Implicit Neural Representations of Skeletons for Historical Glyphs

Pooja Shetty

M.Sc. Artificial Intelligence, Universität Erlangen-Nürnberg (FAU), Erlangen
pooja.shetty@fau.de

Abstract. Skeleton extraction is an important process for the shape analysis and crucial feature extraction of an image. Several methods exist to accomplish this task but they come with the drawback that the images should have smooth surfaces. But this is not possible in all scenarios such as Historical Glyphs. This project tries to represent skeletons as Implicit Neural Representations using SIREN network to generalize over any kind of surface. SIREN network uses sine activation function which makes it capable of representing complex functions. In our experiment, we observe that this network is not very effective in representing the glyph skeletons.

Keywords: Implicit Neural Representation, Medial Axis, Deep Learning

1 Introduction

By skeletanization we mean extracting the center line structure of an object. There are different definitions of skeleton but the most important definition is given by Blum [1] which defines the "Medial Axis" of a surface. The medial axis is defined as the set of points inside a shape that have more than one closest point on the object's boundary. This means each point on the medial axis is equidistant from at least two points on the boundary of the shape. The concept is closely related to the notion of the "symmetry axis," which Blum originally envisioned as a way to describe biological and natural shapes through their symmetrical properties. The goal is to reduce the information in the image while preserving the structural connectivity, which can be useful for tasks like shape analysis, pattern recognition, and feature extraction. Hence here by skeleton we mean medial axis transform.

There are several ways to represent the skeletal representation like binary map, flux field (see [14]) or Distance Transform. In any way we are always mapping from one image representation (input image) to an output image representation (skeleton representation). In this project we would not use img2img approach but instead implicit representation of skeletons.

Implicit neural representations are a way to parameterize signals of any kind. Conventionally signals like images are represented in a discrete format for example images as discrete grids of pixels. Implicit Neural Representations parameterize a signal as a continuous function that maps the domain of the signal

(i.e., a coordinate, such as a pixel coordinate for an image) to whatever is at that coordinate (for an image, an R,G,B color). Of course, these functions are usually not analytically tractable - it is impossible to "write down" the function that parameterises a natural image as a mathematical formula. Implicit Neural Representations thus approximate that function via a neural network. Implicit neural networks have the benefit that they are not coupled to spatial resolution instead an image is coupled to the number of pixels. This is because they are continuous functions. The memory required to parameterize the signal is independent of spatial resolution, and only scales with the complexity of the underyling signal. They can also be sampled at arbitrary spatial resolutions (from [12]).

Our idea is to represent the image skeleton as an Implicit Neural Representation. We use an AutoEncoder model that takes the input image and computes its latent representation. This latent representation has the important features of the input image. The latent representation along with one or more query coordinates (x,y) of the resultant image are fed into a MLP decoder network (in our case SIREN Net) that outputs a scalar for each query coordinate. These scalars correspond to the distance of (x,y) to the relative closest point on the skeleton. Here we are using Signed Distance Function to represent the skeleton. Signed Distance Function is used to represent the distance of a given point in space to the nearest surface of an object. The function assigns positive or negative values based on the point's position relative to the boundary—inside or outside. Hence the scalars represent the signed distance values at (x,y).

2 Related Works

Skeleton extraction from geometric data is a well researched problem and a lot of work has already been done in this area. We seek your attention to [7] which compares the different 2D skeleton extraction techniques.

We are here mostly interested in Medial Axis extraction. The medial axis is mathematically defined as the set of points having more than one closest point on the surface [1]. We have lot of methods which describe the Medial Axis extraction methods. [5] and [9] obtain the medial axis by implementing error minimization techniques.[4] and [3] extract medial axis by approximating it using a Voronoi Diagram.

Deep learning techniques have also been used for skeleton computation. For example, Point2Skeleton [6] uses a PointNet encoder [8] to synthesize skeletal point, and predict their links using a graph auto-encoder. However the skeletal points are not mathematically defined and the losses only encourage a medial-axis like position. Furthermore it requires to train on a shape database and is therefore limited to shapes that fit in the learned latent space. P2MAT-NET [15] estimates the set of medial points and medial spheres using a ground truth skeleton-shape dataset and links the medial points using a Delaunay triangulation which is later pruned.

Deep Medial Fields [10] was the first paper to link neural fields and medial axis transform, by estimating a medial field. For each point in the ambient space, it evaluates the shape width (i.e. the distance between the surface and the medial axis) in the "slice" containing the point. However this medial field is discontinuous near the surface, and two neural networks are used for handling this discontinuity.

Recently the paper [2] focuses on a novel approach to use implicit neural representations for extracting medial skeletons of 3D shapes. This deviates from traditional applications of implicit neural representations, which are primarily designed for surface extraction and rendering. Instead, this work extends the application to extract the medial axis or skeleton of shapes, which is traditionally challenging due to the need for precise distance measurements from the object's surface by using a Total Variation term([2]).

Most of the above mentioned skeleton extraction methods require smooth surfaces. Our aim is to perform implicit neural representation of the skeleton by using SIREN network to try and see if this drawback can be avoided.

3 Basic Architecture

AutoEncoders

Autoencoders are a type of artificial neural network used primarily for unsupervised learning of efficient codings, typically for the purpose of dimensionality reduction or feature extraction. The main goal of an autoencoder is to learn a representation (encoding) for a set of data, typically for the purpose of dimensionality reduction, by training the network to ignore signal noise. The basic architecture of an autoencoder consists of an encoder and a decoder. The encoder compresses the input into a latent-space representation. It learns to reduce the input dimensions and capture the important features in a more compressed form. The latent representation also called the Bottleneck has fewer dimensions than the input data. The decoder attempts to reconstruct the input data from the latent space representation. Its goal is to generate output as close as possible to the original input, effectively learning the distribution of the data.

In this project we have used U-net(from [11]) architecture as autoencoder model. Though U-net was specifically designed for image segmentation tasks, its applicability goes beyond it. Architecture wise U-net also has an encoder and a decoder just like an autoencoder except that it has an additional skip connection between the corresponding layers of the encoder and the decoder. U-Net model processes a given image by progressively lowering the feature map resolution in the encoder and then increasing the resolution in the decoder. The skip connections are given through reset blocks at each resolution. Because of skip connections, U-Net models can learn deep contextual and local information effectively and are better at extracting both coarse and fine feature information.

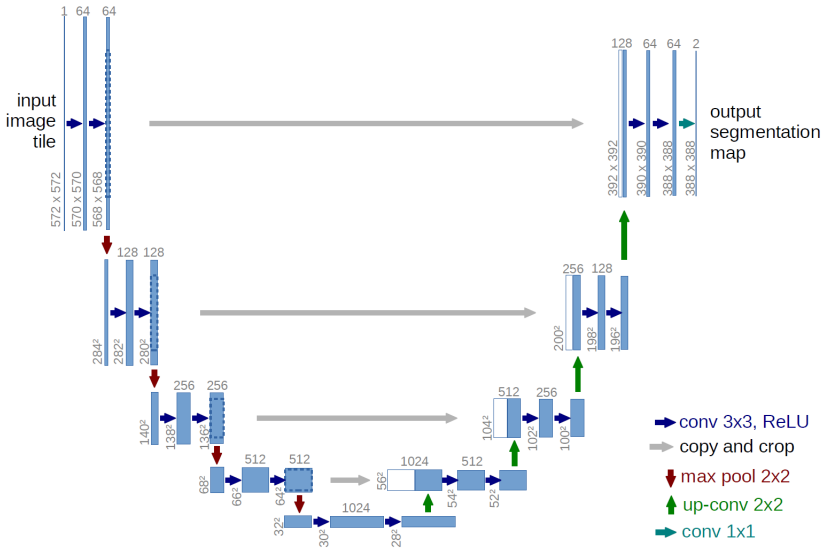

Fig. 1: U-net architecture from [11]

SIREN network

SIREN, or Sinusoidal Representation Network (from [13]), is a periodic activation function for implicit neural representations that uses the sine as a periodic activation function:

$$\Phi(x) = W_n(\varphi_{n-1} \circ \varphi_{n-2} \circ \ldots \circ \varphi_0)(x) + b_n, \tag{1}$$

where $x_i \mapsto \varphi_i(x_i) = \sin(W_i x_i + b_i)$. Here, $\varphi_i : \mathbb{R}^{M_i} \to \mathbb{R}^{N_i}$ is the i-th layer of the network. It consists of the affine transform defined by the weight matrix $W_i \in \mathbb{R}^{N_i \times M_i}$ and the biases $b_i \in \mathbb{R}^{N_i}$ applied on the input $x_i \in \mathbb{R}^{M_i}$, followed by the sine nonlinearity applied to each component of the resulting vector.

SIREN has been shown to be particularly effective for tasks involving the modeling of complex natural signals and their derivatives, such as audio, images, and other spatially or temporally coherent signals. This is mainly possible because of use of sinusoidal activation function. Sinusoids are naturally able to represent complex patterns and waveforms, which are common in many types of natural data. The can also manage high-frequency content in data, making them excellent for tasks where precision and detail are critical. The use of sinusoidal activations also make the entire network smoothly differentiable. This property is advantageous for applications in physics-informed neural networks, where one might need to compute gradients or higher-order derivatives of the network output with respect to the input.

4 Dataset and Data Preprocessing

We have used the glyph dataset from the university database for this project. The dataset has glyph images and their corresponding skeletons extracted through conventional techniques. There are a total of 27,956 glyph-skeleton pairs.

Each image or skeleton has a shape of 64x64. For each image skeleton pair, the image is first transformed to range [0,1] and normalized. The skeleton is first converted to its signed distance values using the Euclidean Distance Transform and then we also transform it to range [0,1] and normalize it. Each sample from the dataloader consists of image, 64x64 pixel coordinate pairs and their corresponding signed distance values.

Fig.2 shows an example glyph image and its corresponding skeleton and signed distance representation.

(a) (b) (c)

Fig. 2: Fig.2 (a) is a glyph image from the dataset (b) is its corresponding skeleton image (c) is the signed distance transform of the skeleton image

5 Implementation

Our model is trained using an AutoEncoder and SIREN network as mentioned in Section 3. The training is divided into two parts. First, all the glyph images from the trainset are trained using an AutoEncoder model to extract their latent representation. We use U-net as the AutoEncoder model here. We use image reconstruction loss at the end of the decoder which is computed using the Mean Squared Error between the original and the reconstructed image. The output of the encoder gives us the image latent values consisting of image important features. We used a learning rate of 0.0001 and a batch size of 128.

Next the latent representation of the image is concatenated with each pair of 4096 pixel coordinates and fed into the SIREN network. The original SIREN network is slightly modified by increasing the number of layers and neurons in each layer to handle the huge data. SIREN uses sine activation function. This network is then supposed to output the skeleton for the image by outputting the signed distance value corresponding to each pixel value. We use Mean Squared Error between the actual and predicted signed distance values. We used an initial

learning rate of 0.0001, batch size of 256, 512 hidden features, 5 hidden layers and omega(a frequency factor which simply multiplies the activations before the nonlinearity) as 30. This gave us a loss of about 4. Again retraining from here step wise by reducing the learning rate by 0.1 for every 30 epochs we were able to reduce the loss to 1.01.

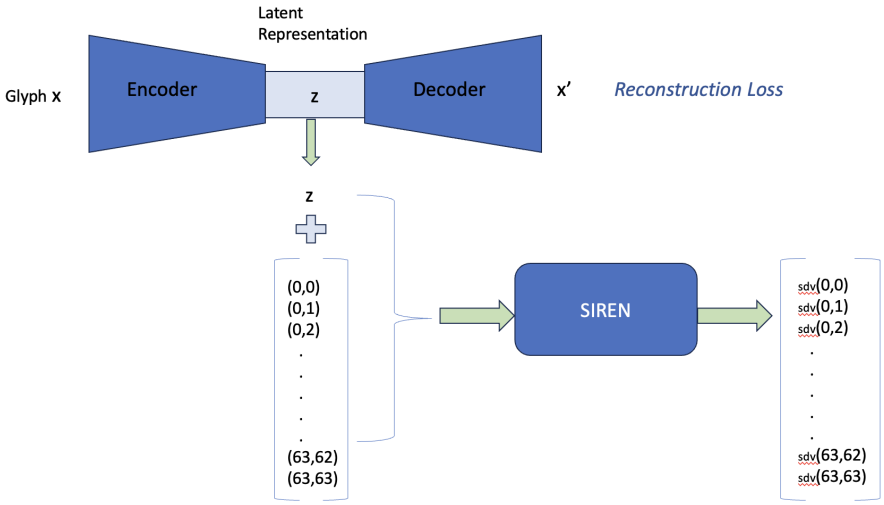

Fig. 3: Block diagram explaining the entire training process

6 Results

The U-net model learns the latent representations very well. This was verified by extracting the latent values of some glyphs and running them through the decoder to see if they construct the same image.

The hyperparameters of the SIREN network as given in Section 5 are selected after trying out several values. It is seen that any deviation from the mentioned hyperparameter worsens the loss for the given trainset. Even after that we could only reduce the loss to 1.01.

The output we obtain from the model is the signed distance representation. In order to get the skeleton image, we used the threshold method. Ideally all the values equal to 0 should correspond to the skeletal structure. But in our case getting an exact 0 in all cases is not possible so we want the values corresponding to the pixels representing the skeleton to be as close to zero as possible. So we set a threshold T and set every pixel less than T as 0 and the rest as 255.

Fig.4 shows the results obtained for different threshold values T.

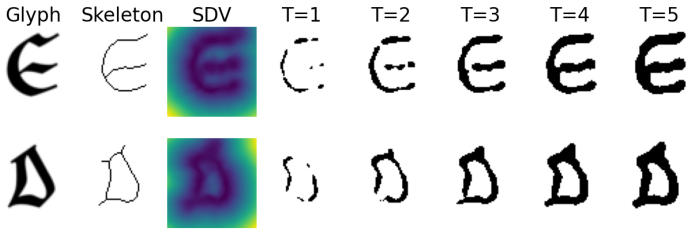

Fig. 4: Skeletal images for different threshold values

We can observe that for T=1, we do not see clear edges. As we increase the value of T we see clear skeleton structure but their thickness also increases giving us structures similar to the actual image, not its skeleton.

7 Conclusion

We know that Implicit Neural Representation is a great tool to represent complex patterns specially with SIREN network. But with this project we could see that we did not accomplish the desired objective of representing the skeletons of our glyph dataset as a continuous function. The SIREN network was not able to fully accommodate all the variations in the train set. It does work perfectly well for individual images but suffers for larger dataset where it needs to accommodate many number of minute details. Further fine-tuning is needed to get the desired output but the direction to choose is unclear. One of the difficulties here is understanding the information flow from the image features to the siren network and the coordinate combination with the image features. It might be worth checking how the model performs by feeding the skeleton as a binary map and not as a signed distance representation and trying out different coordinate and image latent combinations.

References

1. A transformation for extracting new descriptions of shape | CiNii Research, https://cir.nii.ac.jp/crid/1571980074649393792
2. Clémot, M., Digne, J.: Neural skeleton: Implicit neural representation away from the surface. Computers & Graphics **114**, 368–378 (Aug 2023). https://doi.org/10.1016/j.cag.2023.06.012, https://www.sciencedirect.com/science/article/pii/S0097849323001085
3. Dey, T.K., Zhao, W.: Approximate medial axis as a Voronoi subcomplex. Computer-Aided Design **36**(2), 195–202 (Feb 2004). https://doi.org/10.1016/S0010-4485(03)00061-7, https://linkinghub.elsevier.com/retrieve/pii/S0010448503000617
4. Dey, T.K., Zhao, W.: Approximating the Medial Axis from the Voronoi Diagram with a Convergence Guarantee. Algorithmica **38**(1), 179–200 (Jan 2004). https://doi.org/10.1007/s00453-003-1049-y, http://link.springer.com/10.1007/s00453-003-1049-y
5. Li, P., Wang, B., Sun, F., Guo, X., Zhang, C., Wang, W.: Q-MAT: Computing Medial Axis Transform By Quadratic Error Minimization. ACM Trans. Graph. **35**(1), 1–16 (Dec 2015). https://doi.org/10.1145/2753755, https://dl.acm.org/doi/10.1145/2753755
6. Lin, C., Li, C., Liu, Y., Chen, N., Choi, Y.K., Wang, W.: Point2Skeleton: Learning Skeletal Representations from Point Clouds (Apr 2021), http://arxiv.org/abs/2012.00230, arXiv:2012.00230 [cs]
7. Németh, G., Kovács, G., Fazekas, A., Palágyi, K.: A method for quantitative comparison of 2D skeletons. Acta Polytechnica Hungarica **13**(7), 123–142 (2016)
8. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation (Apr 2017), http://arxiv.org/abs/1612.00593, arXiv:1612.00593 [cs]
9. Rebain, D., Angles, B., Valentin, J., Vining, N., Peethambaran, J., Izadi, S., Tagliasacchi, A.: LSMAT Least Squares Medial Axis Transform. Computer Graphics Forum **38**(6), 5–18 (Sep 2019). https://doi.org/10.1111/cgf.13599, https://onlinelibrary.wiley.com/doi/10.1111/cgf.13599
10. Rebain, D., Li, K., Sitzmann, V., Yazdani, S., Yi, K.M., Tagliasacchi, A.: Deep Medial Fields (Jun 2021), http://arxiv.org/abs/2106.03804, arXiv:2106.03804 [cs]
11. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation (May 2015), http://arxiv.org/abs/1505.04597, arXiv:1505.04597 [cs]
12. Sitzmann, V.: Awesome implicit representations - a curated list of resources on implicit neural representations (2020), https://github.com/vsitzmann/awesome-implicit-representations, if this overview is useful to you, please cite as below
13. Sitzmann, V., Martel, J.N.P., Bergman, A.W., Lindell, D.B., Wetzstein, G.: Implicit Neural Representations with Periodic Activation Functions (Jun 2020), http://arxiv.org/abs/2006.09661, arXiv:2006.09661 [cs, eess]
14. Wang, Y., Xu, Y., Tsogkas, S., Bai, X., Dickinson, S., Siddiqi, K.: DeepFlux for Skeletons in the Wild (Nov 2018), http://arxiv.org/abs/1811.12608, arXiv:1811.12608 [cs]
15. Yang, B., Yao, J., Wang, B., Hu, J., Pan, Y., Pan, T., Wang, W., Guo, X.: P2MAT-NET: Learning medial axis transform from sparse point clouds. Computer Aided Geometric Design **80**, 101874 (Jun 2020). https://doi.org/10.1016/j.cagd.2020.101874, https://linkinghub.elsevier.com/retrieve/pii/S0167839620300613