MICAD2022
Medical Imaging and Computer-Aided Diagnosis

MICCAI
MICCAI Society Endorsed Event

Organ Segmentation: A Journey from Level Sets to Shape Denoising

Adrian Barbu
Professor, Statistics Department
Florida State University

Acknowledgements: Orhan Akal, Cheng Long

Organised by: University of Leicester

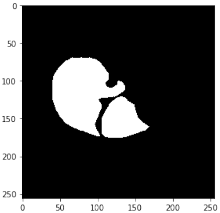
Supported by: Spectrum Academy, Springer

Outline

Introduction

Chan-Vese, CV:

- A level set method that minimizes a Mumford-Shah integral
- Simultaneously evolves a level set surface and fits locally constant intensity models for the interior and exterior regions
- Its length-based contour regularization is quite simple and too weak for many applications
- Performs poorly compared with the state of the art object segmentation methods



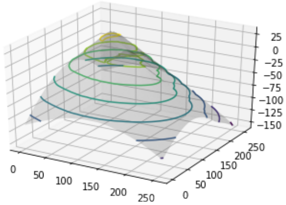


Figure: Signed distance transform of a given object. 0-level set of distance transform gives the object boundary $C = \{(x, y) | \varphi(x, y) = 0\}$

Chan-Vese Overview

The Chan-Vese Active contour [Chan and Vese, 2001] is aimed at minimizing the following Mumford-Shah energy

$$E(C) = \int_{C_i} (I(u) - \mu_i)^2 du + \int_{C_o} (I(u) - \mu_o)^2 du + \nu |C| \tag{1}$$

where

- I denotes the image intensity,
- C is the curve to be fitted,
- C_i, C_o are the regions inside and respectively outside the curve C,
- μ_i and μ_o are the intensity averages of image I inside and respectively outside the curve C.

Mumford-Shah to Chan-Vese

A direct approach to minimize the energy would be to derive the Euler-Lagrange equation and obtain a curve evolution equation:

$$\frac{\partial C}{\partial t} = f(\kappa)\vec{N} \tag{2}$$

where

- κ is the curvature of C,
- f is some function of the curvature,
- $\vec{N}$ is the normal vector to the curve.

Such a direct approach is difficult to implement because of topological changes:

- self intersection of the curve
- splitting into sub-objects, developing holes, or multiple objects merging into one

These make managing the curve representation quite challenging.

Chan-Vese

- The curve C, is represented as the 0 level set of a surface φ, $C = \{(x, y) | \varphi(x, y) = 0\}$.
- Usually $\varphi(x, y)$ is initialized as the signed Euclidean distance transform of C
- The energy (1) is extended to an energy of the level set function φ:

$$
\begin{aligned}
E(\varphi, I) = \int (I(u) - \mu_o(\varphi(u)))^2 (1 - H_\epsilon(\varphi(u)))\, du + \\
\int (I(u) - \mu_i(\varphi(u)))^2 H_\epsilon(\varphi(u)) du + \nu \int \delta_\epsilon(\varphi(u)) |\nabla \varphi(u)| du
\end{aligned}
\tag{3}
$$

where

- H_ϵ is a smoothed Heaviside function and δ_ϵ is its derivative.
- The parameter ν controls the curve length regularization $\int |\nabla \varphi|$

Chan-Vese: Heaviside

- The following smoothed Heaviside function could be used

$$H_\epsilon(z) = \begin{cases} 0 & \text{if } z < -\epsilon \\ 1 & \text{if } z > \epsilon \\ \frac{1}{2}[1 + \frac{z}{\epsilon} + \frac{1}{\pi}\sin(\frac{\pi z}{\epsilon})] & \text{if } |z| < \epsilon \end{cases} \tag{4}$$

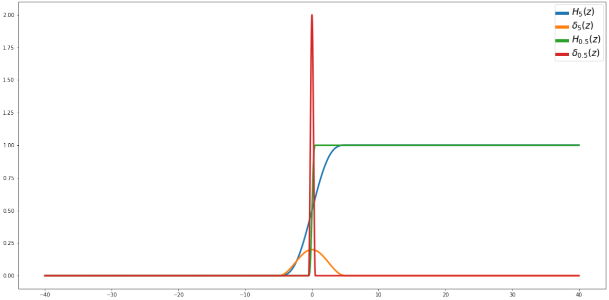


Figure: Heaviside $H_\epsilon(z)$ and it's derivative dirac delta function $\delta_\epsilon(z)$ for $\epsilon = 5$ and $\epsilon = 0.5$

Chan-Vese: Euler-Lagrange Equation

The energy is minimized alternatively by updating μ_i, μ_o

$$\begin{aligned} \mu_i^t &= \frac{\int I(u) H_\epsilon(\varphi^t(u)) du}{\int H_\epsilon(\varphi^t(u)) du}, \\ \mu_o^t &= \frac{\int I(u)[1 - H_\epsilon(\varphi^t(u))] du}{\int [1 - H_\epsilon(\varphi^t(u))] du}, \end{aligned} \tag{5}$$

then assuming μ_i^t, μ_o^t fixed the solution φ needs to satisfy the Euler-Lagrange equation:

$$\delta_\epsilon(\varphi)[\nu(\frac{\nabla \varphi}{|\nabla \varphi|}) + (I - \mu_o^t)^2 - (I - \mu_i^t)^2] = 0 \tag{6}$$

which can be done iteratively:

$$\varphi^{t+1} = \varphi^t + \eta[\kappa(\varphi^t) + (I - \mu_o^t)^2 - (I - \mu_i^t)^2] \tag{7}$$

CVNN: Chan-Vese CNN

We generalize the Chan-Vese by replacing the curvature term $\kappa(\varphi) = \frac{\nabla\varphi}{|\nabla\varphi|}$ in the Euler-Lagrange equation (7) with a generic shape function $g(\varphi, \beta)$ with parameters β. Obtain the iterative algorithm

$$\varphi^{t+1} = \varphi^t + \eta\delta_\epsilon(\varphi^t)(g(\varphi^t, \beta) + (I - \mu_o)^2 - (I - \mu_i)^2) \quad (8)$$

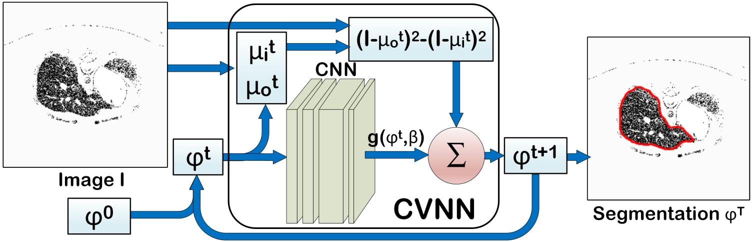


Figure: Our RNN model merging CNN with Chan-Vese

Proposed method

The problem that we are trying to address is

1. To impose better shape priors in the Chan-Vese formulation
2. To learn these shape priors using training examples instead of setting them by hand to a predefined form.

Our formulation uses

1. A Convolutional Neural Network (CNN) to encode the shape prior
2. The whole Chan-Vese evolution becomes a Recurrent Neural Network (RNN)
3. Training the shape model is done in an end-to-end fashion by backpropagation.

Loss Function

We are going to use the Combo loss [Taghanaki et al., 2019]

$$
\begin{aligned}
L(\boldsymbol{\beta}) = & \alpha_1 \left(-\frac{1}{N} \sum_{i=1}^{N} \alpha_2 Y_i \ln \hat{R}_i - (1-\alpha_2)(1-Y_i) \ln(1-\hat{R}_i) \right) \\
& - (1-\alpha_1) \frac{\sum_{i=1}^{N} Y_i \hat{R}_i + s}{\sum_{i=1}^{N} Y_i + \sum_{i=1}^{N} \hat{R}_i + s}
\end{aligned} \tag{9}
$$

where s is a small positive smoothing factor, $\boldsymbol{\beta}$ are the U-Net weights, $\hat{R}_i \in [0,1]$ is the prediction for voxel i after sigmoid normalization, and $\alpha_1, \alpha_2 \in [0,1]$ are tuning parameters, fixed as $\alpha_1 = 0.7$ and $\alpha_2 = 0.5$.

Backpropagation

Following in the footsteps of [Sun and Tappen, 2011] to obtain the gradient of the loss function L

$$\frac{\partial L}{\partial \beta} = \sum_{k=1}^{T} \frac{\partial L}{\partial \varphi^k} \frac{\partial \varphi^k}{\partial \beta} = \frac{\partial L}{\partial \varphi^T} \cdot \sum_{k=1}^{T} \{ \frac{\partial \varphi^k}{\partial \beta} \cdot \prod_{t=k}^{T-1} \frac{\partial \varphi^{t+1}}{\partial \varphi^t} \}. \tag{10}$$

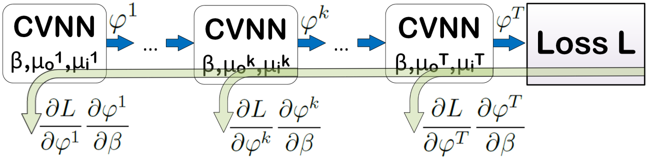


Figure: For backpropagation, we unwind the model T times and compute the gradient using Eq. (15).

Backpropagation

Using the the update (8), we get

$$\frac{\partial \varphi^{t+1}}{\partial \varphi^t} = 1 + \eta\Bigg(\frac{\partial g(\varphi^t, \beta)}{\partial \varphi^t} - 2(I - \mu_o) \cdot \frac{\partial \mu_o(\varphi^t)}{\partial \varphi^t} + 2(I - \mu_i) \cdot \frac{\partial \mu_i(\varphi^t)}{\partial \varphi^t}\Bigg), \tag{11}$$

where

$$\frac{\partial \mu_i(\varphi^t)}{\partial \varphi^t} = \frac{\delta_\epsilon(\varphi^t) \cdot (I - \mu_i)}{\int H_\epsilon(\varphi^t(x))dx}, \tag{12}$$

$$\frac{\partial \mu_o(\varphi^t)}{\partial \varphi^t} = \frac{\delta_\epsilon(\varphi^t) \cdot (\mu_o - I)}{\int (1 - H_\epsilon(\varphi^t(x)))dx} \tag{13}$$

and H_ϵ has been defined in Eq. (4) and δ_ϵ is its derivative.

Backpropagation

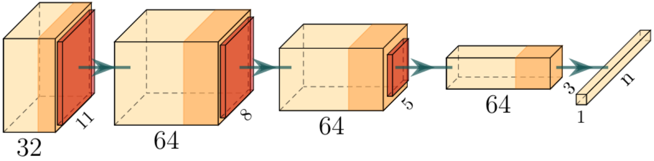
We also get

$$\frac{\partial \varphi^{t+1}}{\partial \beta} = \frac{\partial \varphi^{t+1}}{\partial g} \cdot \frac{\partial g}{\partial \beta} = \eta \cdot \frac{\partial g}{\partial \beta} \tag{14}$$

Consequently

$$\frac{\partial L}{\partial \beta} = \frac{\partial L}{\partial \varphi^{T}} \cdot \eta \cdot \sum_{k=1}^{T} \{ \frac{\partial g(\varphi^{k-1}, \beta)}{\partial \beta} \cdot \prod_{t=k}^{T-1} \frac{\partial \varphi^{t+1}}{\partial \varphi^{t}} \} \tag{15}$$

Preprocessing: Pixel-wise Detection



- Use a trained classifier to find organ pixels.
- Yields a very coarse initial segmentation i.e. detection map.
- Total number of parameters is 204033.

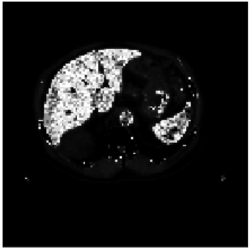
Pre-Processing: Probability map I

Algorithm 2 Probability Map Computation

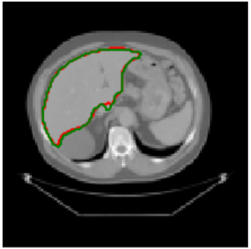
Input: CT scan C, initial binary mask D, number of bins N^{bins}.
Output: Probability map I.

1: Extract pixels inside the mask $\mathbf{u} = C(D > 0)$
2: Construct N^{bins} equally spaced *bins* in the range $[\min(\mathbf{u}), \max(\mathbf{u})]$
3: Compute counts $\mathbf{n} = histogram(\mathbf{u}, bins)$
4: Obtain $I(x) = \mathbf{n}(C(x)) / \max(\mathbf{n})$

detection map



probability map



segmentation and ground truth

CNN architecture

For CNN we used a network with

- 3 convolutional layers with 3 filters of size 3×3 with padding,
- A convolutional layer with 3 filters of size 1×1,
- Exponential linear unit (ELU), instead of ReLU activation,
- A convolutional layer with 1 filter of size 1×1.

The filters of size 3×3 used padding such that the size of the output was kept the same as the size of the input.

Multiple Initializations φ^0 for Training

To avoid overfitting, we need to train the CNN to recover from different initializations. We initialize with:

- Detection map
- Thresholded probability map with different thresholds and some connected components removed
- Distorted ground truth Y as shown below

Figure: Left: ground truth. Middle: distorted by added or punched semicircles with random radius at random border locations. Right: The middle image is corrupted by adding Gaussian noise and used as initialization for training.

φ's after 3 iterations

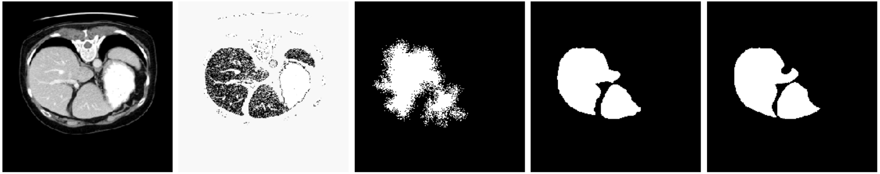


Figure: From left to right: CT Image (I), probability map, φ^0(initialization), φ^3 (segmentation), ground truth mask.

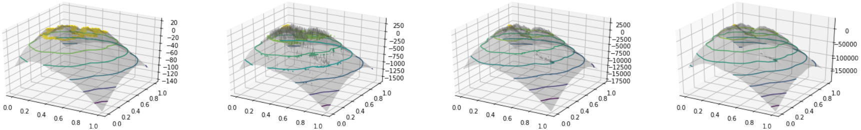


Figure: Level surface evolution through 3 iterations of CVNN. Left to right: φ^0, φ^1, φ^2, φ^3

Experiments & Results

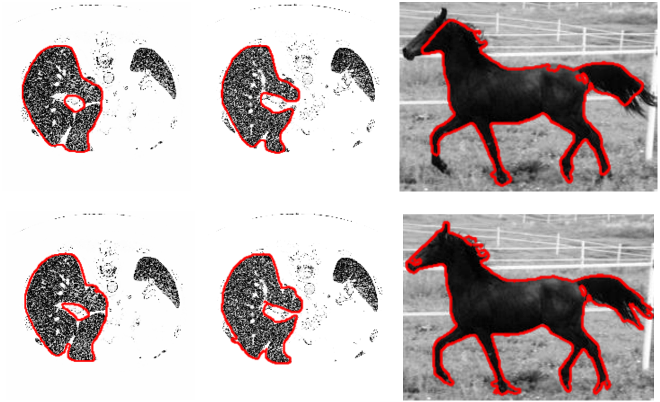


Figure: 2D segmentation results. Top: Chan-Vese with 100 iteration. Bottom: CVNN with 3 iterations.

Dice Coefficient

We will use the Dice coefficient in our experiments

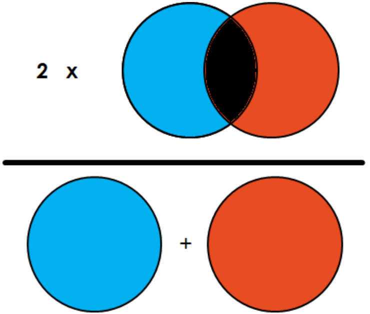


Figure: Computation of $Dice(A, B) = \frac{2|A \cap B|}{|A|+|B|}$.

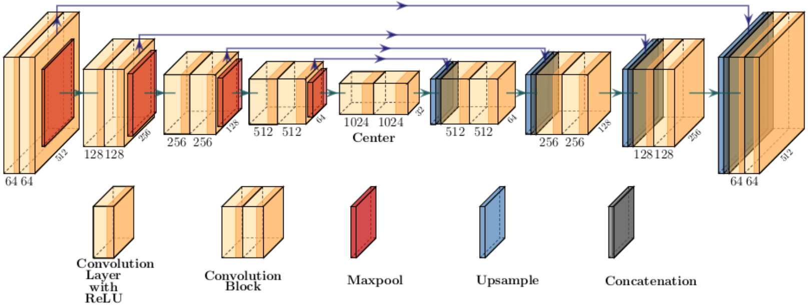
Experiments & Results

	CV-1^{it}	CV-10^{it}	CV-100^{it}	CVNN-1^{it}	CVNN-2^{it}	CVNN-3^{it}
Liver	90.85	91.72	92.69	92.48	93.56	**94.04**
Horse	50.61	53.83	67.66	**68.72**	68.36	68.10

Table: Dice coefficients on the test set obtained with 4-fold cross validation. 'CV - n^{it}' stands for standard Chan-Vese with n iterations, and 'CVNN - n^{it}' stands for our proposed method with n iterations. Note Dice of the initialization fed into CV and CVNN for liver is 90.43

Liver results are based on 4-fold cross validation. For Weizmann horse dataset [Borenstein and Ullman, 2002] 200 images reserved for testing and 128 of images which had aspect ratio close to 3x4 resized to 180x240 and used for training, testing images are not resized.

CVNN-UNet



Replacing the CNN with a U-Net

3D CVNN-UNet

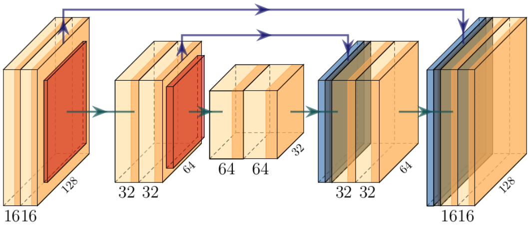


Figure: The 3D UNet architecture with 2 convolution blocks, each convolution block has 2 convolution layers.

- Each convolution layer is followed by an exponential linear unit, ELU, activation.
- Has only $\sim 353,000$ parameters to be optimized.
- Needs low-resolution inputs due memory issues
- The Chan-Vese update takes place in 3D, not in 2D.

3D CVNN-UNet

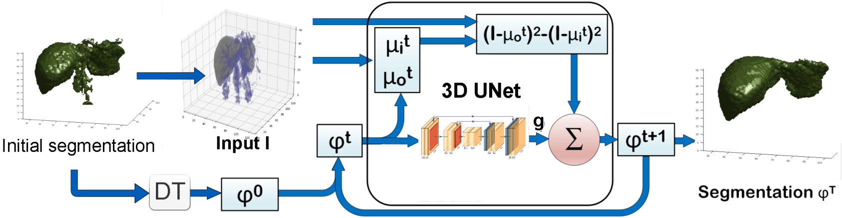


Figure: The 3D CVNN-UNet diagram.

The CVNN iterations

$$\varphi^{t+1} = \varphi^t + \eta\delta_\epsilon(\varphi^t)(g(\varphi^t, \beta) + (I - \mu_o)^2 - (I - \mu_i)^2) \tag{16}$$

3D Applications: Data

Same as [Gibson et al., 2018]:

- A multi-organ segmentation dataset.
- 90 CT scans: 43 from the TCIA Pancreas-CT dataset ([Clark et al., 2013]) and 47 from the BTCV dataset ([Landman et al., 2015])
- Provided reference segmentations for all 90 CT scans and up to 14 abdominal organs
- Some of the BTCV patients had metastatic liver cancer or other forms of abdominal pathologies.

3D Data Preprocessing

We have

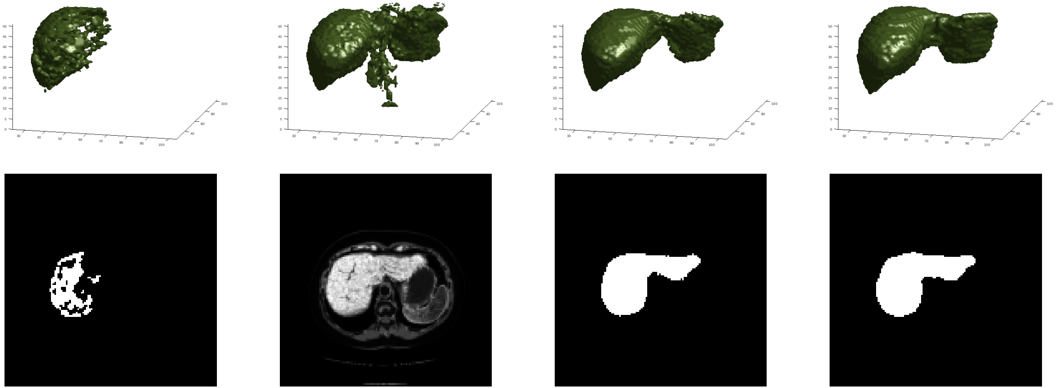
- resized the CT scans so that each CT scan in itself is isotropic (has the same resolution in all three directions)
- the axial dimensions are 512×512.

For the isotropic input size is $512 \times 512 \times 4k$

- resized to $256 \times 256 \times 2k$ (medium-resolution)
- resized to $128 \times 128 \times k$ (low-resolution)

Different resolution levels of the same CT scan gives us the flexibility to experiment with both computationally efficient and expensive 3D CVNN methods.

3D CVNN-UNet Results



detection $\varphi > 0$ | prob. map | segmentation | ground truth

Figure: Segmentation example of the 3D CVNN-UNet on a CT scan from the BTCV dataset.

3D CVNN with medium-resolution input

We used the same CVNN arhitecture that we used for 2D except each convolution kernel is $3 \times 3 \times 3$, with a total number of **592** parameters.

- Took the segmentation maps obtained from 4 iterations of 3D CVNN-UNet, and upsampled them 2x
- Used these upsampled segmentations as new detection maps and populated probability maps given newly derived detection maps using the probability map algorithm mentioned earlier.

3D CVNN

There are three main purposes for this step;

- To further improve the accuracy given the new detection and probability maps.
- The low-resolution segmentation would look coarse when upsampled. We aim to obtain finer segmentations for the upsampled input.
- To show that when we combine 3D CVNN-UNet and 3D CVNN, we can achieve high accuracy for medium resolution input while reducing computation complexity.

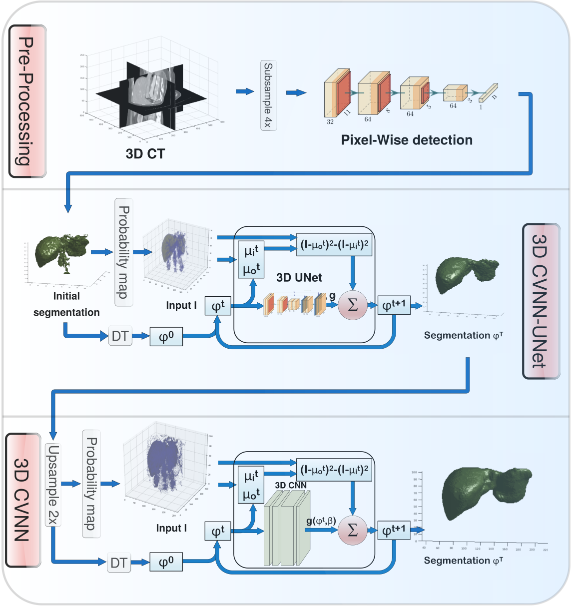


Figure: Diagram of the Deep Chan-Vese 3D approach.

Our Approach vs. State of the Art

Arhitecture	res	x-val folds	vols test	Dice	bnd err(mm)	95%Hauss/ dist(mm)	Segm. time(s)
DEEDS +JLF (Wang et al '12)	144	9	90	94	2.1	6.2	4740
VoxResNet (Chen et al. '16)	144	9	90	95	2.0	5.2	< 1
VNet (Milletari et al '16)	144	9	90	94	2.2	6.4	< 1
DenseVNet (Gibson et al. '18)	144	9	90	96	1.6	4.9	12
ObeliskNet (Heinrich et al '19)	144	4	43	95.4[1]	-	-	< 1
SETR (Zheng et al '21)	96	5	30	95.4	-	-	25
CoTr (Xie et al '21)	96	5	30	96.3	-	-	19
UNETR (Hatamizadeh et al '22)	96	5	30	97.1[2]	-	-	12
nnU-Net (Isensee et. al '20)	128	1	13	96.4[3]	1.7	-	10
DISSM (Raju et al '22)	-	1	13	96.5[4]	1.1	-	12
3D CVNN-UNet (ours)	128	4	90	95.6	1.67	4.42	0.53
Deep Chan-Vese 3D (ours)	256	4	90	95.2	1.49	4.40	0.64

Table: Comparison with the state of the art for liver segmentation. The 9-fold cross-validation results are from [Gibson et al., 2018], the 5-fold results from Hatamizadeh et. al. 2022, and the 1-fold from Raju et. al 2022.

Ablation Study

Questions?

- What is the contribution of U-Net vs CNN?
- What is the contribution of 3D vs 2D?

	3D	U-Net	φ^0	**1-it**	**2-it**	**3-it**	**4-it**
2D CVNN	-	-	87.58	92.75	93.66	93.63	93.68
2D CVNN-UNet	-	+	87.58	92.61	93.72	93.63	93.75
3D CVNN	+	-	87.58	88.29	90.23	91.43	91.74
3D CVNN-UNet	+	+	87.58	92.83	94.41	95.09	95.52

Table: Ablation results comparing 2D vs. 3D approaches and CNN vs UNet. The results are shown as average Dice scores obtained with 4-fold cross-validation.

Discussion

The Chan-Vese NN

- Reduced cost of computation.
- Outperforms Chan-Vese by CVNN both in terms of speed and accuracy.
- Competitive with the state of the art models, using a small number of parameters.

Looking Deeper at the CVNN

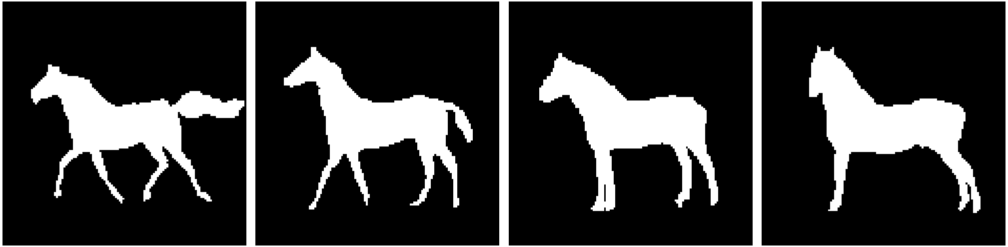
The CVNN Model:

$$\varphi^{t+1} = \varphi^t + \eta\delta_\epsilon(\varphi^t)(g(\varphi^t, \beta) + (I - \mu_o)^2 - (I - \mu_i)^2)$$

Motivation

- The CNN or U-Net $g(\varphi^t, \beta)$ helps update a level set function φ.
- What matters is only the zero level set $\varphi^{-1}(0) =$ a shape.
- The input φ^0 is a corrupted (noisy) shape.
- The CNN is trained to recover from many noisy shapes.
- How well can it work in one step, without the data term?
- What kind of possible corruptions=noises are there?
- We call this task Shape Denoising.

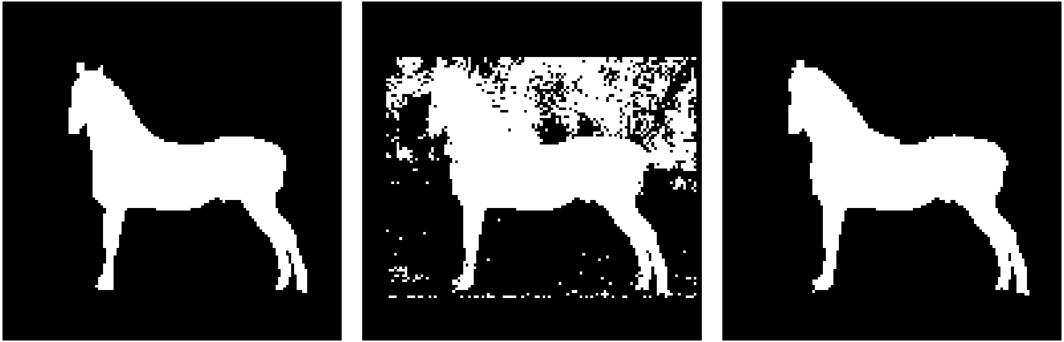
Shape Denoising



- The shapes are represented as binary images of a certain size, e.g. 128×128
- The focus of our study is to study shape denoising - the task of recovering a shape corrupted by noise
- Similar to Post-DAE [Larrazabal et al., 2020] but different focus
- We will introduce different types of shape noise
- We will evaluate and compare the performance of seven shape modeling methods for shape denoising.

The Shape Denoising Problem

Shape denoising is the process of removing the noise from a shape, with the goal of obtaining a shape as close to the original shape as possible.

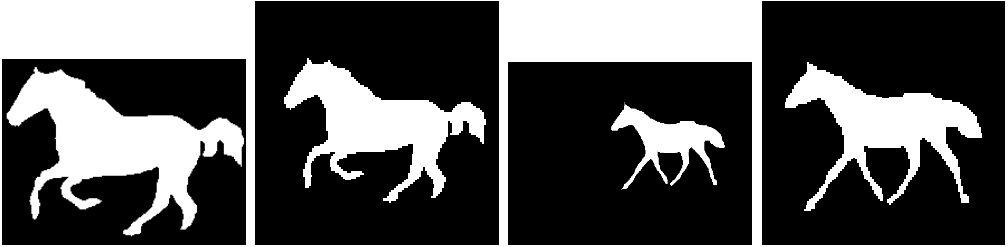


(a) Original shape (b) Noisy shape (c) Denoised shape

Figure: Shape denoising example. The noisy shape (b) has been obtained from the original shape (a) by a noise inducing process. A shape denoising method is used to obtain the denoised shape (c).

Shape Alignment

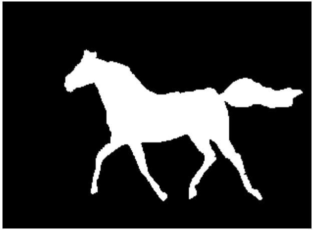
- We used binary images of size 128×128.
- All binary images were aligned to have the objects centered and of approximately the same size.

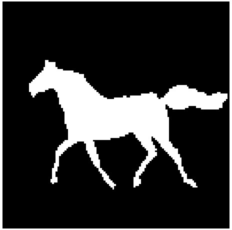


(a) mask image1 (b) aligned image1 (c) mask image2 (d) aligned image2

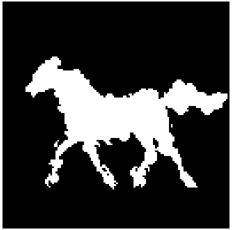
Figure: Two alignment examples from the Weizmann Horse Dataset.

Six types of noise

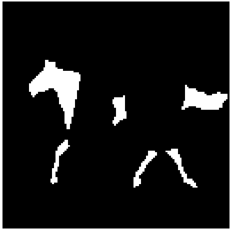

original,


aligned


salt and pepper


circle


real image


occlusion


detection image


thr. probability

Salt and Pepper Noise

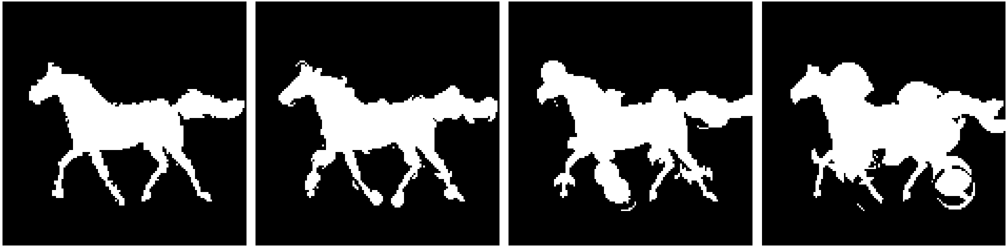
The salt and pepper noise is obtained by flipping each pixel to its opposite value with a probability p.

(a) $p = 0.01$ (b) $p = 0.05$ (c) $p = 0.10$ (d) $p = 0.15$

Figure: Salt and pepper noise with different levels p.

Circle Noise

The circle noise is obtained by adding semicircles or punching holes at random locations on the boundary between the foreground and the background.

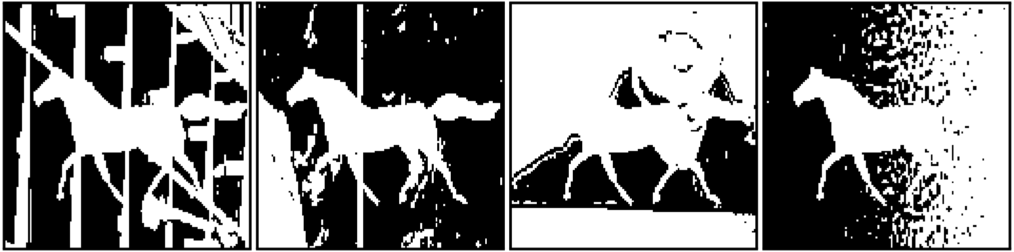


(a) $r = 1$ (b) $r = 3$ (c) $r = 6$ (d) $r = 10$

Figure: Circle noise with different levels.

Real Image Noise

- Noisy backgrounds are obtained by thresholding real images using various thresholds
- The shape background pixels are replaced with the noisy background

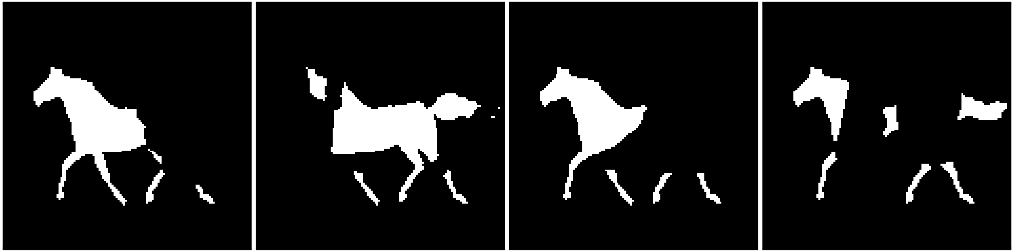


(a) sample 1 (b) sample 2 (c) sample 3 (d) sample 4

Figure: Examples of real image noise.

Occlusion Noise

The shape foreground pixels are occluded using thresholded real images.



(a) sample 1 (b) sample 2 (c) sample 3 (d) sample 4

Figure: Examples of occlusion noise.

Detection Image Noise

Detection image noise is obtained as the segmentation of an object from a color or grayscale image using a trained CNN (convolutional neural network).

(a) object shape (b) color image (c) detection image noise

Figure: Example of detection image noise in the Weizmann Horse Dataset.

Thresholded Probability Noise

Uses a color image I and a binary image M representing the object in the image.

- Denote $C_1 = \{(x, y), M(x, y) = 1\}$ as the foreground region and $C_0 = \{(x, y), M(x, y) = 0\}$ as the background region.
- k-means clustering is used to partition the image into k clusters, obtaining cluster indices for all pixels $L \in \{1, 2, ..., k\}^N$.
- For cluster $i \in \{1, 2, ..., k\}$, obtain the number of pixels that belong to foreground or background:

$$\begin{aligned} N_{i1} &= |\{(x, y)|L(x, y) = i \wedge (x, y) \in C_1\}|, \\ N_{i0} &= |\{(x, y)|L(x, y) = i \wedge (x, y) \in C_0\}| \end{aligned} \tag{17}$$

- Then the probability map of color image I can be computed as follows:

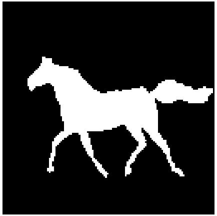
$$P(x, y) = \frac{N_{j1}}{N_{j1} + N_{j0}}, \text{ where } j = L(x, y). \tag{18}$$

MICAD2022 Medical Imaging and Computer-Aided Diagnosis

MICCAI Society Endorsed Event

Thresholded Probability Noise

Binary noisy shapes are obtained by thresholding the probability map P.



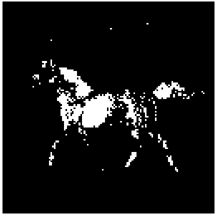
object shape

color image

prob map

threshold 0.04

threshold 0.5



threshold 0.98

Clean Image Datasets

The **Weizmann Horse dataset** [Borenstein et al., 2004]:

- Contains 327 horse images and their corresponding mask images.
- 159 images were randomly selected as the training set S_{train}^{clean} and the other 168 images as the test set S_{test}^{clean}.

The Caltech-UCSD Birds 200 dataset [Welinder et al., 2010] contains photos of 200 bird species.

- We use 417 images of seven Flycatcher species in our experiments.
- 207 images were randomly selected as the training set S_{train}^{clean} and the other 210 images as the test set S_{test}^{clean}.

IoU

In our experiments, the criterion we use to estimate the performance of modeling against noises is Intersection over Union (IoU), also known as the Jaccard Index.

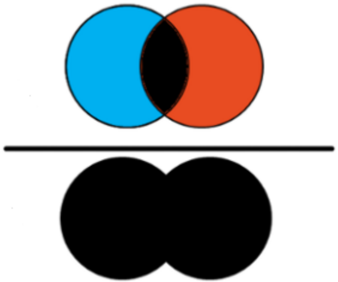


Figure: Computation of $IOU(A, B) = \frac{|A \cap B|}{|A \cup B|}$.

Noisy Image Datasets

We denote the set $\{S_{test}^{salt}, S_{test}^{circle}, S_{test}^{real}, S_{test}^{occlusion}, S_{test}^{detection}, S_{test}^{probability}\}$ as S_{test}^{all}.

Dataset	IoU(begin)	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1
Horse	S_{test}^{salt}	1000	1000	1000	1000	1000
	S_{test}^{circle}	77	1000	1000	1000	1000
	S_{test}^{real}	1000	1000	1000	1000	1000
	$S_{test}^{occlusion}$	479	622	690	938	1779
	$S_{test}^{detection}$	3	7	14	77	60
	$S_{test}^{probability}$	298	673	1095	1345	525
Bird	S_{test}^{salt}	1000	1000	1000	1000	1000
	S_{test}^{circle}	416	1000	1000	1000	1000
	S_{test}^{real}	1000	1000	1000	1000	1000
	$S_{test}^{occlusion}$	435	556	809	972	1000
	$S_{test}^{detection}$	22	30	44	56	15
	$S_{test}^{probability}$	843	962	934	519	70

Table: Number of noisy test images in each noise category.

Shape Denoising Methods Evaluated

The following methods have been evaluated for shape modeling and denoising:

- Active Shape Model (ASM) [Cootes et al., 1995]
- Deep Boltzman Machine (DBM) [Salakhutdinov and Hinton, 2009]
- Centered Convolutional DBM (CDBM) [Yang et al., 2021]
- Energy Based Model (EBM) [Pang et al., 2020]
- U-Net [Ronneberger et al., 2015]
- DeepLab V3+ [Chen et al., 2018]
- Masked Autoencoder (MAE) [He et al., 2021]

Masked Autoencoder (MAE)

The masked autoencoder (MAE) [He et al., 2021]

- Scalable self-supervised learner
- Trained to fill-in missing parts of an image

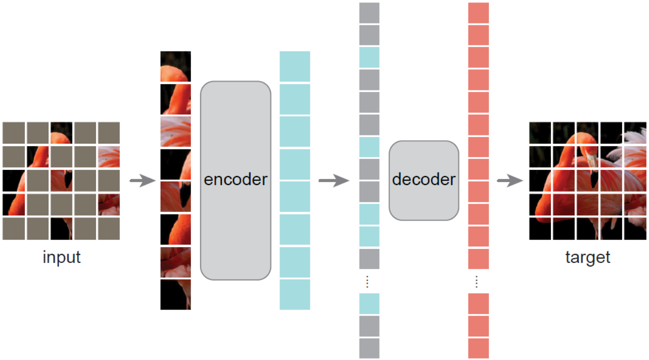


Figure: MAE architecture [He et al., 2021]

Salt and Pepper Noise Example

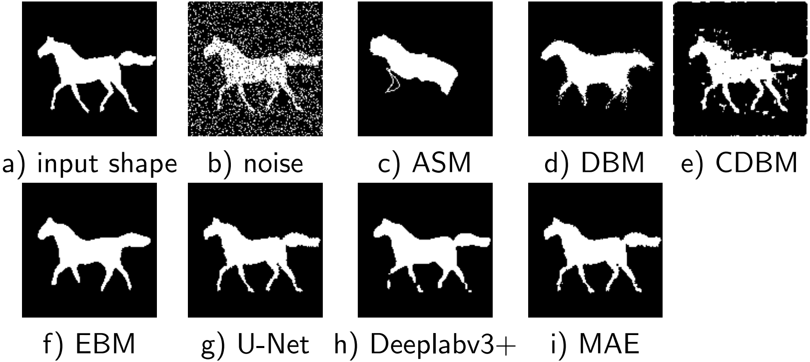


Figure: Example of results of different methods on a shape perturbed by salt and pepper noise.

Salt and Pepper Noise Results

- Performance comparison of all methods against salt and pepper noise

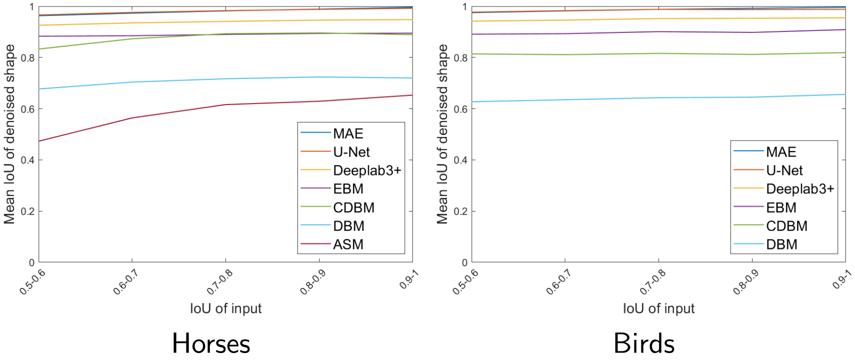


Figure: Performance on salt and pepper noise data S_{test}^{salt}.

Circle Noise Example

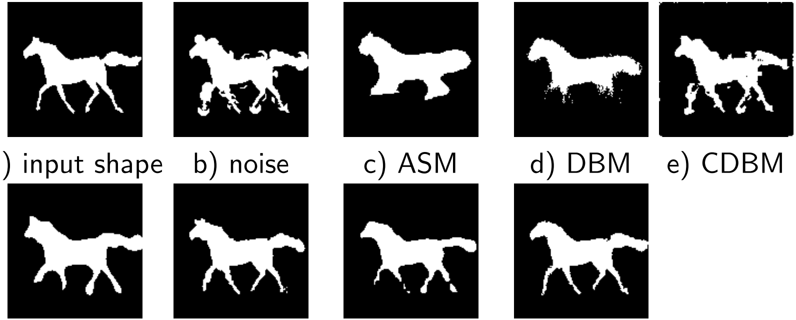
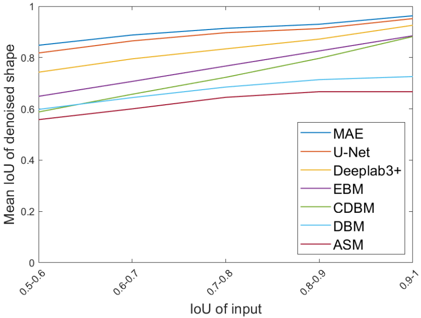


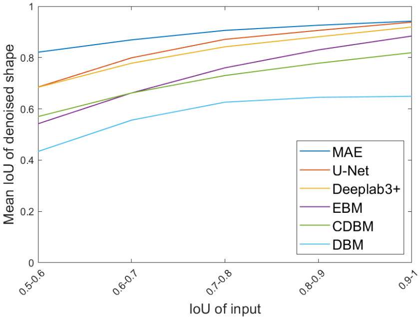
Figure: Example of results of different methods on a shape perturbed by circle noise.

Circle Noise Results

- Performance comparison of all methods against circle noise



Horses



Birds

Figure: Performance on circle noise data S_{test}^{circle}.

Real Image Noise Example

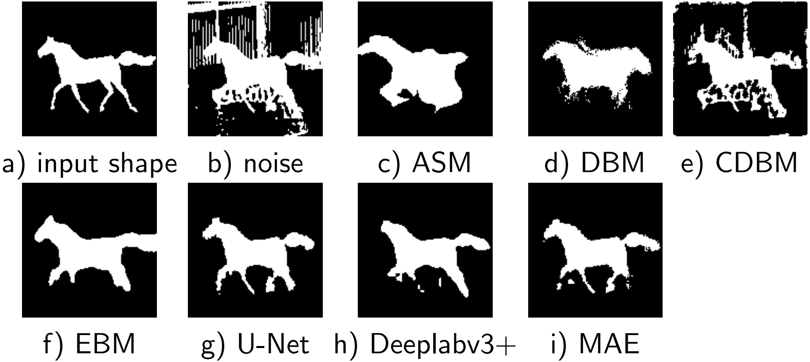
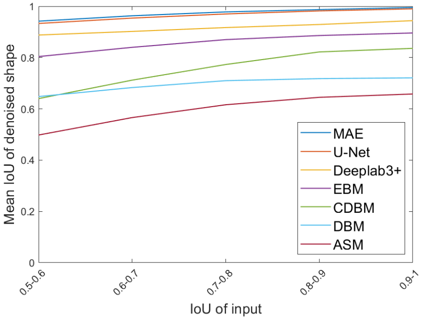


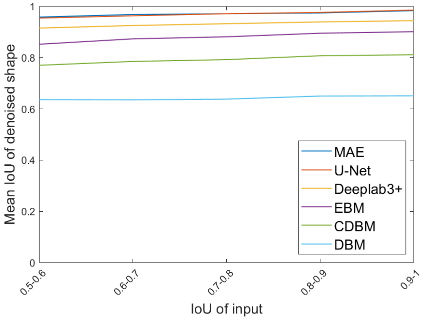
Figure: Example of results of different methods on a shape perturbed by real image noise.

Real Image Noise Results

- Performance comparison of all methods against real image noise



Horses



Birds

Figure: Performance on real image noise data S_{test}^{real}.

Occlusion Noise Example

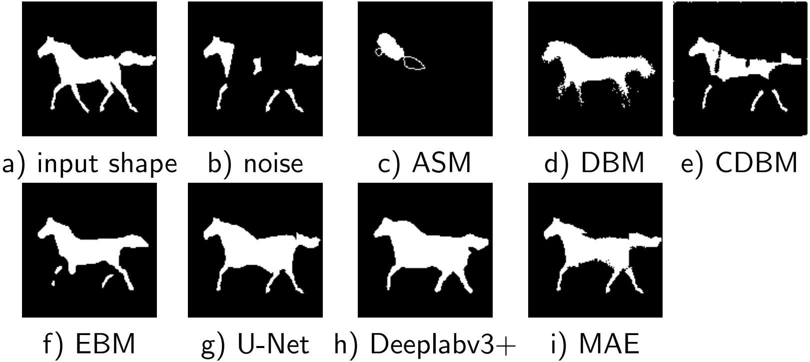
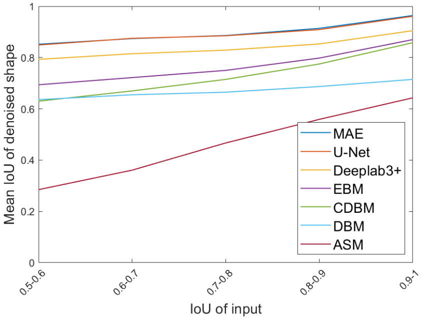


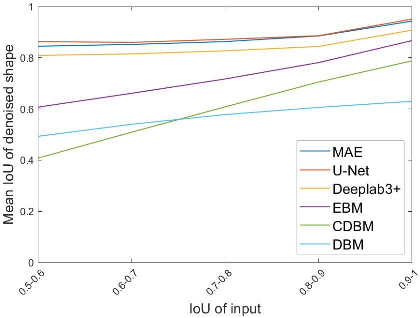
Figure: Example of results of different methods on a shape perturbed by occlusion noise.

Occlusion Noise Results

- Performance comparison of all methods against occlusion noise



Horses



Birds

Figure: Performance on real image noise data $S_{test}^{occlusion}$.

MICAD2022 Medical Imaging and Computer-Aided Diagnosis
MICCAI Society Endorsed Event

Detection Image Noise Example

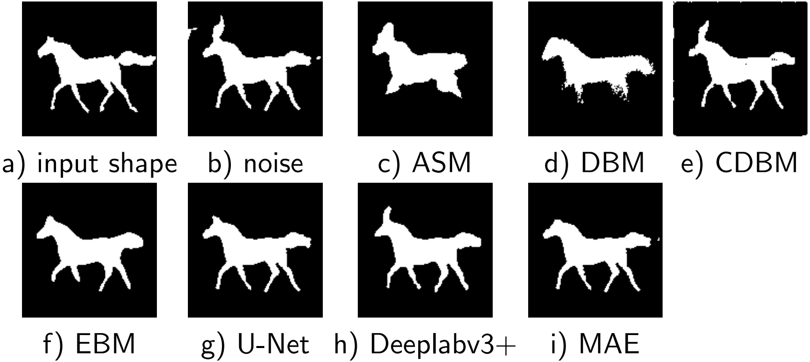


Figure: Example of results of different methods on a shape perturbed by detection image noise.

Detection Image Noise Results

- Performance comparison of all methods against detection image noise

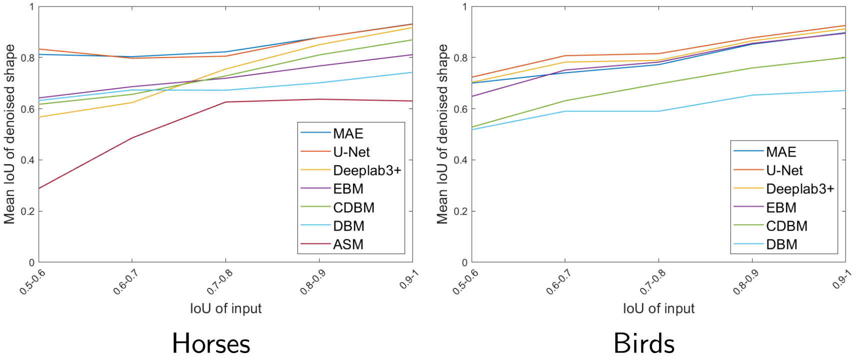


Figure: Performance on real image noise data $S_{test}^{detection}$.

Thresholded Probability Noise Example

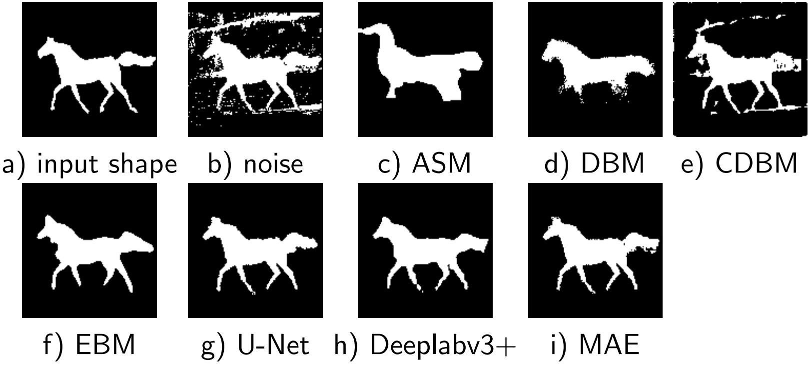


Figure: Example of results of different methods on a shape perturbed by thresholded probability noise.

Thresholded Probability Noise Results

- Performance comparison of each methods against thresholded probability noise

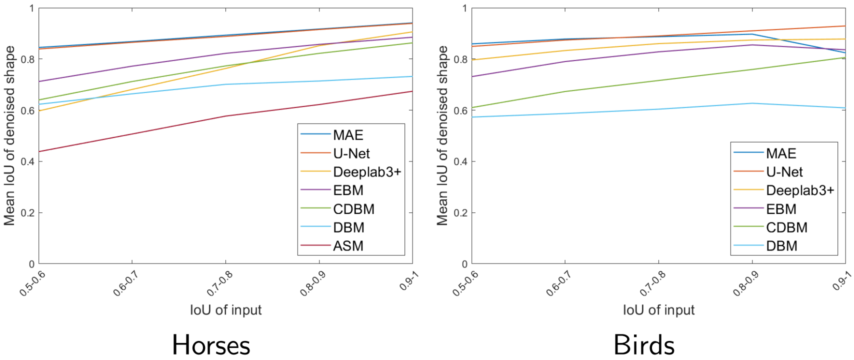


Figure: Performance on thresholded probability noise data $S_{test}^{probability}$.

Summary and Discussion

Comparing the methods:

- Experiments reveal that MAE and U-Net are the best shape denoising methods we evaluated for all six types of noise.
- DeepLabv3+ is the third best shape denoising method for the six noise types in most situations.
- EBM outperforms CDBM on all six noise types, especially when dealing with real image noise.

Comparing the noise types:

- The salt and pepper noise is the easiest to deal with, followed by real image noise.
- Circle noise and occlusion noise are more challenging than the above two, especially when the noise level is high.
- The most challenging noises among these six are the thresholded probability noise and detection image noise.

Conclusions

CVNN - a method for object segmentation

- Generalizes the Chan-Vese level set method, replacing the length-based regularization with a trained CNN shape model
- The CNN is trained end-to-end by backpropagation
- Fast and competitive with state of the art 3D liver segmentation methods
- Multiple types of initializations are used to avoid overfitting
- The initializations lead us to the shape denoising problem

A study of methods for shape denoising

- Shapes are represented as binary images
- Shapes are aligned by translation and scaling
- We introduced six types of shape noise
- We evaluated seven shape modeling/denoising methods on these types of noise

Future Work

Future work:

- Study shape denoising in the wild, where the shapes are not aligned
- Study methods trained on aligned shapes vs. methods trained on unaligned shapes
- Come back to CVNN, apply what we learned to object segmentation

References I

Borenstein, E., Sharon, E., and Ullman, S. (2004).
Combining top-down and bottom-up segmentation.
In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 46–46. IEEE.

Borenstein, E. and Ullman, S. (2002).
Class-specific, top-down segmentation.
In *ECCV*, pages 109–122.

Chan, T. F. and Vese, L. A. (2001).
Active contours without edges.
IEEE Transactions on Image Processing, 10:266–277.

References II

- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018).
  Encoder-decoder with atrous separable convolution for semantic image segmentation.
  In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818.
- Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al. (2013).
  The cancer imaging archive (tcia): maintaining and operating a public information repository.
  Journal of digital imaging, 26(6):1045–1057.
- Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995).
  Active shape models-their training and application.
  Computer vision and image understanding, 61(1):38–59.

References III

- Gibson, E., Giganti, F., Hu, Y., Bonmati, E., Bandula, S., Gurusamy, K., Davidson, B., Pereira, S. P., Clarkson, M. J., and Barratt, D. C. (2018). Automatic multi-organ segmentation on abdominal ct with dense v-networks. *IEEE transactions on medical imaging*, 37(8):1822–1834.
- He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. (2021). Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*.
- Landman, B., Xu, Z., Igelsias, J., Styner, M., Langerak, T., and Klein, A. (2015). Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge.
- Larrazabal, A. J., Martínez, C., Glocker, B., and Ferrante, E. (2020). Post-dae: anatomically plausible segmentation via post-processing with denoising autoencoders. *IEEE transactions on medical imaging*, 39(12):3813–3820.

References IV

Pang, B., Han, T., Nijkamp, E., Zhu, S.-C., and Wu, Y. N. (2020).
Learning latent space energy-based prior model.
arXiv preprint arXiv:2006.08205.

Ronneberger, O., Fischer, P., and Brox, T. (2015).
U-net: Convolutional networks for biomedical image segmentation.
In *MICCAI*, pages 234–241. Springer.

Salakhutdinov, R. and Hinton, G. (2009).
Deep boltzmann machines.
In *Artificial intelligence and statistics*, pages 448–455. PMLR.

Sun, J. and Tappen, M. (2011).
Learning non-local range markov random field for image restoration.
In *CVPR*, pages 2745–2752. IEEE Computer Society.

References V

Taghanaki, S. A., Zheng, Y., Zhou, S. K., Georgescu, B., Sharma, P., Xu, D., Comaniciu, D., and Hamarneh, G. (2019).
Combo loss: Handling input and output imbalance in multi-organ segmentation.
Computerized Medical Imaging and Graphics, 75:24–33.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. (2010).
Caltech-UCSD Birds 200.
Technical Report CNS-TR-2010-001, California Institute of Technology.

Yang, J., Liu, S., and Wang, X. (2021).
Centered convolutional deep boltzmann machine for 2d shape modeling.
Personal and Ubiquitous Computing, pages 1–11.