# Abstract

In this paper, a model is presented for deformable image registration using an approach based on the VoxelMorph algorithm. This model is trained on the MNIST dataset as a lightweight demonstration of the convolutional approach to image registration.

# Introduction

Image registration is an important step in many medical imaging applications. Deformable image registration, where the shape of the image is allowed to be warped to achieve congruence between the initial and target images, is particularly useful in applications such as neuroimaging and organ segmentation, where images may represent cross-sections of volumes that shift in a 3-dimensional space. By mapping a deformation field from one image to another, annotated boundaries in one image can be mapped to their corresponding boundaries in different layers of a scanned volume, allowing for important clinical and research processes such as the 3-dimensional mapping of different organ systems, or the aggregation of multi-modal data into one aligned space.

Previous deterministic algorithms for deformable image registration have been slow and computationally intensive, limiting their practical clinical application. By training a convolutional neural network (CNN) to approximate the deformation field between pairs of images, this field can be generated significantly faster by the VoxelMorph algorithm than older methods. Efficiency improvements not only provide for more convenient application for the user, but also allow models to be applied practically on much larger datasets, and in use cases where long runtimes present a barrier to deployment.

In this paper, this approach is demonstrated using the MNIST dataset. This dataset, which is made up of grayscale images of handwritten integers from 0 to 9, is both freely available and relatively lightweight, making it ideal for training on consumer-grade hardware.

# Methods

## Dataset

The MNIST dataset is a freely-available set of 70,000 single-channel, 28 x 28 grayscale images of handwritten integers from 0 to 9, with 60,000 original test images and 10,000 test images. The original training split was further divided into 48,000 training and 12,000 validation images. For model training, the training and validation data were subset to 6,000 and 1,000 images respectively of only the number 7. A batch size of 179 was used in the training dataset, where each image in each batch was randomly matched to another image in the training data. Self-matching was allowed in order to allow the model to learn null mappings.

## Network Design

Given a fixed image **F** and a target image **T** in **R^2** from the MNIST dataset, we attempt to find a continuous vector field **phi** describing the deformation from **F** to **T**. The goal is for this vector field to maintain continuity in the deformation such that, when applied to medical imaging, anatomical structures in the fixed image are preserved in the reconstructed target image. Because this model is trained using the MNIST dataset, interpretability of the mapping is not a focus of this project.

To learn a function **f(F, phi) = T**, we use a CNN approach based on the VoxelMorph algorithm. Three convolutional layers are used to encode the image information and reduce the complexity of the represented dataset using max pooling. Three convolutional layers are then used to decode the encoded information via upsampling and convert it into a two-layer field representing the x- and y-axis deformation for each pixel in **F**. All layers were batch-normalized with a 20% dropout rate, and the hidden layers used a leaky ReLU activation function over 256 filters. The output layer used a tanh activation function to normalize the final field estimate.

## Model Training

The loss of the final output was calculated using a two-part function:

**Loss = sum( (f(F, phi) – T )^2 + || \grad{phi} ||^2 )**

The first loss term represents the mean-squared error associated with the accuracy of the reconstruction of the final image. The second term is used to reward smoothness in the deformation field, to ensure continuity, where **\grad{phi}** is approximated using finite differences. The two terms are weighted according a tuning parameter **lambda\_smooth**.

Parameters were learned using an Adam optimizer with L2 penalty **lambda** = 0.1. A grid search was performed to identify the optimal learning rate and **lambda\_smooth**. For both hyperparameter tuning and final training, the model was trained over 40 epochs.

Models were trained using a subset of the data that only included the number 7. The test loss was then generated for two scenarios, one for mapping instances of the number 7 to the number 7, and one for mapping the number 4 to the number 4 in order to determine how generalizable the model is beyond its initial training set.

## Contributions

Sam Albertson was responsible for all model design, training, and tuning, and for the final write up.

# Results

* Best hyperparameters
* Training vs. validation loss
* Test error on 7
* Test error on 4

# Discussion