **b**.

With the interpolation points assigned, we then look to define the other two control points p_1 and p_2. These lie on the plane that both contains S and is also perpendicular to the xy-plane:

$$p_1 \leftarrow \begin{bmatrix} p_{0_x} + \frac{|p_{3_x} - p_{0_x}|}{2} \\ p_{0_y} + \frac{|p_{3_y} - p_{0_y}|}{2} \\ p_{0_z} \end{bmatrix} \tag{11}$$

The point p_1 lies on the xy-plane, with x and y values halfway between p_0 and p_3. The point p_2 lies directly underneath p_3, halfway between the z values of p_0 and p_3:

$$p_2 \leftarrow \begin{bmatrix} p_{3_x} \\ p_{3_y} \\ p_{0_z} + \frac{|p_{3_z} - p_{0_z}|}{2} \end{bmatrix} \tag{12}$$

With the control points established, we can then define the Bézier curve as a function of a parameter $t \in [0,1]$:

$$B(t) = (1-t)^3 p_0 + 3(1-t)^2 t p_1 + 3(1-t)t^2 p_2 + t^3 p_3 \tag{13}$$

This is then used to calculate the transformed value of p, where t is relative to the placement of p on S:

$$p \leftarrow B(\frac{|p_0 - p|}{|S|}) \tag{14}$$

The selection of p_1 and p_2, as well as the resulting Bézier curve, can be seen in Fig. 3 **c**.

As all control points lie in a plane perpendicular to the xy-plane, the transformed values of p_x and p_y are unaffected; only p_v is changed. In the simple version of the Hairy transform, this transformed value of p_v is sufficient as a form of distance heuristic to the mask. However, by considering the normal to the curve $B(t)$ at p we can find further uses for the transform.

Curve Normals. To calculate the normal to $B(t)$ for a value $t \in [0,1]$ we can use the cross product of the derivative $B'(t)$ and the normal to the plane defined by the control points p_0, p_1 and p_3. For the former, we have:

$$B'(t) = 3(1-t)^2(p_1 - p_0) + 6t(1-t)(p_2 - p_1) + 3t^2(p_3 - p_2)$$

For the latter, the three points are not collinear as long as $p_0 \neq p_3$, meaning we can calculate the plane normal $\mathbf{n}_{plane}$ using the control points:

$$\mathbf{n}_{plane} = (p_1 - p_0) \times (p_1 - p_3) \tag{15}$$

From this the curve normal $\mathbf{n}_{B_{(t)}}$ for a given value t is:

$$\mathbf{n}_{B_{(t)}} = B'(t) \times \mathbf{n}_{plane} \tag{16}$$

Example curve normals can be seen in Fig. 3 **d**.

As each point p lies on a single curve B for a unique value of t, we can refer to $\mathbf{n}_{B_{(t)}}$ more simply as $\mathbf{n}_p$, i.e. the normal at point p. This value can then be returned as the output of `bezier`, giving us the Hairy transform for a point p:

$$f_h(p) = \left\{ \begin{array}{ll} [0,0,1]^\top & \text{for } p \in C \\ \texttt{bezier}(p, f_{d_2}(P)) & \text{for } p \notin C \end{array} \right\} \tag{17}$$

For the truncated version of f_h we have:

$$f_{t\text{-}h}(p) = \left\{ \begin{array}{ll} [0,0,1]^\top & \text{for } p \in C \\ \texttt{bezier}(p, f_{t\text{-}d_2}(P)) & \text{for } p \notin C \end{array} \right\} \tag{18}$$

Use of Curve Normals

The previous steps allow us to use f_h to convert an image P into an array of normal vectors $P_{\mathbf{n}}$ with shape $[h, w, 3]$. For a given point p on P we can use the corresponding normal $\mathbf{n}_p$ as an efficient heuristic for two pieces of information.

The first is the distance from the nearest mask pixel. This can be calculated as:

$$d_c = 1 - \mathbf{n}_p \cdot [0, 0, 1]^\top \tag{19}$$

or more simply, the z-coordinate of $\mathbf{n}_p$.

The second piece of information is the *divergence score* for a (non-activated) segmentation, which can be easily generated at the same time as the loss function. For an image P, a machine segmentation S can be considered an array of shape $[h, w]$, with each value $s_{ij} \in [0, 1]$ (as this is before application of an activation function). To calculate the divergence score $\texttt{div}(P, S)$ we first need to get the relevant non-mask vectors:

$$\mathbf{n} = \sum (f_h(P - C) \cdot S\,) \tag{20}$$

We then normalise the sum of these vectors, then take the length of the x and y components:

$$\texttt{div}(P, S) = |\,\hat{\mathbf{n}}|_{xy} \tag{21}$$

Integration with Cross Entropy Loss

In standard Cross Entropy (CE) loss, the loss for a pixel p is calculated as the sum of the loss across all classes:

$$CE(p) = \sum_{c \,\in\, classes} -y_c \log(\hat{y_c}) \tag{22}$$

When the ground truth is transformed with a transform f, the new CE loss becomes:

$$CE_f(p) = \sum_{c \,\in\, classes} -y_c \log(\hat{y_c} \cdot 2f(p, P)) \tag{23}$$

As the transform is only applied to the ground truth, the value of $f_(p, P)$ only needs to be calculated once ahead of training. The scalar multiplier is used to account for the original normalised values $f(p, P) \in [0, 1]$.

4. Datasets

The experiments make use of: synthetic data constructed specially for illustration purposes, as well as a cardiac MRI dataset.

4.1. Synthetic Data

The synthetic data is a set of 10 ground truths, each 16 by 16 pixels. Each pixel may only have the value `0` or `1`. For each image there are 5 artificial machine segmentations, each with exactly 10 false positive pixels (there are no false negative pixels).

The use of the synthetic data is to illustrate on a more manageable scale how the use of ground truth augmentations would change the resulting CE loss.

4.2. ACDC

The ACDC dataset is a publicly available set of cine-MRI cardiac images, taken for the purpose of measuring variations in the diastolic volume and ejection fraction of the heart [3]. In this dataset we consider the segmentation tasks of identifying the left and right ventricles as independent tasks. In each case we only used images from the dataset containing these features, resulting in 1,772 images for the left ventricle and 1,540 images for the right ventricle.

5. Experiments

5.1. Synthetic Data

The results for the first three synthetic images can be seen in Fig. 4, showing the range in resulting CE values as well as the `div` values. As expected, not using a ground truth transform resulted in an identical maximum CE loss for all machine segmentations, and when all false positive pixels were outside of the truncation radius the T-Dist and T-Hairy transforms would result in the same maximum CE loss value. We can also see that when an transform is used the CE Loss is near zero for the first segmentation, which suggests that the transformed CE loss may need to be adjusted to stop it from being too low in these cases. Generally the truncated transforms resulted in the greatest spread of values, with T-Hairy slightly above T-Dist.

The `div` values for both T-Hairy and Hairy transforms can be seen above each segmentation in Fig. 4, referred to as T-Div and Div respectively. Much as expected, T-Div $>$ Div; also, T-Div reaches its maximum value once all pixels lie outside the truncation distance. The distance from the mask appears to have significant weight over the clustering of the pixels, for example when considering the fourth and fifth segmentations of Image 2 in Fig. 4.

5.2. ACDC

Experimental Setup

A U-Net was trained for 80 epochs for each task, using an 80/20 split of train/test images. The U-Net architecture was designed specifically for biomedical imaging segmentation [15], and it continues to perform highly across medical imaging modalities [17].

For each model CE loss was used, enhanced with one of the five types of ground truth transform as described in the previous section:

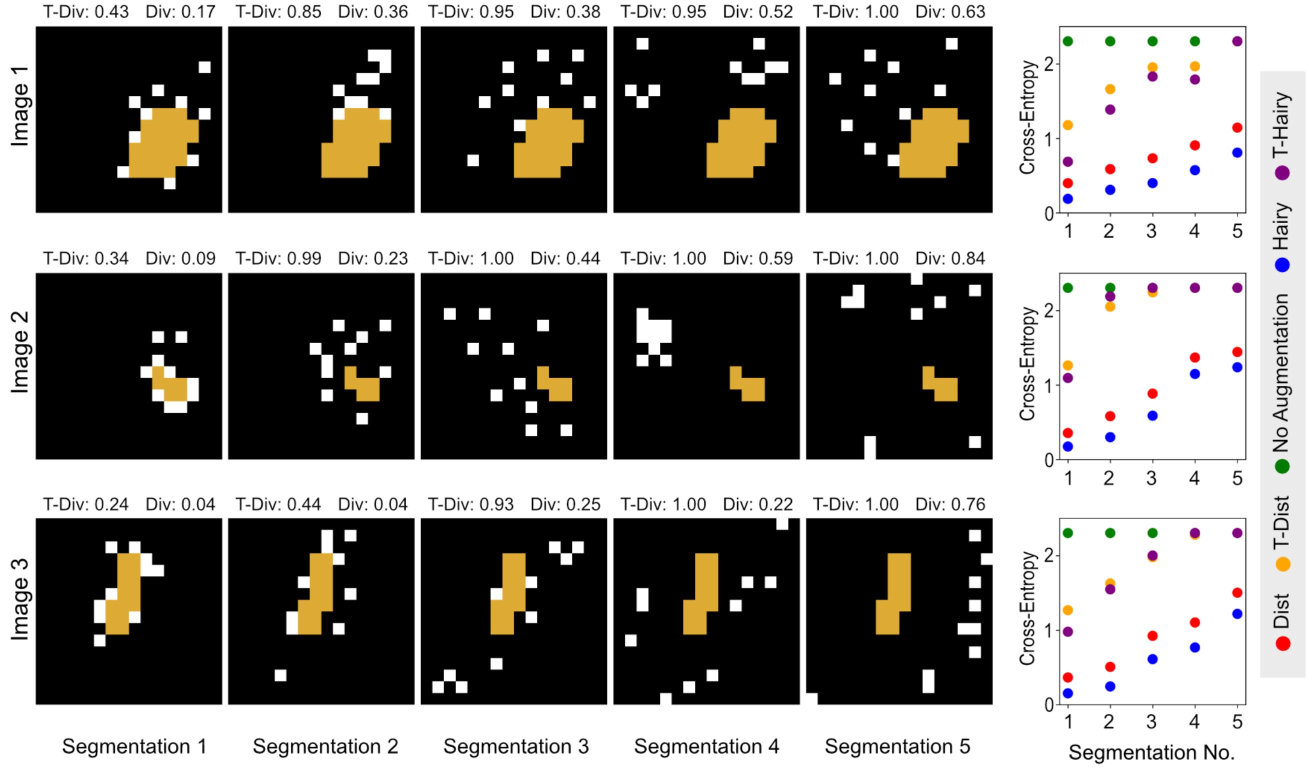


Figure 4. Three of the synthetic ground truths, with each with 10 false positive pixels (in white). The right column shows a comparison of the CE values for each transform and each set of false positive pixels.

- No transform (**None**)
- Distance transform (**Dist**)
- Distance transform with truncation (**T-Dist**)
- Hairy transform (**Hairy**)
- Hairy transform with truncation (**T-Hairy**)

The predicted masks were evaluated with a range of measures, including:

- **Dice** - the Sørensen-Dice similarity coefficient [7]
- **SBM** - the Symmetric Boundary Match percentage [19] with a radius of 5 pixels
- **RVD** - the Relative Volume Difference between the ground truth and machine segmentation
- **Miss%** - the percentage of the number of masks in the test set in which the predicted mask entirely missed the ground truth
- **FNVF/FPVF** - the False Negative Volume Fraction and False Positive Volume Fraction [14], measures of the number of false negatives and positives respectively.

Training was repeated five times for each model, with the test and training sets shuffled randomly each time. The results presented below are the mean metric values across the test sets.As all transforms were applied to the ground truths prior to training, there was no significant increase in training time when using transformed ground truths. Training was implemented with PyTorch 2.0.0, with a 6.0GB NVIDIA GeForce RTX 3060 GPU.

Results

Tables 1 and 2 list the mean and standard deviation values for the performance metrics used. For most metrics the difference between models was slight, with the notable exception being the Miss%. Across both the left and right ventricles the standard deviation values were generally consistent across the models , with the highest deviation in the RVD and Miss% values.

Left Ventricle For the left ventricle there was little to no change seen in the first three metrics, although there was a reduction in the average number of masks missed when using an transform (in particular the T-Hairy transform).

Right Ventricle For the right ventricle there was slightly more variance across all of the metrics used, both between models and across each test set. Generally the transformed models performed worse that the non-transformed model,

Mean Values LV						
Aug	Dice	RVD	SBM	FNVF	FPVF	Miss%
None	0.909 ± 0.13	**0.116** ± 0.30	0.916 ± 0.12	**0.084** ± 0.13	0.00053 ± 0.0005	0.90 ± 0.75
Dist	0.909 ± 0.13	0.121 ± 0.33	0.917 ± 0.12	0.088 ± 0.13	**0.00050** ± 0.0005	0.68 ± 0.47
T-Dist	**0.910** ± 0.12	0.118 ± 0.32	**0.918** ± 0.12	0.085 ± 0.13	0.00052 ± 0.0005	0.79 ± 0.38
Hairy	0.909 ± 0.12	0.120 ± 0.32	0.917 ± 0.12	0.085 ± 0.12	0.00051 ± 0.0005	0.79 ± 0.75
T-Hairy	0.909 ± 0.12	0.118 ± 0.31	0.916 ± 0.12	0.086 ± 0.13	0.00051 ± 0.0005	**0.51** ± 0.39

Table 1. Mean and Standard Deviation values for the Left Ventricle, with the optimal result for each metric in bold.

Mean Values RV						
Aug	Dice	RVD	SBM	FNVF	FPVF	Miss%
None	0.841 ± 0.20	0.194 ± 0.68	0.853 ± 0.19	0.162 ± 0.20	0.00096 ± 0.0013	2.01 ± 1.15
Dist	0.840 ± 0.20	**0.189** ± 0.61	0.852 ± 0.19	0.169 ± 0.21	**0.00090** ± 0.0013	1.95 ± 0.92
T-Dist	0.838 ± 0.20	0.196 ± 0.64	0.850 ± 0.19	0.167 ± 0.20	0.00096 ± 0.0013	**1.82**± 1.02
Hairy	0.840 ± 0.19	0.196 ± 0.62	0.852 ± 0.19	0.166 ± 0.20	0.00093± 0.0012	1.88 ± 1.15
T-Hairy	**0.842** ± 0.20	0.193 ± 0.66	**0.854** ± 0.19	**0.160** ± 0.20	0.00098 ± 0.0013	2.01 ± 1.13

Table 2. Mean and Standard Deviation Values for the Right Ventricle, with the optimal result for each metric in bold.

except for T-Hairy which performed marginally better with the right ventricle on multiple metrics (Dice, SBM, and FNVF).

When considering the difference between the masks for the left and right ventricle, an significant factor is the irregularity of the shape. In this work we chose the consider the *Aspect Ratio* as a measure of irregularity, for which a value close to 1 indicates a shape with similar height and width (such as a circle), whereas a higher value indicates a more elongated, less regular shape. In Fig. 5 the distribution of Aspect Ratio values for the left and right ventricle masks is shown, highlighting significantly more variance in the right ventricle. Examples of a right ventricle with aspect ratio 4.1 and 13 can be seen in Fig. 6.

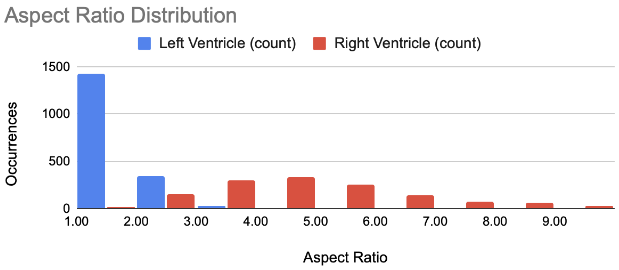


Figure 5. The distribution of Aspect Ratio values for the left and right ventricles within the ACDC dataset. Values close to 1 indicate a more regular shape.

To see how the transforms performed on more irregular masks, we compared the same performance metrics again on the right ventricle test set, but this time using the upper quartile of the aspect ratio distribution. The change in values for the Dice and Miss% can be seen in Fig. 7.

A clear take-away from these results is that in almost all cases, models perform worse with the more irregular masks. When considering the other metrics used for comparison the change in performance was similar to that of the Dice values; a drop in performance roughly consistent across the models. Overall the non-transformed and T-Hairy models saw the least change, indicating that they were not only the best performing models overall but also the most robust.

Most effective, however, is the change in Miss% – unlike every other model, the T-Hairy model actually saw a significant decrease in the percentage of entirely missed masks. These results indicate that using the T-Hairy transform may lead to a better performance with more irregular segmentations, resulting in a higher accuracy and less masks missed.

6. Conclusion

As of writing, despite the growing need for techniques that do not increase the complexity of the model, ground truth transforms are still an under-utilised method in deep learning. This work shows how ground truth enhancement can improve the performance of segmentation methods. Such results should encourage the investigation of these techniques as as a way to increase model performance efficiently.

The proposed T-Hairy ground truth transform is particularly effective when applied to segmentation tasks for more irregular shapes. The careful analysis based on bespoke synthetic data suggests that this may be because it only ef-

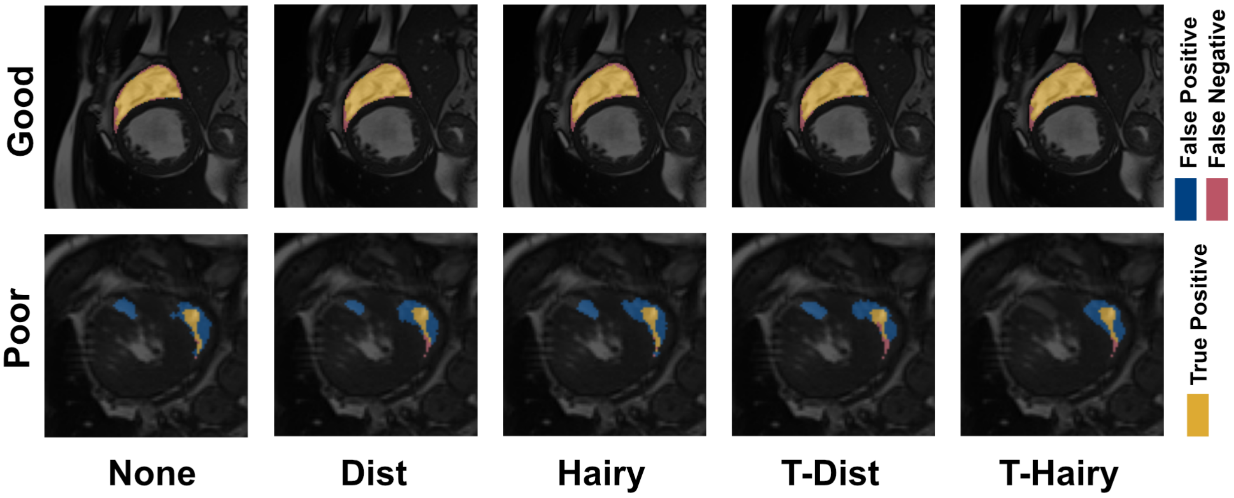


Figure 6. Segmentation examples from two different scans of the right ventricle. **Top:** A case where all transforms resulted in an accurate segmentation (Aspect Ratio: 4.1) . **Bottom:** A case where no transform resulted in a satisfactory segmentation, in particular with large amounts of false negative pixels (Aspect Ratio: 13.0). Even though the T-Hairy model misses the mask entirely, the predicted mask is the closest both geometrically and topologically. This illustrates the specific benefit of the T-Hairy model.

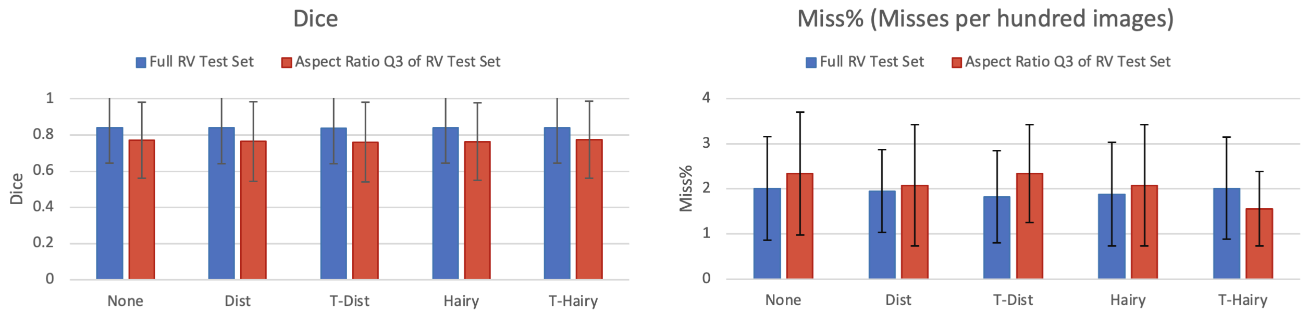


Figure 7. Bar charts comparing the mean Dice and Miss% values for the right ventricle test set, when considering full test set (blue) or the upper quartile of the aspect ratio distribution within the test data.

fects the CE Loss within the truncation radius, allowing for a broader range of reasonable values within this distance. Whilst current improvement is slight, there is a notable benefit in the number of machine segmentations where an object is missed entirely.

Use of the curve normals also appears to provide additional insight into the distribution of a segmentation prior to activation, in particular highlighting clustering outside of the target object. Utilising these normals may still require further refinement however, as the values are highly dependent on the distance of individual pixels from the mask.

7. Further Work

The next areas to be developed regarding the transform techniques include the use of signed transforms for all four transform techniques described, as well as introducing a lower limit for the CE values. The Hairy and T-Hairy techniques should also be tested with the `div` values incorporated, potentially using hyperparameters to control the weighting of the different elements.

When considering the data used, subsequent experiments should look to incorporate additional datasets, ideally ones containing topologically complex components such as holes or concavities. Subsequent experiments could also use other medical imaging modalities, or apply the transform in a 3D setting.

References

[1] Nicolas Audebert, Alexandre Boulch, Bertrand Le Saux, and Sébastien Lefèvre. Distance transform regression for spatially-aware deep semantic segmentation. *Computer Vision and Image Understanding*, 189:102809, 2019. 2

[2] Marin Benčević, Irena Galić, Marija Habijan, and Danilo Babin. Training on polar image transformations improves

biomedical image segmentation. *IEEE access*, 9:133365–133375, 2021. 2

[3] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, et al. Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: is the problem solved? *IEEE transactions on medical imaging*, 37 (11):2514–2525, 2018. 5

[4] Shekhar S Chandra, Ying Xia, Craig Engstrom, Stuart Crozier, Raphael Schwarz, and Jurgen Fripp. Focused shape models for hip joint segmentation in 3d magnetic resonance images. *Medical image analysis*, 18(3):567–578, 2014. 1

[5] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018. 3

[6] Foivos I Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:94–114, 2020. 2

[7] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 6

[8] Evgin Goceri. Capsnet topology to classify tumours from brain images and comparative evaluation. *IET Image Processing*, 14(5):882–889, 2020. 1

[9] Christina Gsaxner, Birgit Pfarrkirchner, Lydia Lindner, Antonio Pepe, Peter M Roth, Jan Egger, and Jürgen Wallner. Pet-train: Automatic ground truth generation from pet acquisitions for urinary bladder segmentation in ct images using deep learning. In *2018 11th Biomedical Engineering International Conference (BMEiCON)*, pages 1–5. IEEE, 2018. 2

[10] Julian Guerrero, Septimiu E Salcudean, James A McEwen, Bassam A Masri, and Savvakis Nicolaou. Real-time vessel segmentation and tracking for ultrasound imaging applications. *IEEE transactions on medical imaging*, 26(8):1079–1090, 2007. 1

[11] Yanming Guo, Yu Liu, Theodoros Georgiou, and Michael S Lew. A review of semantic segmentation using deep neural networks. *International journal of multimedia information retrieval*, 7:87–93, 2018. 1

[12] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Boundary-aware instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5696–5704, 2017. 3

[13] Shruti Jadon. A survey of loss functions for semantic segmentation. In *2020 IEEE conference on computational intelligence in bioinformatics and computational biology (CIBCB)*, pages 1–7. IEEE, 2020. 2

[14] Estimated False Negatives and True Positives. The false-negative fraction for papanicolaou smears. *Arch Pathol Lab Med*, 121:270–272, 1997. 6

[15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 5

[16] Carlo Russo, Sidong Liu, and Antonio Di Ieva. Spherical coordinates transformation pre-processing in deep convolution neural networks for brain tumor segmentation in mri. *Medical & Biological Engineering & Computing*, 60:121–134, 2022. 2

[17] Nahian Siddique, Sidike Paheding, Colin P Elkin, and Vijay Devabhaktuni. U-net and its variants for medical image segmentation: A review of theory and applications. *Ieee Access*, 9:82031–82057, 2021. 5

[18] Yuan Xue, Hui Tang, Zhi Qiao, Guanzhong Gong, Yong Yin, Zhen Qian, Chao Huang, Wei Fan, and Xiaolei Huang. Shape-aware organ segmentation by predicting signed distance maps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12565–12572, 2020. 2

[19] Varduhi Yeghiazaryan and Irina Voiculescu. Family of boundary overlap metrics for the evaluation of medical image segmentation. *Journal of Medical Imaging*, 5(1): 015006–015006, 2018. 6

[20] Muhammad Nauman Zahoor and Muhammad Moazam Fraz. Fast optic disc segmentation in retina using polar transform. *IEEE Access*, 5:12293–12300, 2017. 2

[21] Ling Zhang, Xiaosong Wang, Dong Yang, Thomas Sanford, Stephanie Harmon, Baris Turkbey, Bradford J Wood, Holger Roth, Andriy Myronenko, Daguang Xu, et al. Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation. *IEEE transactions on medical imaging*, 39(7):2531–2540, 2020. 2