

LANDMARK LOCALIZATION ON COLOR CODED DIFFUSION ANISOTROPY
IMAGES USING CONVOLUTIONAL NEURAL NETWORKS



by
Ahmet Emin Yetkin

Submitted to Graduate School of Natural and Applied Sciences
in Partial Fulfillment of the Requirements
for the Degree of Master of Science in
Biomedical Engineering

Yeditepe University
2019

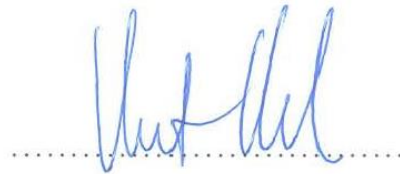
LANDMARK LOCALIZATION ON COLOR CODED DIFFUSION ANISOTROPY
IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

APPROVED BY:

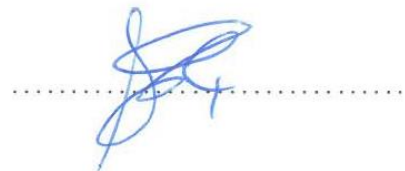
Assist. Prof. Dr. Andaç Hamamcı
(Thesis Supervisor)
(Yeditepe University)



Prof. Dr. Ali Ümit Keskin
(Yeditepe University)



Assist. Prof. Dr. Füsün Er
(Okan University)



DATE OF APPROVAL: / / 2019

ACKNOWLEDGEMENTS

This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) (Project: 116E407).



ABSTRACT

LANDMARK LOCALIZATION ON COLOR CODED DIFFUSION ANISOTROPY IMAGES USING CONVOLUTIONAL NEURAL NETWORKS

Landmark localization, finding exact location of structures in an image is a first stage of many complex computer vision problems. Locating specific landmarks on brain images is one of the stages in defining the target in functional surgery and in estimating point wise correspondence in image registration. Nowadays, various types of convolutional neural networks (CNN) have been proposed that are able to interpret complex computer vision problems. In this study, a CNN based landmark detector is employed to locate specific landmarks at given MNI coordinates, on an individual's diffusion MR brain images. MR diffusion images, with their high degree of heterogeneity, especially in white matter, provide a rich set of features compared to other basic structural images such as T1 or T2 weighted images. Results show that finding a specific point on brain using diffusion characteristics by CNN based model is sustainable and has a potential to be a base for image registration techniques.

ÖZET

RENK KODLU DİFÜZYON ANİSOTROPİ GÖRÜNTÜLERİNDE EVRİŞİMSEL SİNİR AĞLARI YORDAMIYLA İŞARETÇİ BULMA

İşaretçi tayini, yani belirli bir yapının görüntü içindeki kesin konumunu belirlemek bu çalışmanın amacı olan görüntü çakıştırma işlemleri gibi bir çok bilgisayarlı görü probleminin ilk safhasını oluşturmaktadır. Beyin görüntüleri üzerinde belirli işaretçileri saptamak, işlevsel cerrahide ve nokta bazlı görüntü çakıştırma işlemlerinde başarılması önemli hedefler arasında yer almaktadır. Günümüzde, karmaşık bilgisayarlı görü problemlerinde kullanılacak evrişimsel sinir ağ modelleri (ESA) öne geliştirilmektedir. Bu çalışmada, MNI koordinatları bilinen bir noktayı bireyin beyin difüzyon görüntüsünde tespit etmek için ESA tabanlı işaretçi bulucu sunulmuştur. Yüksek heterojeniteye sahip MR difüzyon görüntüleri, özellikle beyaz maddede T1 veya T2 ağırlıklı MR görüntüleri gibi diğer yapısal görüntüleme yöntemlerine göre daha zengin nitelikler sunmaktadır. Sonuçlar göstermektedir ki, beyinde yer alan belirli noktaları bulmak için beynin difüzyon karakteristiğini ESA bazlı bir yöntemle bulmak sürdürülebilirdir ve görüntü çakıştırma yöntemleri için temel oluşturacak potansiyele sahiptir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	ii
ABSTRACT.....	iv
ÖZET	v
LIST OF FIGURES	viii
LIST OF TABLES	xi
LIST OF SYMBOLS/ABBREVIATIONS.....	xii
1. INTRODUCTION	1
1.1. MOTIVATION	1
1.2. CONTRIBUTIONS.....	3
2. BACKGROUND.....	4
2.1. DIFFUSION MR IMAGES	4
2.2. ARTIFICIAL NEURAL NETWORKS	6
2.2.1. Perceptrons.....	8
2.2.2. Activation Functions.....	9
2.2.3. Loss Function.....	11
2.2.4. Back Propagation.....	11
2.2.5. Optimization	13
2.2.6. Deep Learning.....	14
2.2.7. Deep Feed-Forward Neural Networks.....	15
2.3. CONVOLUTIONAL NEURAL NETWORKS	17
2.3.1. Types of Layers	20
3. MATERIALS AND METHODS	24
3.1. CNN STRUCTURE AND TRAINING	24
3.2. TESTING AND EVALUATIONS	27
3.2.1. HCP Dataset.....	27
3.2.2. Clinical Dataset.....	28
4. RESULTS.....	30
4.1. RESULTS ON HCP DATA.....	30

4.2. RESULTS ON CLINICAL DATA.....	33
5. DISCUSSION.....	45
6. CONCLUSION	48
REFERENCES	50
APPENDIX A.....	53



LIST OF FIGURES

Figure 1.1. a) Person's camera image, b) 3D model obtained from his MRI.....	2
Figure 1.2. Comparison of T1WI and DTI based Color map	3
Figure 2.1. Ellipsoid model for anisotropic diffusion.....	5
Figure 2.2. (a) FA vs (b) Color FA	6
Figure 2.3. Structure of a neuron(left), Structure of artificial neuron (right)	7
Figure 2.4. Artificial neural network with 1 hidden layer	8
Figure 2.5. Sigmoid function	10
Figure 2.6. Relu vs elu	10
Figure 2.7. Effect of a change on a weight to next layers.....	14
Figure 2.8. Standard gradient descent.....	14
Figure 2.9. Venn diagram of relations between selected AI approaches.....	15
Figure 2.10. Implementation of convolution operation in images	18
Figure 2.11. Representation of neurons in CNN	19
Figure 2.12. LeNet-5 architecture.	19
Figure 2.13. Pooling operation	21

Figure 2.14. Up-sampling operation.....	22
Figure 3.1. Sample of input data and label heatmap that shows position of selected landmark	25
Figure 3.2. CNN layer structure.....	26
Figure 3.3. CNN prediction that shows success all of the patients.....	29
Figure 3.3. CNN prediction that shows success none of the patients.....	29
Figure 4.1. Average success rate for each landmark on glass brain template	30
Figure 4.2. Average success rate for each landmark on glass brain template	30
Figure 4.3. Regional average distance between FNIRT and CNN predictions	31
Figure 4.4. Average distances of FNIRT and CNN predictions on glass brain template ...	31
Figure 4.5. Regional average differences of OVL Scores	32
Figure 4.6. Average difference between the OVL Scores on glass brain template	32
Figure 4.7. Predicted points by CNNs over 7 patient data for Landmark nu. 01-10.....	34
Figure 4.8. Predicted points by CNNs over 7 patient data for Landmark nu. 11-20.....	35
Figure 4.9. Predicted points by CNNs over 7 patient data for Landmark nu. 21-30.....	36
Figure 4.10. Predicted points by CNNs over 7 patient data for Landmark nu. 31-40.....	37
Figure 4.11. Predicted points by CNNs over 7 patient data for Landmark nu. 41-50.....	38

Figure 4.12. Predicted points by CNNs over 7 patient data for Landmark nu. 51-60..... 39

Figure 4.13. Predicted points of CNNs over 7 patient data for Landmark nu. 61-70..... 40

Figure 4.14. Predicted points of CNNs over 7 patient data for Landmark nu. 71-80..... 41

Figure 4.15. Predicted points of CNNs over 7 patient data for Landmark nu. 81-90..... 42

Figure 4.16. Regional averages of success rates of clinical data..... 43



LIST OF TABLES

Table 3.1. The regional distribution of landmarks.....24

Table 4.1. Average and standard deviations of the evaluation results based on regions.....33

Table 4.2. Average results based on regions of clinical data.....44



LIST OF SYMBOLS/ABBREVIATIONS

x	Neural network input
y	Target output
α	Learning rate
θ	Neural network parameters
ANN	Artificial neural network
API	Application programming interface
CNN	Convolutional neural network
CPU	Central processing unit
DTI	Diffusion tensor imaging
FA	Fractional anisotropy
GPU	Graphical processing unit
HCP	Human connectome project
MLP	Multilayer perceptron
MRI	Magnetic resonance imaging
SGD	Stochastic gradient descent
VTK	Visualization toolkit

1. INTRODUCTION

Landmark localization, finding the exact location of structures in an image is the first stage of many complex computer vision problems, including registration [1], segmentation [2], recognition[3], pose estimation [4, 5], object localization [6], and many more specialized problems. Landmark detection and localization can be helpful in clinical applications and/or provide possibilities to other research projects [7, 8]. Locating specific landmarks on brain images is one of the stages in defining the target in functional surgery and in estimating pointwise correspondence in image registration [9, 10].

The human visual system is marvellous. It can recognize, classify, interpret objects within less than a second. Even though this procedure seems effortlessly to us, it can be very hard to express it to recognize an object in an algorithmic way since it's hard to make precise rules that work both under special and general circumstances. The artificial neural network is one of many methods that has been proposed over years to overcome these common challenges. When looking back to last two decades of related literature, it's the most promising approach solving complex computer vision problems.

1.1. MOTIVATION

In pose estimation, detection, or recognition procedures, locations of landmarks are useful since it enables highlighting key features of an image, and making easier to understand data and patterns of it. Likewise, in registration, determining locations of related landmarks can be used in space normalization and linear transformation processes which are prior conditions to nonlinear image registration processes. Manual landmark annotation is a time consuming, labour-intensive, prone to false marking method that leads the system to fatal flow, inconsistent and expensive method [7].

In image-guided surgery, inferring pose of the body to register pre-and intra-interventional data is a well-studied problem [11]. The 2D-3D registration process includes merging 3D image modalities such as MRI, CT with 2D image modalities such as ultrasound, x-ray, optic camera images into the same coordinate system. As inter step, detecting and locating mutual landmarks is a preferable method.

Although the problem is too complex on a global scale without prior knowledge on the application domain, the CNN model could detect mutual landmarks from both domains [12]. As specified in [12], the problem of registering one's head MR volume to planar camera as shown in Figure 1.1 images is addressed, motivated by the incision planning in neurosurgery [13]. Locating landmarks in both spaces could serve this purpose, especially, when annotating data from different domains could be expensive. On the other hand, domain generalization, which in this case refers to learning landmarks on a domain which annotation has lower costs, and applying it to another domain which annotation is not available or has high cost of time or source.

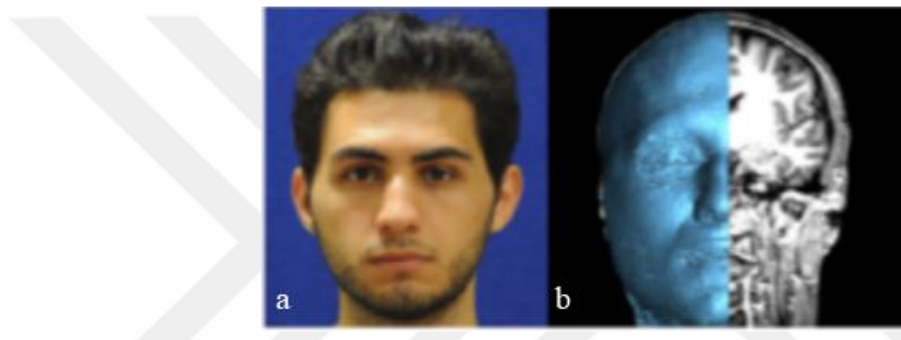


Figure 1.1. a) Person's camera image, b) 3D model obtained from his MRI [12]

Thus, working on different imaging modalities is not the only case. While registering medical images from same imaging modality, finding specific points on both domains is also necessity. It is possible to estimate the rigid transformation if a sufficient number of such landmark correspondences could be reliably detected.

MR diffusion images, with their high degree of heterogeneity, especially in white matter, provide a rich set of features compared to other basic structural images. In other structural imaging methods such as T1 or T2 weighted images, they mainly focus on protons (1H) in water molecules (H_2O). Since molecular structure of white matter causes homogenous view as seen in Fig 1.2. Diffusion images carry more information about fibers and their orientations than T1 weighted MR images.

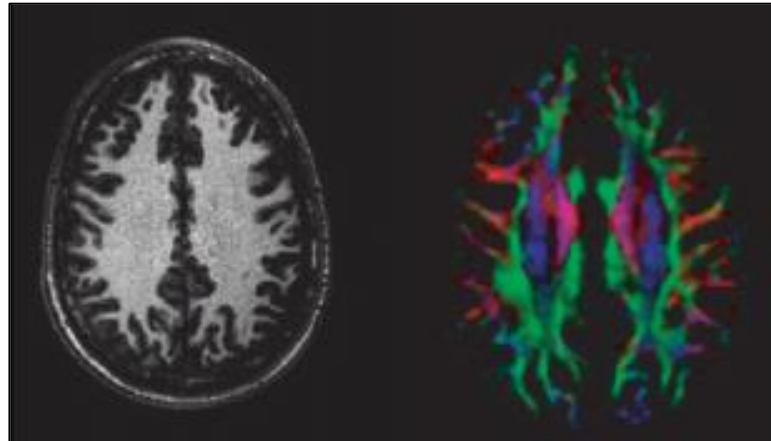


Figure 1.2. Comparison of T1WI and DTI based Color map [14].

Diffusion tensor images are successfully implemented as the image term in the deformable registration framework [15]. Those methods seek a solution on a local scale for the correspondence of the tissue microstructure with a high degree of regularization. In [9], a method based on white matter fiber connection patterns derived from diffusion tensor imaging data is proposed to predict cortical landmarks, which are named as DICCOLs, in a new single brain.

1.2. CONTRIBUTIONS

In this dissertation, a convolutional neural network-based landmark localization method is proposed and its application areas are discussed. To the best of our knowledge, it's the first study that aims to detect and locate landmarks on color coded diffusion images.

This study mainly focuses on importing and adapting achievements from heatmap producing CNN models on feature localization related computer vision problems [6, 30]. The proposed CNN model is a result of adaption of learning outcomes from hourglass alike CNN model which was proposed for human pose estimation [30] to medical landmark localization by regressing heatmaps [8]. All convolution, pooling, and upsampling operations take place in 3D and learn features from 4-dimensional tensor patches from color FA maps.

2. BACKGROUND

2.1. DIFFUSION MR IMAGES

Diffusion tensor imaging is a non-invasive imaging method that takes reference as motion of water (diffusion) molecules which enables to estimate the location and orientation of the target structure, such as white matter tracts, or muscle fibers. There are detectable signal losses in voxels where diffusion take place caused by acquisition of random phase spins. These signal losses would be less in some regions where diffusion is restricted. Diffusion weighted images are acquired by applying special diffusion encoding gradients to map the diffusion. These motions can be measured along any axis desired. Opportunity of measuring diffusion along any axis that is important since prior knowledge of fiber anatomy is not always available, especially in brain. Nonetheless, freely diffusing water is not meaningful alone since it's homogeneous, almost same in any direction. This type of diffusion is called isotropic diffusion. In this scenario, only one parameter is enough to describe diffusion, which would be a diameter of sphere. On the other hand, in biological tissues such as muscles or brain, water motion is inclined along certain directions due to restrictions by biological structures. This type of movement is called anisotropic diffusion and mostly observed in structures called fibers. As agreed, anisotropic diffusion is more biologically relevant and carries more information. Theoretically, fibers must align to the orientation which has the biggest diffusion constant. It's harder to describe anisotropic diffusion than isotropic diffusion since its shape is related with ellipsoid in 3D. In 2D plane, anisotropic diffusion is observed as an oval. Obviously, there are more parameters needed to express this kind of motion. An oval requires 3 parameters to be expressed properly which are length of longest and shortest axes, and orientation of the shape. In 3D, this number jumps to 6.

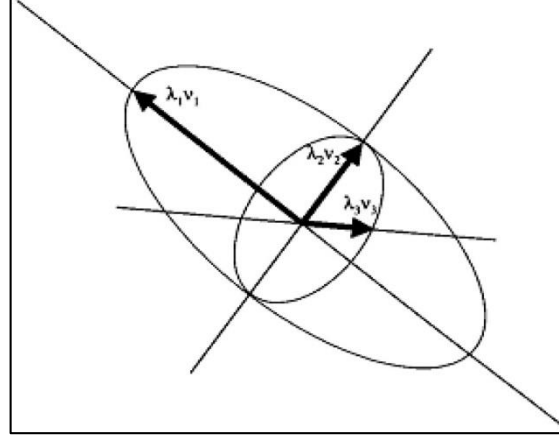


Figure 2.1. Ellipsoid model for anisotropic diffusion

Since D has 6 independent components, there has to be at least 6 independent measurements to be acquired. Diffusion tensor is defined as in equation 2.1.

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} \quad (2.1)$$

where D_{xx}, D_{yy}, D_{zz} are diffusion coefficients measured along axes, and other six terms represent motions between each pair of directions. D is a symmetric tensor. That means, $D_{ij} = D_{ji}$. The eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ and the eigenvectors e_1, e_2, e_3 are used to define D of ellipsoid that are obtained from diagonalization of D as in equation 2.2.

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} = [e_1 \ e_2 \ e_3]^T \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} [e_1 \ e_2 \ e_3] \quad (2.2)$$

Another parameter that is derived from these values is fractional anisotropy (FA) coefficient which is given in equation 2.3. The FA value has a range of 0 to 1. It's value is 0 when diffusion is isotropic, and 1 when diffusion is strongly anisotropic.

$$FA = \frac{1}{2} \frac{\sqrt{((\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2)}}{\sqrt{(\lambda_1^2 + \lambda_2^2 + \lambda_3^2)}} \quad (2.3)$$

Using only FA maps provides grayscale images which carries certain informations about fibers, yet it can be enhanced.

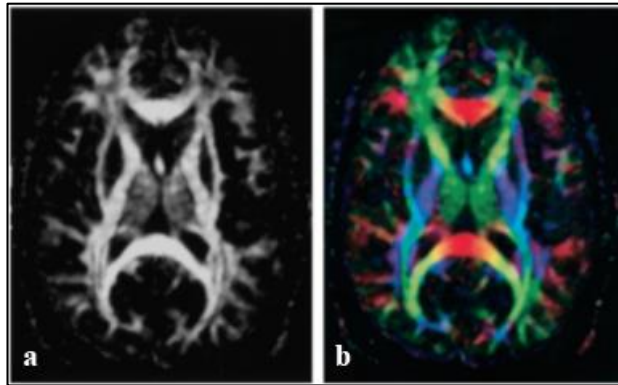


Figure 2.2. (a) FA vs (b) Color FA [14]

The eigenvector e_1 , the principal eigenvector, provides an estimate of the fiber direction. Therefore FA weighted e_1 maps are used for assigning fiber orientations colors for better visualization. In Figure 2.2, A) is FA map whereas B) is color FA map. Each color indicates different direction. Red is left-right, green is anterior-posterior, blue is superior-inferior.

2.2. ARTIFICIAL NEURAL NETWORKS

In this chapter, methods to be used for landmark localization are explained. Artificial Neural Network, or Neural Network which is a type of machine learning method, is establishing the backbone of our works.

Artificial Neural Network is more than an algorithm, it's a framework that can be used while working with complex data and problems without being explicitly programmed for a specific task. ANN finds one of the best possible approximation. As in equation 2.4, It maps an input x to target y by learning parameters θ with a help of hyper-parameters μ [27]. Hyper-parameters are non-learnable parameters that are determined by user before training.

$$y = f(x; \theta; \mu) \quad (2.4)$$

As mentioned above, we have x and y pairs in our dataset. Finding relation between these two of them is supervised learning. Feed-forward Neural networks are mostly used within the context of supervised learning. However, it's possible to train neural network models in context of unsupervised learning. The most popular example of unsupervised learning of neural networks would be autoencoders. Yet, unsupervised learning methods lie beyond the scope of this study.

Artificial Neural Networks are inspired by signal transmission through neuron cells of living organisms [16]. A Neuron cell and an artificial neuron can be compared briefly by looking in Figure 2.3. The 2 major concepts that made foundations of Neural Networks are Threshold Mechanism and Hebbian Learning [16] which is a learning model that takes basis as neural plasticity. Thresholds in neurons produce discrete outputs that are cell is fired or cell is not fired when input with continuous range coming from receptors and/or other neurons. In other words, there is a mechanism that can be modelled as step function. This is an activation function of neuron. In addition, inputs are weighted before going through activation function.

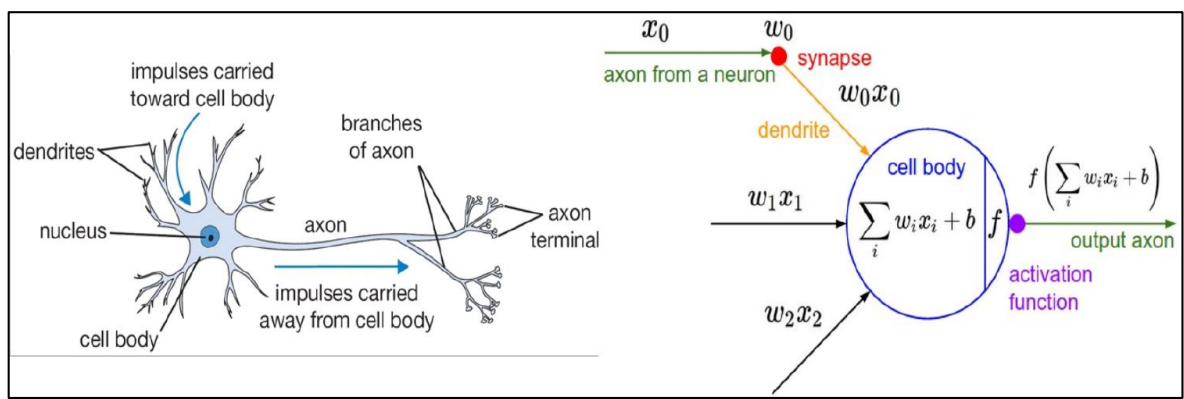


Figure 2.3. Structure of a neuron(left), Structure of artificial neuron (right) [27]

A weight is assigned to each neuron connection. Weight between two neurons represents strength of the connection. Information can be restored using this weights. Learning takes places when these weights get updated and create patterns as a similar approach to plasticity that occurs in our nervous system. An adult human brain contains about 10^{11} neurons and 10^{14} connections of neurons called synapses. It's a good criterion to compare human intelligence with ANN.

A standard feed-forward artificial neural network is composed of layers of neurons that are connected to each other in a certain way so that data x flows through one way and generate y . Neurons in the same layer are not connected to each other. As in Figure 2.4, a neuron makes connections with neurons in the next and previous layers. Each of the neurons gets activated in terms of weight and bias values that are assigned to them in the first place. Achieving the task that is given to Neural networks is highly dependent on number of layers and neurons, and weight and bias values of neurons.

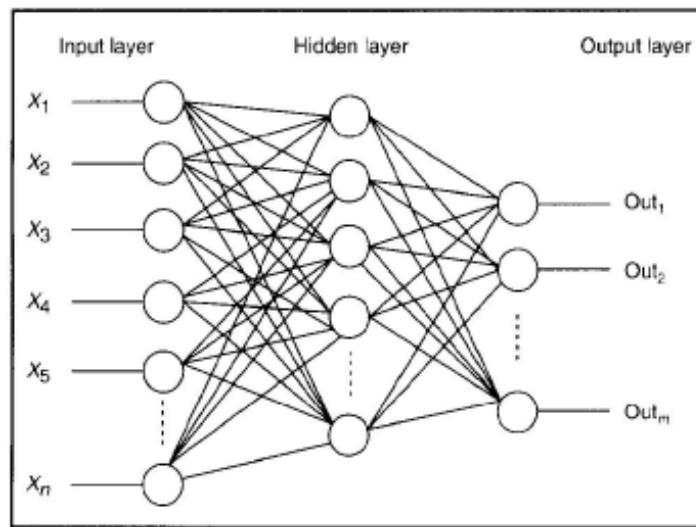


Figure 2.4. Artificial neural network with 1 hidden layer

2.2.1. Perceptrons

Perceptron is a linear classifier. It's an early steps of comprehensive neural networks that are used nowadays. A perceptron takes several values as an input and produces single binary output in terms of response of activation function to weighted input. Activation function takes weighted inputs as an input and passes its output to the neurons of the next layer. In order to increase capacity of system, modelling neurons with more complex activation function rather than step function with constant threshold value would be more efficient and more realistic. It can be nonlinear functions such as sigmoid, tanh or a linear function such as relu (rectifier linear unit).

$$f(x) = \begin{cases} 1, & w \cdot x + b > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2.5)$$

Let's consider activation function is step function with threshold value of 0. Its output is identified as 1, if the sum of the multiplication of inputs and their assigned weights is higher than threshold, and 0 otherwise as in equation 2.5, where w is weights, x input, b bias of neuron.

Perceptrons offer practical and interpretable decision making system. Expressiveness of perceptrons is promising yet limited [17]. In order to measure what can we do with

perceptrons, modelling logic gates would be effective approach. Although there isn't any problem expressing NOT, AND, OR NAND, NOR gates, XOR gate can't be expressed with single layer perceptron. This indicates that, if classes are not linearly separable, single layer perceptron can't separate. So single layer perceptron has certain capacity and to express some functions, using more than single layer perceptron is necessary. Multi-layer perceptrons are perceptrons with hidden layers as seen in Figure 2.4. That means, there are mid layers between input and output layers. Increasing layer size also increases the capacity of expressiveness of neural network. Nevertheless, not every problem and its solution are expressed in binary. Making the system produce output with continuous range rather than certain binary result could solve it. Step function that has certain threshold value can be replaced with some other activation function, such as sigmoid, relu or relu variations. Therefore, output also has a continuous range.

2.2.2. Activation Functions

Activation functions determines the range of output. There are nonlinear activation functions such as sigmoid as in equation 2.6 and Figure 2.5 that squashes their output to certain range that is (0, 1) for sigmoid. relu, (rectifier linear unit) as in equation 2.8 and Figure 2.6 (red line) sets zero if input value is smaller than 0, and produces same value with input if input is bigger than 0. That makes its range $[0, \infty)$. Derivation of sigmoid function is given in equation 2.7 and derivation of relu is given in equation 2.9.

$$S(z) = \frac{1}{(1 + e^{-z})} \quad (2.6)$$

$$S'(z) = S(z) \cdot (1 - S(z)) \quad (2.7)$$

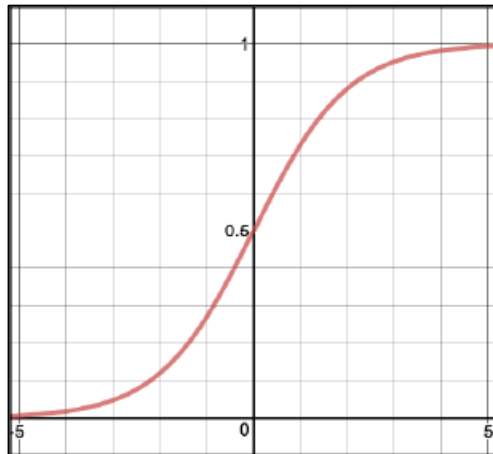


Figure 2.5 Sigmoid function

$$R(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases} \quad (2.8)$$

$$R'(z) = \begin{cases} 1, & z > 0 \\ 0, & z < 0 \end{cases} \quad (2.9)$$

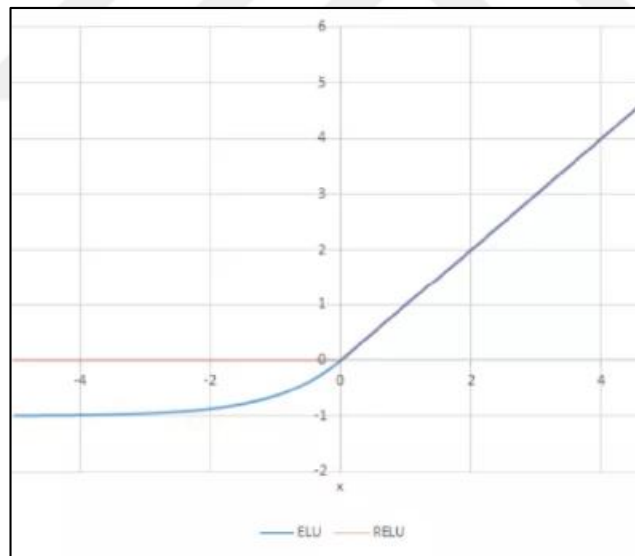


Figure 2.6. Relu vs elu

Characteristics of derivative of activation functions have important role since they are used in error calculation and backpropagation, derivative. Although nonlinear functions could be good classifiers than step function, they have a significant limitation that when number of layers increases, outputs start to pile up limits of the function. Big changes on the x is not observed on y . This problem is called as vanishing gradients. On the other hand, activation

functions with linear characteristics such as relu, don't have a problem like this since it sets output to input value as long as it is positive. It's proved that this approach is more efficient in multi-layer architectures.

$$L(z) = \begin{cases} z, & z > 0 \\ \alpha \cdot (e^z - 1), & z \leq 0 \end{cases} \quad (2.10)$$

$$L'(z) = \begin{cases} z, & z > 0 \\ \alpha \cdot e^z, & z < 0 \end{cases} \quad (2.11)$$

Another activation function is elu [18] as in equation 2.10 and Figure 2.6 (blue line) where α is a pre-defined constant coefficient. Actually, It's an upgraded version of relu. It aims performance increase by adjusting behaviour of relu on negative values. It's proposed that assigning proportionally small values rather than just setting all negative values to zero makes gradients more effectively distributed.

2.2.3. Loss Function

Loss function, or cost function is a function which is desired to be minimized, indicates how good is the result of the system by comparing the result with the desired result. Loss function evaluates the output and reduces it to a number which is called error. There several loss functions to be used for neural networks nowadays. In equation 2.12 squared error loss function is given. Type of problem has a determining role on choosing loss function. Cross entropy can be used on binary or multi-class classification problems whereas variations of squared error can be used on regression problems

$$Error = \frac{1}{2} \sum_i (y_i - h(w)_i)^2 \quad (2.12)$$

2.2.4. Back Propagation

Neurons learn by adjusting its weights. There are 2 essential procedures that adjust weights that are forward pass and backward pass. In forward pass, weights are initialized, an output

is produced. The error is calculated between output which is produced by neural network and ground truth in terms of loss function.

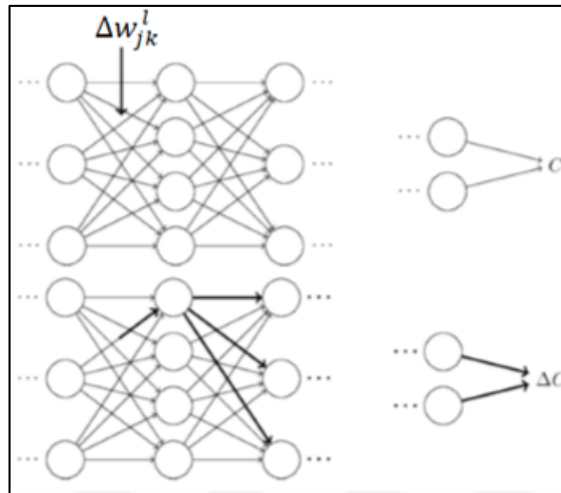


Figure 2.7. Effect of a change on a weight to next layers

Derivative of the error function tells us whether error is increasing or decreasing. Backward pass is about propagating calculated error with respect to output of last layer back to the rest of network till the first layer using gradient descent so that all weights and bias values can be updated properly [19]. A concise explanation to backpropagation would be its practical implementation of chain rule of derivatives to calculate the gradient of the loss function in terms of learnable parameters which are weights and biases as in Figure 2.7. That leads to contribution of each parameter to loss as in algorithm 3.1 where l is number of layer, σ is related activation function, w is weight, b is bias of a neuron, $\nabla_a C$ is the rate of change of C w.r.t. the output activations.

Algorithm 3.1. Backpropagation

1. Input x : samples from training dataset
2. Compute in feed forward : $z^l = w^l a^{l-1} + b^l$, $a^l = \sigma(z^l)$ for each $l = 2, 3, \dots, L$
3. Calculate output error : $\delta^L = \nabla_a C \cdot \sigma'(z^L)$
for l **from** $L-1$ **to** 2
 compute : $\delta^l = ((w^{l+1})\delta^{l+1}) \cdot \sigma'(z^l)$
end for
Output : The gradient of cost function is given by $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$ and $\frac{\partial C}{\partial b_j^l} = \delta_j^l$

Afterwards, optimizers take place to update these values to minimize loss.

2.2.5. Optimization

Optimization algorithms are used to minimize loss (another name for error function output) through gradients that were calculated by backpropagation. Computed output values will be updated in the direction of optimal solution. There are 2 major categories of optimization algorithms which are first order and second order optimization algorithms.

The second order optimization algorithms use second order derivative. The second order derivative tells increment or decrement in first order derivative which leads to an opinion about curvature of function. Nonetheless, it is not used as much as first order optimization algorithms since its cost is higher although it may outperform first order optimization algorithms.

In first order optimization algorithms use gradient values of target function. The first order derivative tells if the function increases or decreases at a specific point. A gradient shows the rate of change on a specific direction of function [20]. It can be said that gradient is generalized state of derivative for multi-variable dependent functions. The most popular first order optimization algorithm is gradient descent.

$$\theta \leftarrow \theta - \alpha \cdot \nabla J(\theta) \quad (2.13)$$

Gradient descent plays crucial role on finding the minimal point of loss function. As in equation 2.13, it updates parameters θ by gradient of loss functions, where α is the learning rate that penalizes the rate of change in order to prevent making radical changes on parameters. If learning rate is so small it will take longer for system to converge, meanwhile if learning rate is large, it will probably overshoot the global minima over and over again.

Since gradient descent calculates gradient of whole dataset for 1 update it would be very slow process and its computational cost can't be afforded always since the memory may be inadequate for entire dataset.

$$\theta \leftarrow \theta - \alpha \cdot \nabla J(\theta; x(i); y(i)) \quad (2.14)$$

As in equation 2.14 where $x(i)$ and $y(i)$ are examples from training data, Stochastic gradient descent (SGD) is more suitable for larger datasets. Per one update, one training example is enough for SGD. Most of the time it's faster than gradient descent. Another thing about SGD is frequent updates. It makes parameters have high variance that causes fluctuations in loss. There's an approach called mini-batch gradient descent. It uses n data points where $n > 1$. There are several ways that is proposed by Geoffrey Hinton to improve optimization algorithms in one of his courses over Coursera, such as using momentum, adaptive learning rate. As an example RMSprop is introduced [21]. RMSProp is a gradient based optimization algorithm that uses momentum technique to converge with an that during a calculation of a weight, it divides learning rate by a magnitude of gradients for that weight.

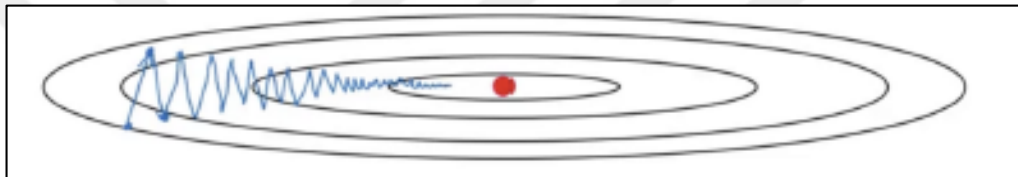


Figure 2.8. Standard gradient descent

As seen in Figure 2.8 there is a difference between step sizes between directions. The main idea behind momentum is to use gradient to use velocity of an rolling object on a surface with curvatures rather than its position. If incline gets steeper on direction that object goes, momentum helps to adjust velocity. Momentum helps to prevent oscillation by restricting steps taken in y direction. This also gives a chance to choose higher learning rates.

2.2.6. Deep Learning

Deep Learning, which is a concept that enables us to design neural networks with enhanced learning capacity by adding numerous more layers is a specialized machine learning area that is aimed to overcome near human-level or above human-level intelligence required tasks by machines. This idea is almost as old as artificial neural networks, yet it had different names over the years, with different philosophical aspects. In Figure 2.9, the relationship between deep learning methods and other AI approaches is shown.

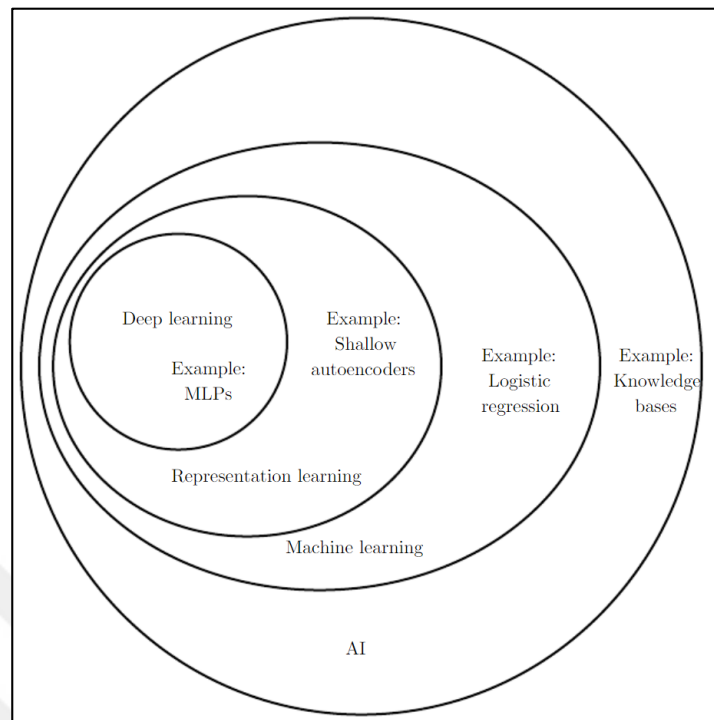


Figure 2.9. Venn diagram of relations between selected AI approaches [27].

Rise of deep learning started at the beginning of the 2000s. The concept of deep learning includes regulations and training methods that are explained in section 2.3. On the other hand, the computational power of machines had reached a certain level that enabled theoretical knowledge that made training deeper neural networks (neural networks with more layers) possible transferred into practice. The use of GPU in neural network training is worth mentioning since it remarkably shortened the training periods. Also, not only hardware also software infrastructure has a role. User-friendly frameworks written on high-level programming languages have become very popular and been supported by large communities. Support of large crowds had lead deep learning to find itself more area of use as the number of available training dataset has become available.

2.2.7. Deep Feed-Forward Neural Networks

Adding more layers and expecting consistent and accurate results are not realistic. The more layers are added to ANN, the harder it gets to converge. Because backpropagation algorithm might not find global minimum and confuses one of the local minimum points with global minimum. This possibility is directly proportional to number of parameters, thus number of

layers. When using only backpropagation came short on deep feed forward networks, new challenge is born. There are few significant approaches that are proposed.

One of them is greedy layer-wise training which indicates each layer must be trained separately [22]. In other words, each layer is treated as output layer one at a time. After getting meaningful weights comparing to random weights, all layers are merged. Therefore ANN is more determinant.

As mentioned above, one of the significant developments is using linear activation functions to prevent vanishing gradients problem.

First thing that is expected from neural network (NN) is to make correct predictions. Yet, it's not enough by itself. What really matters is making predictions by learning patterns in the training data, rather than memorizing answers. In order to train neural network, there are requirements to fulfil. There are some adjustments on hyper-parameters, such as setting the right activation functions. These changes should be done by considering response of neural network to data and task. Before that, training dataset must be inspected. It's as important as neural network design. If dataset has high variance, NN may likely underfit, which means NN are hesitant among possibilities. It can be observed as fluctuations in loss graph during training and low success ratio over validation dataset. High variance in dataset is not the only reason of underfitting. If NN doesn't have enough parameters to express the target function, it may not converge. If dataset has high bias, in other words, if there are lots of examples of a certain case, NN might choose to memorize these examples. It's called as overfitting. It can be observed as high accuracy on training set, whereas these accuracy doesn't show up on examples that NN hasn't seen before, such as validation set. Having more parameters than needed may also lead up to overfitting. Another efficient way to prevent overfitting is dropout [23]. It's a regulation that's applied to neurons that involves setting weights of a neurons to zero with a certain probability. Therefore, if a neuron has high weight values it would be filtered by setting its weight to zero. If it was representing something important, it would most likely be set to a similar value. But if it was set by chance or a result of an imbalance in data, it would be set to lower value as it should be.

Therefore, training dataset must be normalized, standardized. An effective way of normalization is batch normalization [35]. It maintains the activations of layers of NN in a certain range. Standardization of outputs increases total performance.

2.3. CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) are not so different than ordinary neural networks. They have neurons with learn-able parameters called weights and biases. They have also activation functions. They also have loss function like regular neural networks do. Thus CNNs are good at focusing on local structures than regular neural networks. CNNs are suitable for the data that has spatial arrangement.

Let's define convolution operation first as the way it is used in the context of Convolutional Neural Networks. Convolution is a response of a function when another function passes through it. It's a mathematical method to combine two different signals to generate a new signal [26]. In equation 2.15 spatial convolution formula in discrete form is given.

$$f[x, y] * g[x, y] = \sum_{-\infty}^{+\infty} \sum_{-\infty}^{+\infty} f[n_1, n_2] \cdot g[x - n_1, y - n_2] \quad (2.15)$$

Convolution is a linear operation. In fact, it's composed of multiplication and addition. In the image, the first signal is input data, and the second signal is the convolution kernel or convolution filter. Elements of kernels are considered as neurons. A kernel is slid over an image. Centre element of kernel is placed over each pixel in the image. Value of weighted sum of centred pixel and its neighbours is assigned to one of the pixels of the output array that spatially corresponds to centred pixel as seen in Figure 2.10. Size of the convolution kernel determines local receptive field, which means how far neighbours are to be considered. With increasing size of kernel increases accuracy of operation, but also increases the computational cost.

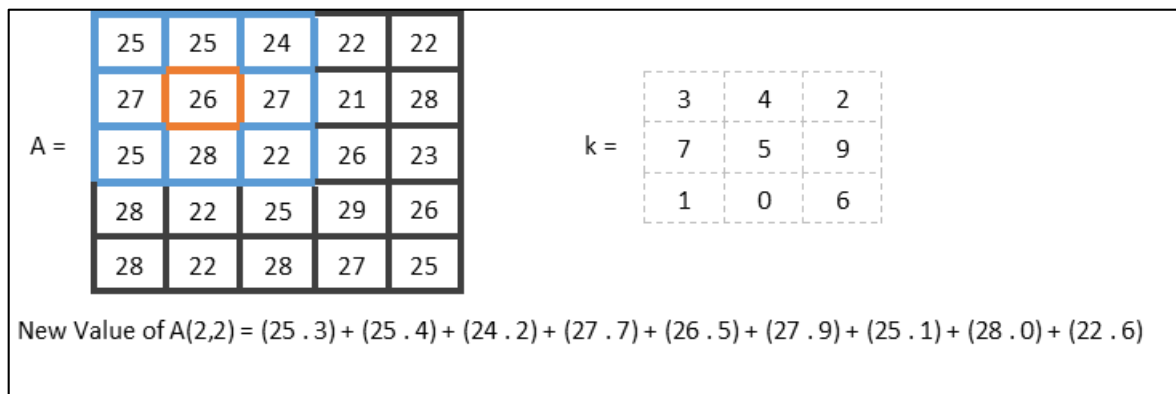


Figure 2.10. Implementation of convolution operation in images

Regular feed forward neural networks is not suited for multidimensional data such as images. For example, for a tiny image size of $28 \times 28 \times 3$ (width, height, color channels), $28 \times 28 \times 3 = 2352$ weights needed for first layer of neural network(NN). Although it is not that high, when size of image is more relatable such as $256 \times 256 \times 3$, that means $256 \times 256 \times 3 = 196608$ weights, just for first layer. Besides, many of these connections are redundant. Images are composed of local structures. There is no need to look at all the pixels of the image at once to inform about an object that holds relatively small percentage of place in the image.

There are certain key concepts that brings CNN forward, which are sparse interactions, parameter sharing, and equivariant representations [27]. Sparse interactions occur when kernel is smaller than image, which usually is. It's enough a feature to appear once in the image in somewhere, rather than dominating structure in the image, for CNN to activate. Parameter sharing is using the same parameters over and over again. Equivariant representation refers to changing in the input affects the output in the same way. It must be said, Convolutional layers have equivariant at some level. If data has significant transformations, it requires another manoeuvres.

Using same neurons over and over again as in Figure 2.7 is not only computationally efficient, it also works better than regular NNs. Starting with LeNet-5 that made itself mentioned by recognizing handwritten digits [24], and followed by a huge success of AlexNet on ImageNet classification [25], convolutional neural networks (CNN) have been showing great success in computer vision tasks and outperformed earlier methods for two decades. Although LeNet-5 achieved human level accuracy on recognizing handwritten digits, it's a simpler task than general purpose object classification. On the other hand, AlexNet showed CNNs can be also useful in large scale problems such as classifying an object among 1000 classes in high resolution images. It proved we hadn't reached the limit of learning capacity of convolutional neural networks.

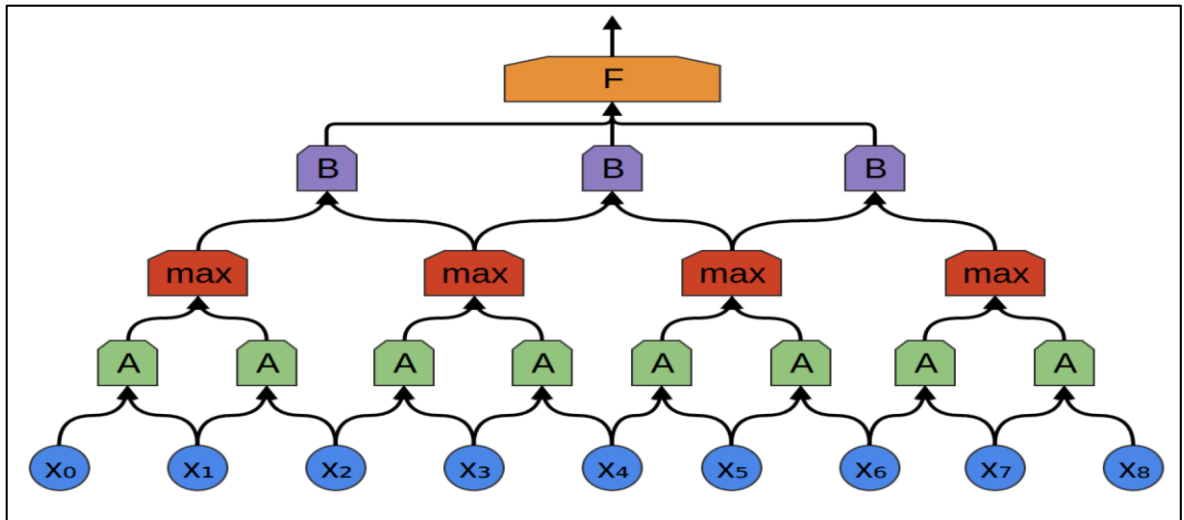


Figure 2.11. Representation of neurons in CNN.

In Figure 2.11, representation of neurons is shown where x_n is input layer, **A** and **B** are convolutional layers, **max** is max pooling layer, and **F** is fully connected layer which is responsible for decision process.

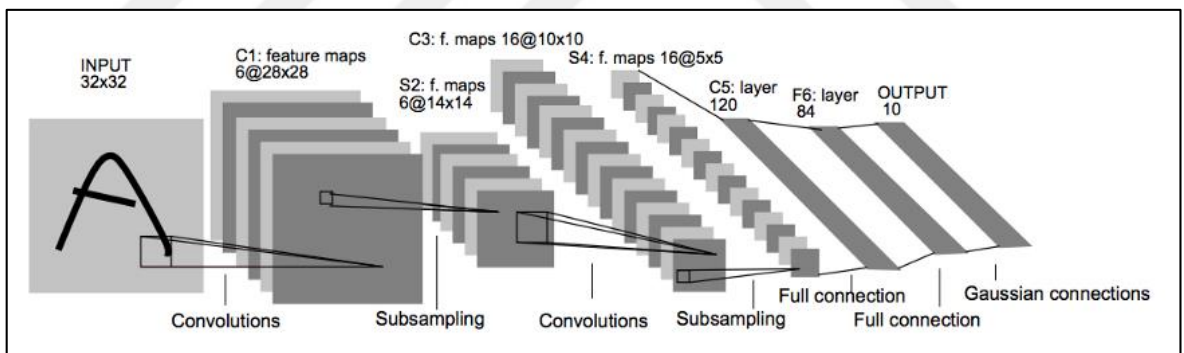


Figure 2.12. LeNet-5 Architecture[24]

In Figure 2.12, architecture of Lenet-5 is shown. It takes an input, convolves by 5×5 kernel. As a result of convolution operation, size of the output is 28×28 . Dimension calculation is given in equation 2.16. After convolution, pooling operation takes places where it down-samples its input from 28×28 to 14×14 . After pooling, feature maps are convolved with 5×5 kernels and different feature maps with 10×10 are generated. After another pooling, output is reshaped so it could be sent to fully connected neural network for classification process. Convolution and pooling operations are used for extracting features from data. The role of pooling operation is reducing number of parameters which also means reducing

computational cost. With pooling, exact location of a feature is also lost. Classifier makes decisions based on features, not where exactly is in the input image.

There are several practical applications that getting more performance on CNN such as data augmentation. Augmenting the dataset is quite popular method make CNN more robust to noise or variances. Data augmentation is a process that generating new samples from dataset by making small changes on original samples from original data. While working with images, changing resolution of image, flipping, cropping, rolling, rotating, translating an image could be useful. The idea behind this procedure is, an object can be perceived different under different conditions by machines. The objective of data augmentation is to regulate response of neurons when encountering different forms of objects in the image [28].

Yet, not every computer vision problem can be represented as a classification problem. In biomedical imaging, it's desired to be more specific. Usually, the aim is to identify each basic unit which can be pixel or voxel since a single one of them might make a difference, for which the traditional convolutional networks designed for simple classification may not satisfy these kinds of expectations. Nowadays, various types of CNNs have been proposed that are able to interpret so many complex problems. Pixel-based localization of an object, landmark or a structure is one of them.

2.3.1. Types of Layers

2.3.1.1. Convolutional Layers

A convolutional layer is an essential layer for CNN where convolution operation takes place. Its basic hyper-parameters to be set are the size of kernels, number of these kernels, stride value, behaviour on edges of the array, type of activation function. What behaviour on the edges meant is, as a result of convolution, there is a reduction in the size of the array. As a practical application is to pad the convoluted array to keep the size of the array fixed. Therefore, not only array size calculations, but also making the array to its initial size could be easier since a symmetrical structure can be created by simply switching pooling layers with up-sampling layers. Kernel size determines how far and along which dimensions to consider to determine a new value of relevant array element.

$$O = \left(\frac{I - K + 2P}{S} \right) + 1 \quad (2.16)$$

Equation 2.16 denotes how to calculate output size **O** after convolution operation where **I** input, **K** kernel size, **P** amount of padding and **S** is amount of stride. In addition, third dimension is related to number of kernels since a different feature map is generated for each kernel.

2.3.1.2. Pooling and Up-Sample Layers

A pooling layer takes several parameters in a certain area as an input and returns a single value as seen in Figure 2.13. This value can be either an average of inputs (average pooling) or the maximum value of inputs (max pooling). Pooling is a basic way to reduce the number of parameters in the model. Parameter reduction is important in two ways which are forcing the model to recognize patterns and correspondences in data by representing it with fewer parameters, and reducing the computational cost of the system. The pooling operation is deterministic, so it doesn't have learn-able parameters. Its basic hyper-parameters are kernel size and stride value.

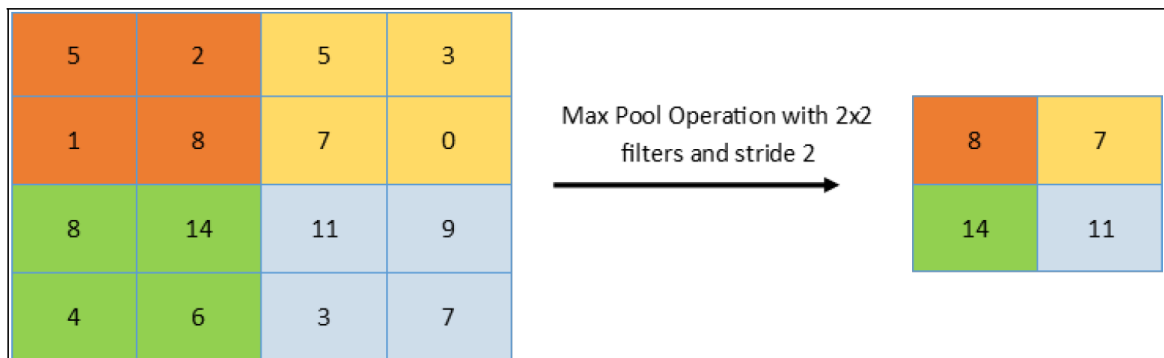


Figure 2.13. Pooling Operation

An up-sampling layer is the opposite function of the pooling operation that is shown in Figure 2.14. It takes several parameters in a certain area as input and doubles them in a certain way, such as nearest neighbour interpolation. Nonetheless, Up-sampling and pooling layers have a lot in common, actually. Up-sampling layer doesn't have any learning parameters, too. It has the same types of hyper-parameters as the pooling layer.

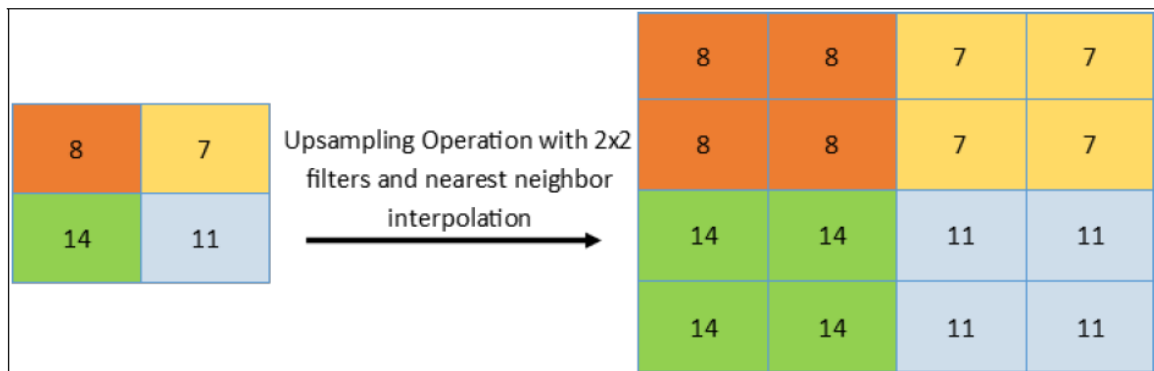


Figure 2.14. Up-sampling operation

2.3.1.3. Dropout Layers

Dropout layers apply dropout regularization to its input. Its only hyper parameter is the dropout rate which is non-trainable value. The dropout rate takes a value between 0 and 1. It determines how possibly activated neurons get set zero.

2.3.1.4. Fully Connected Layers

Fully connected layers refer to layers of ordinary feed forward neural networks. They are added after several convolutional layers. Convolutional layers are responsible for extracting features in data and fully connected layers are responsible for producing output. Their task can be classification or regression. CNN with fully connected layers is a common structure since it's much more efficient to pass data through convolutional layers before the neural network than sending raw data directly to the neural network.

2.3.1.5. Replacing Fully Connected Layers with Convolutional Layers

Replacing fully connected layers can be advantageous due to their computational cost and limited applicability. In a landmark localization problem, fully connected layers may produce a vector k that denotes to coordinates of the related landmark. It can be directly regressing the numbers or by sliding a window over the input image and classifying patches

that are cropped from the input image whether they include the related landmark or not. Instead, as proposed in [29] a convolutional layer with 1 x 1 kernel makes dot product over 3 dimensions which is equivalent to what fully connected layers do. Then it enables to produce the output by reducing the depth of the previous convolutional layer controllably to the desired output, which in our case it's a heatmap that represents the likelihood of position of related landmark [30].

2.3.1.6. Skip Layers

Skip layers is feeding a layer with not only output of previous layers but also output of a few previous layer [31]. It can be summing up feature maps or concatenating feature maps and send it to convolution layer. It is proposed that by doing so, backpropagation works efficiently and achieved preserving locations and characteristics of features in data.

2.3.1.7. Batch Normalization Layers

This type of layers is responsible for normalizing its input. Aim of batch normalization procedure is to increase the stability and performance on deep structures by maintains the activations of previous layer at certain range, for example between 0 and 1 [35].

3. MATERIALS AND METHODS

All of the computation operations are written in python programming language and well known libraries for python such as numpy, matplotlib. Dipy is used for handling and processing diffusion MRI data [37]. Keras is used to build CNN models [32].

3.1. CNN STRUCTURE AND TRAINING

Pre-processed diffusion MR data of 400 randomly chosen subjects, which are available under the 1200 Subjects Data Release of the Human Connectome Project (HCP), are used in CNN training [33]. 90 landmarks are randomly selected in multiple brain regions using the Harvard-Oxford Subcortical Atlas in MNI coordinates. The regional distribution of landmarks is specified in Table 3.1.

Table 3.1. The regional distribution of landmarks

Num. of Landmarks	Brain Region
30	White Matter
30	Gray Matter
10	Ventricles
4	Thalamus
4	Brain Stem
2	Caudate
2	Putamen
2	Pallidum
2	Hippocampus
2	Amygdala
2	Accumbens

Microsoft Azure cloud infrastructure is used to process a big amount of data. By using the “Batch Service” and Python API, 400 virtual machines running Linux operating systems are initiated and necessary software is installed. Each machine downloads the data of one subject

from Amazon servers, fits the tensor model to the diffusion MR data, and computes the eigenvectors and eigenvalues by using the DTIFIT tool of the FSL. Landmarks, defined in MNI coordinates, are transferred to the subject's brain by using the deformation field, which is estimated on T1-weighted MR images by FNIRT tool of FSL and available under the HCP repository [34]. The data are downloaded to the local workstation and color-coded fractional anisotropy (color FA) maps are calculated by element-wise multiplication of FA maps with biggest eigen-vector tensors. For each landmark of each subject, a heat map is generated as a 3-dimensional Gaussian function centred at the landmark location and having a standard deviation of 1 voxel.

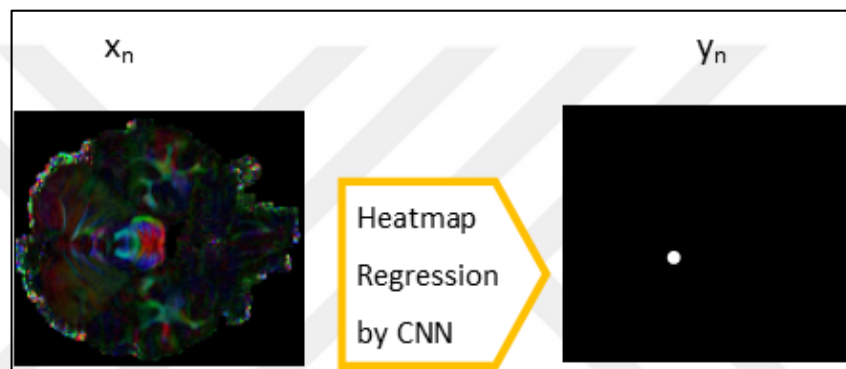


Figure 3.1. Sample of input data and label heatmap that shows position of selected landmark

In Figure 3.1, x_n denotes one of the slices of input data which landmark is located in, y_n denotes one of the slices of target output or label that shows location of landmark. The reason of using sliced images is ease of visualization.

Due to the large data size, instead of feeding the color FA field of the whole brain, $16 \times 16 \times 16$ patches are cropped and used. The locations of patch centers are randomly drawn from a uniform distribution on the brain mask. The corresponding patches at the same location and size are also cropped from the heat map. The patch is named positive if it includes the landmark, and negative otherwise. The training data are fed into the CNN in batches of 20 patches consisting of an equal number of positive and negative samples. The main structure of CNN is illustrated in Figure 3.2. CNN is trained jointly. That means for each of the 90 landmark points, a separate network with the same structure is trained.

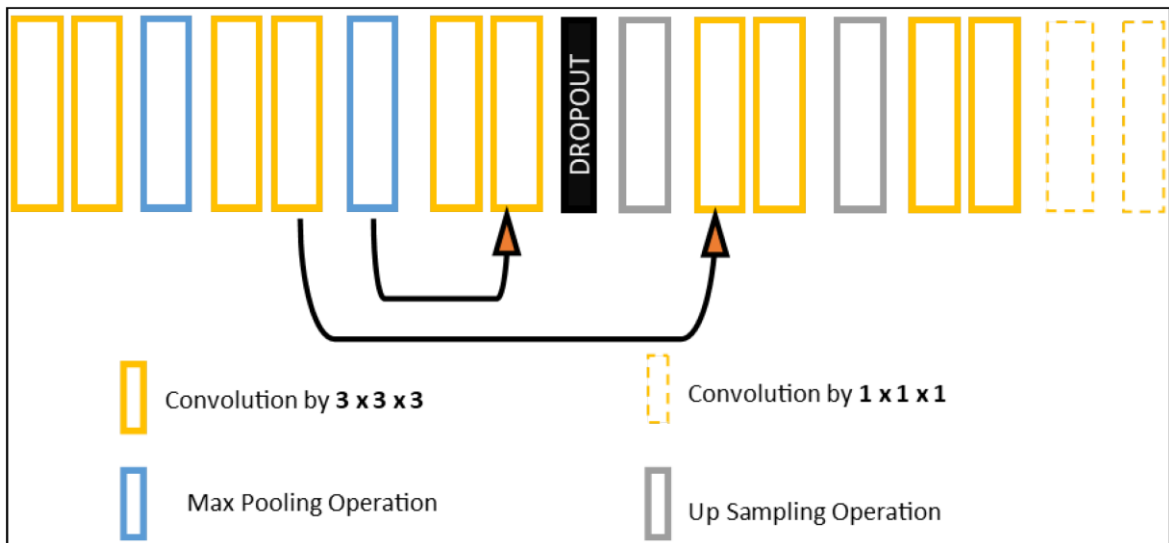


Figure 3.2. CNN layer structure

Due to the volumetric structure of the input data, all convolution, pooling and up-sampling operations are performed in 3 dimensions. Size of the convolution kernels is $3 \times 3 \times 3$ except for the last two layers since the last two layers are assigned to a different task which is reducing feature maps so that the network could generate numerous heatmaps, whereas $2 \times 2 \times 2$ masks are used in pooling and up-sampling. Exponential linear unit (elu) is used for the activation function [18] in all convolution layers except sigmoid function is preferred in the last layer. The tensor is normalized after the elu activation function applied to the output of each convolution layer. The main reason behind this batch normalization procedure is to regulate the layer output to a certain distribution to increase the stability and performance on deep structures[35]. Dropout with 0.25 rate is applied before first upsampling layer.

The input size of the network is $16 \times 16 \times 16 \times 3$, and the output is $16 \times 16 \times 16 \times 1$. The last dimension of the output corresponds to the number of desired heatmaps and is set to 1 for detecting a single landmark at a time. There are 2 residual connections in the network model, as shown in Figure 3.1. The output of the layer is not only passed to the next layer but also skips a few layers and concatenates with another layer. The residual structures are proposed to help to preserve the spatial information and increase the efficiency of backpropagation [31].

Unlike traditional CNNs as in [24], there are no fully connected neural network layers stacked after convolutional layers at the end of the proposed network. In traditional networks, after consecutive pooling and convolution operations, the number of parameters becomes sufficiently small to go under fully connected neurons for regressing coordinates of the desired landmark. Differently, at this stage, the proposed network starts to increase the size of data again by employing upsample layers, which perform nearest neighbour interpolation. The number of upsampling and pooling operations are equal to each other so that input and output could be the same size. After processed data reach the same size of the input, two convolution operations with $1 \times 1 \times 1$ kernel size are applied to generate the heatmap [30].

The depth of first convolutional layer is 64, depth is increased by 2 after pooling operations, and is reduce by 2 after up-sampling operation. The loss function is determined as mean squared error. RMSProp [21] is used for optimization with 2×10^{-5} learning rate and 5×10^{-8} decay. Training for each landmark point last 5000 iterations where one iteration is defined as one weight update as a result of a batch passes through the network.

3.2. TESTING AND EVALUATIONS

3.2.1. HCP Dataset

Color FA maps of 50 randomly selected HCP subjects, that are not included in the training dataset, are used in evaluation studies. A $16 \times 16 \times 16$ window over input image and with half value of window size which is 8. These windows are sent to CNN model. Naturally, there would be overlapped regions, since windows that are sent to CNN are overlapped. So, not all non-zero values include landmark. It's assumed that maximum values on the label map would indicates landmark. The final map is obtained by summing the output heatmaps of the network for each patch and the **location of the maximum voxel** is labelled as the landmark. The process time which is highly depending on the hardware capabilities of the workstation, for the whole brain of a subject takes approximately 90 seconds. That time is highly dependent to hardware specs⁴ of work station such as memory size, type of storage, capability of CPU and GPU.

Assigning number of slide equal to half size makes all points are sent to CNN 2 times. CNN model has a chance to see the landmark more than one. On the other hand, it also has a chance to see structures that CNN thought it was a landmark. Therefore, despite increasing computational cost, sending overlapped patches increases confidence of CNN model.

To evaluate the performance of the proposed landmark detector, results are compared to locations that were marked by using FNIRT tool of FSL. 3 metrics are defined as success rate, accuracy and euclidean distance between the landmarks.

Success Rate: If the euclidean distance of CNN prediction to FNIRT marked point is less than 1.25 cm, the prediction is considered as success, otherwise failure.

Distance: The average of euclidean distances of FNIRT and CNN proposed landmark positions on subjects, which satisfy success criterion, is calculated in millimeter unit.

Accuracy: In order to evaluate the accuracy, overlap of eigenvector-eigenvalue pairs (OVL) measure, which compares the similarity of two tensor fields F and M , is used [36]. OVL is given in equation 3.1 and is used for registration quality assessment in [15].

$$OVL(F, M) = \frac{1}{N_{\mathbb{R}}} \sum_{x \in \mathbb{R}} \frac{\sum_{i=1}^3 \lambda_i^F(x) \lambda_i^M(x) \left(e_i^{F^T}(x) e_i^M(x) \right)^2}{\sum_{i=1}^3 \lambda_i^F(x) \lambda_i^M(x)} \quad (3.1)$$

where λ are eigenvalues, e are eigenvectors, \mathbb{R} is the region of interest selected as 5 voxel cube centered at the landmark location and $N_{\mathbb{R}}$ is the number of voxels in region of interest. For each landmark, the average OVL score of every possible F, M combination on 50 test subjects that satisfy the success criterion is calculated. This value quantifies the consistency of the local tensor fields around the landmark among subjects.

3.2.2. Clinical Dataset

Points that MNI coordinates correspond to landmarks are marked on IIT DTI atlas (v5.0) which is publicly available. Then, predictions of CNN model on clinical color FA maps of 7 patients are visually compared one by one with points marked on atlas to obtain success rate. If CNN model predicts a point which is visually similar point on all of the 7 patients,

its success rate will be 1, if its predictions are not visually similar to the point marked on atlas on all of the 7 patients, its success rate will be 0.

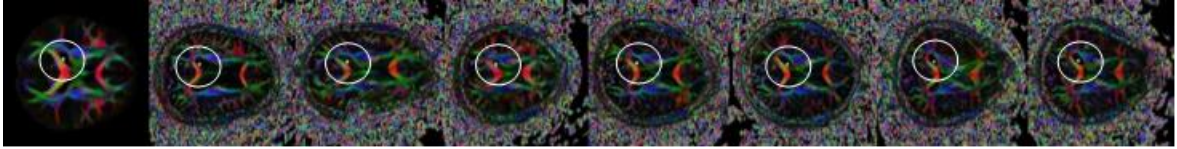


Figure 3.3. CNN prediction that shows success all of the patients, that makes its success rate 1. The first image on the left is atlas image with MNI coordinate of relevant landmark is marked and circled.

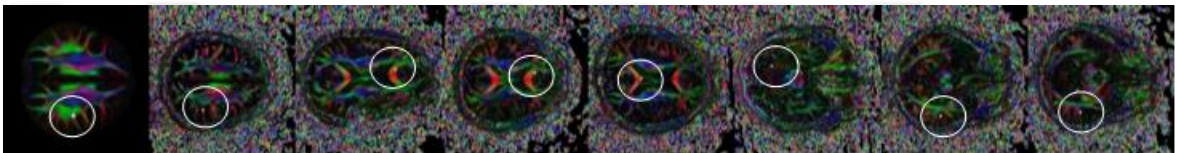


Figure 3.4. CNN prediction that shows success none of the patients that makes its success rate 0. The first image on the left is atlas image with MNI coordinate of relevant landmark is marked and circled.

4. RESULTS

4.1. RESULTS ON HCP DATA

The success rate for each landmark mapped on a glass brain are given in Figure 4.2 and the average value grouped in brain regions are given in the graph, in Figure 4.1 . A zero success rate indicates that the landmark could not be detected in any of the 50 subjects.

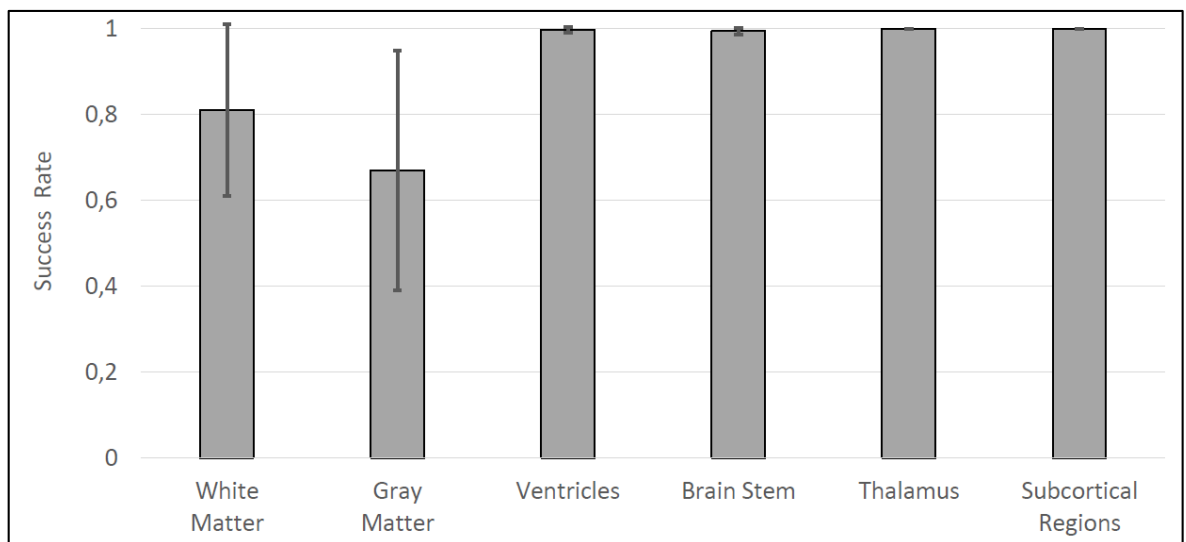


Figure 4.1. Regional averages of success rates

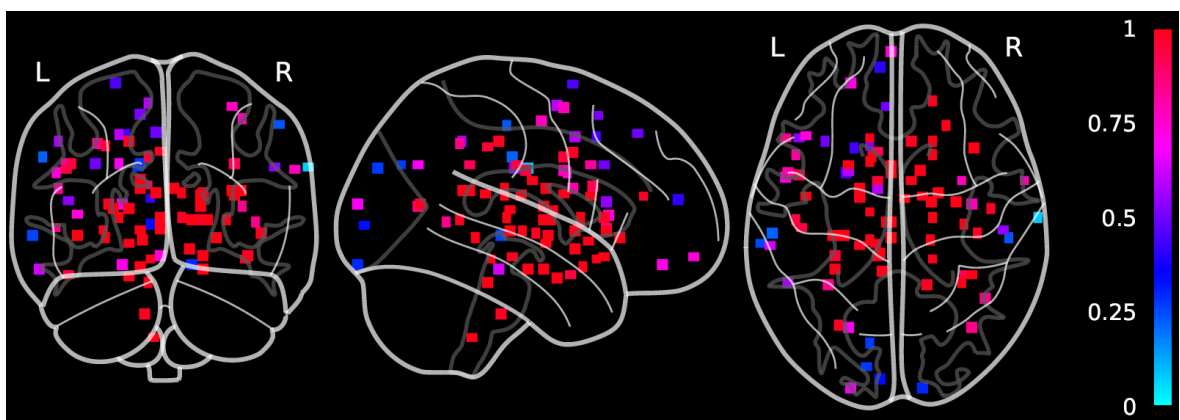


Figure 4.2. Average success rate for each landmark on glass brain template

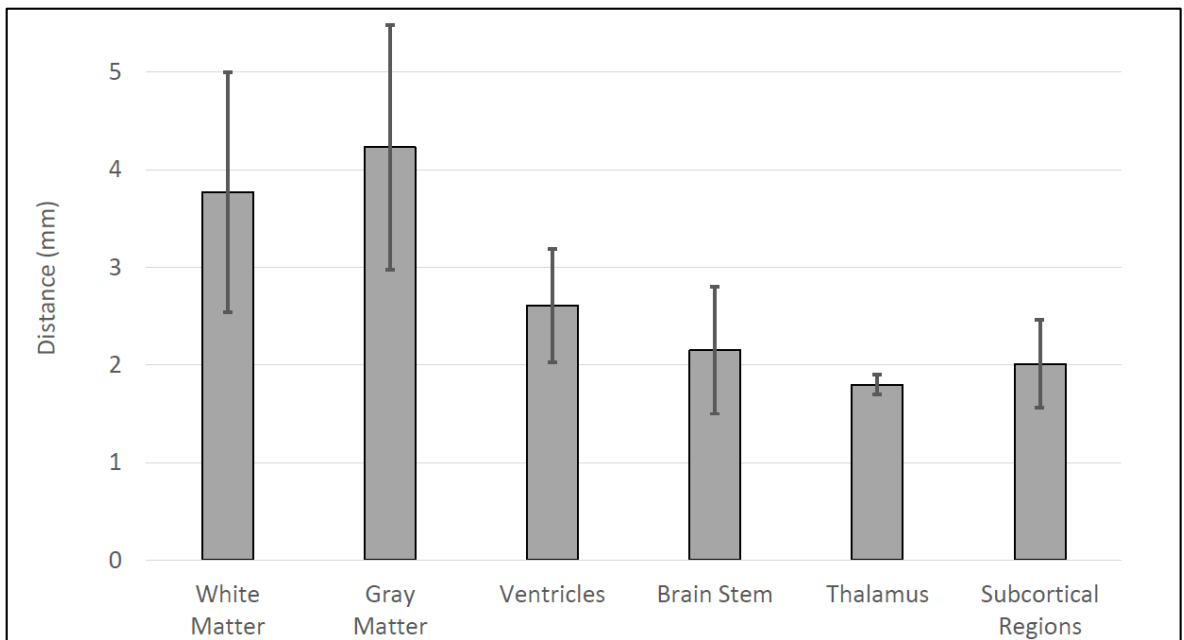


Figure 4.3. Regional average distance between FNIRT and CNN predictions

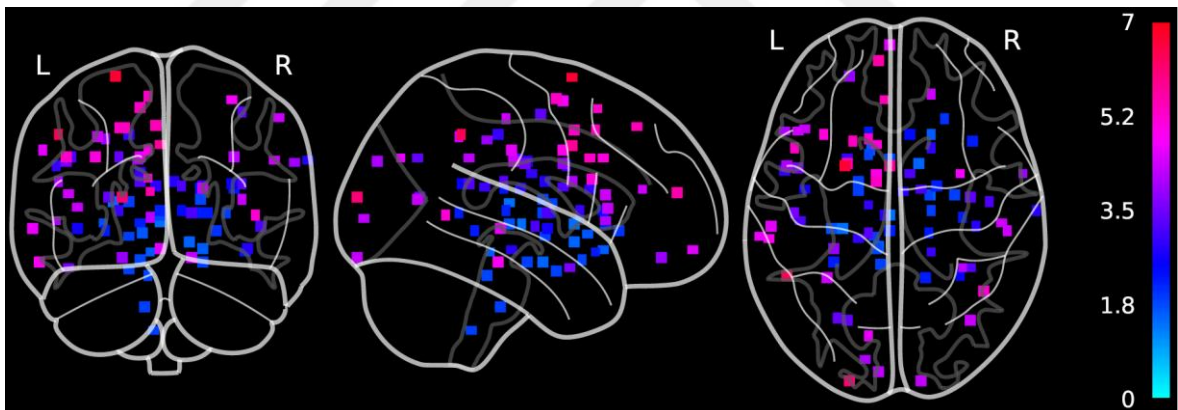


Figure 4.4. Average distances of FNIRT and CNN predictions on glass brain template

The regional distances in millimeter between FNIRT labels and CNN predictions for each landmark mapped on a glass brain are given in Figure 4.3 and the average values grouped in brain regions are given in the graph, in Figure 4.4 .

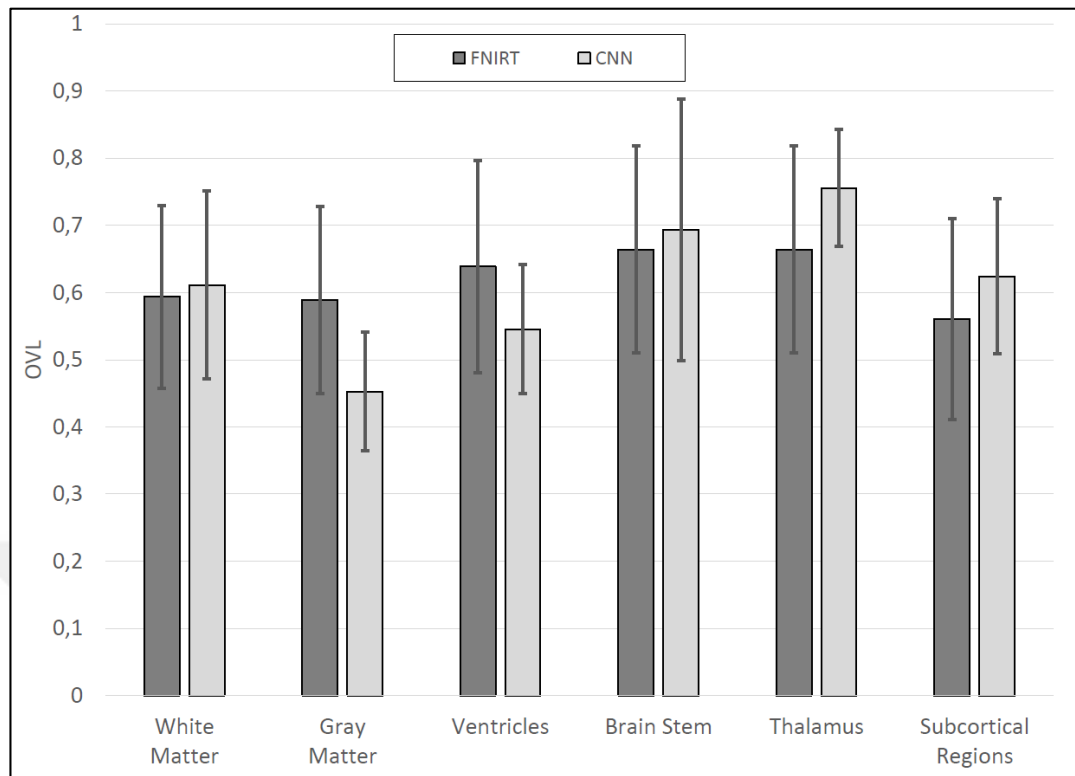


Figure 4.5. Regional average differences of OVL Scores

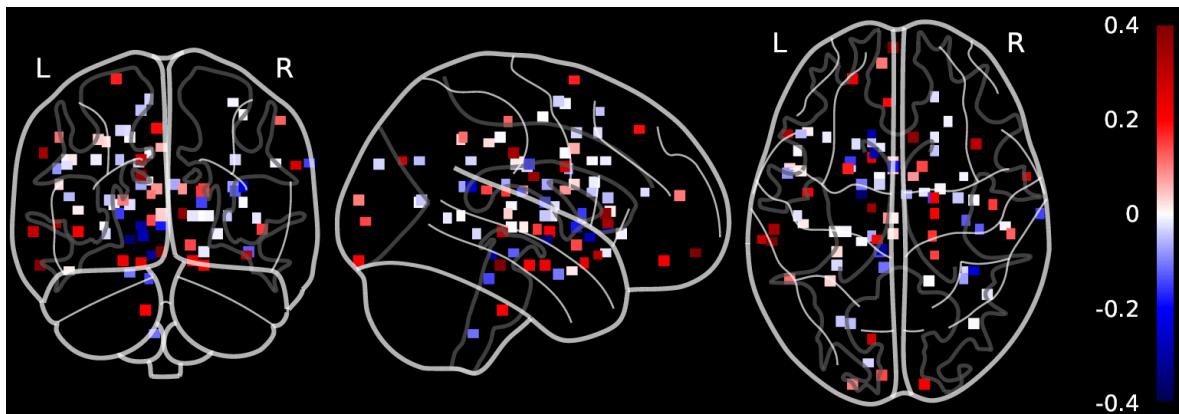


Figure 4.6. Average difference between the OVL Scores on glass brain template

Difference of OVL scores of CNN model prediction from FNIRT labels for each landmark mapped on a glass brain are given in Figure 4.6 and the averaged values of both CNN predictions and OVL scores of FNIRT labels grouped in brain regions are given in the graph, in Figure 4.5.

Table 4.1. Average and standard deviations of the evaluation results based on regions.

Brain Region	Succes Rate	Distance (mm)	FNIRT OVL	CNN OVL
White Matter	0.814±0.208	3.01±0.98	0.59±0.14	0.61±0.14
Gray Matter	0.667±0.283	3.37±1.00	0.58±0.13	0.45±0.08
Ventricles	0.998±0.006	2.09±0.46	0.63±0.16	0.54±0.09
Brain Stem	0.995±0.008	1.72±0.51	0.66±0.15	0.69±0.19
Thalamus	1.000±0.000	1.44±0.08	0.66±0.15	0.75±0.08
Subcortical	1.000±0.000	1.66±0.36	0.56±0.15	0.62±0.11
OVERALL	0.826±0.240	2.72±1.09	0.60±0.14	0.56±0.14

Summation of the results on HCP dataset are presented in table 4.1.

4.2. RESULTS ON CLINICAL DATA

Anonymized 3T MRI data with spin echo scanning sequence, 1.75 x 1.75 mm pixel spacing and 2.5 mm slice thickness, that includes DWI, b-values and b-vectors is acquired from Yeditepe University Hospital with the approval of ethics committee. Diffusion images of 7 patients are fit to a tensor using dipy which is a free and open-source python library for computational neuroanatomy, focusing mainly on diffusion magnetic resonance imaging (dMRI) analysis [37]. Afterwards, color FA maps are generated and sent to CNN model as 16 x 16 x 16 x 3 patches. Maximum point of prediction result are marked on clinical data. Next step is to inspect whether there is a correlation between maximum point and related landmark.

It needs to be said that clinical data is very noisy, unaligned, low-resolution compared to HCP dataset. And there is no pre-process step applied to data in order to measure capability and robustness of CNN model.

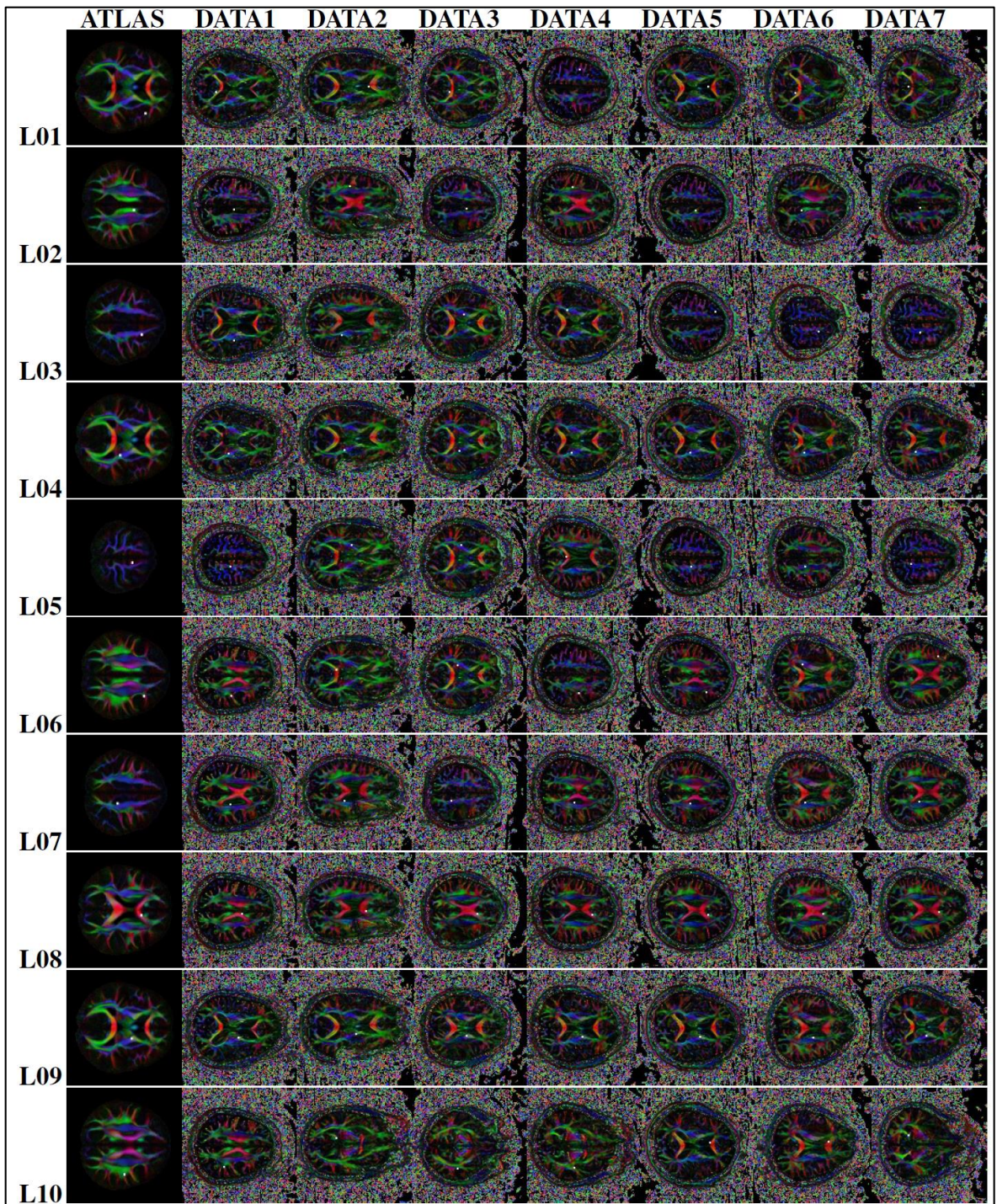


Figure 4.7. Predicted points of CNNs over 7 patient data for Landmark nu. 1-10.

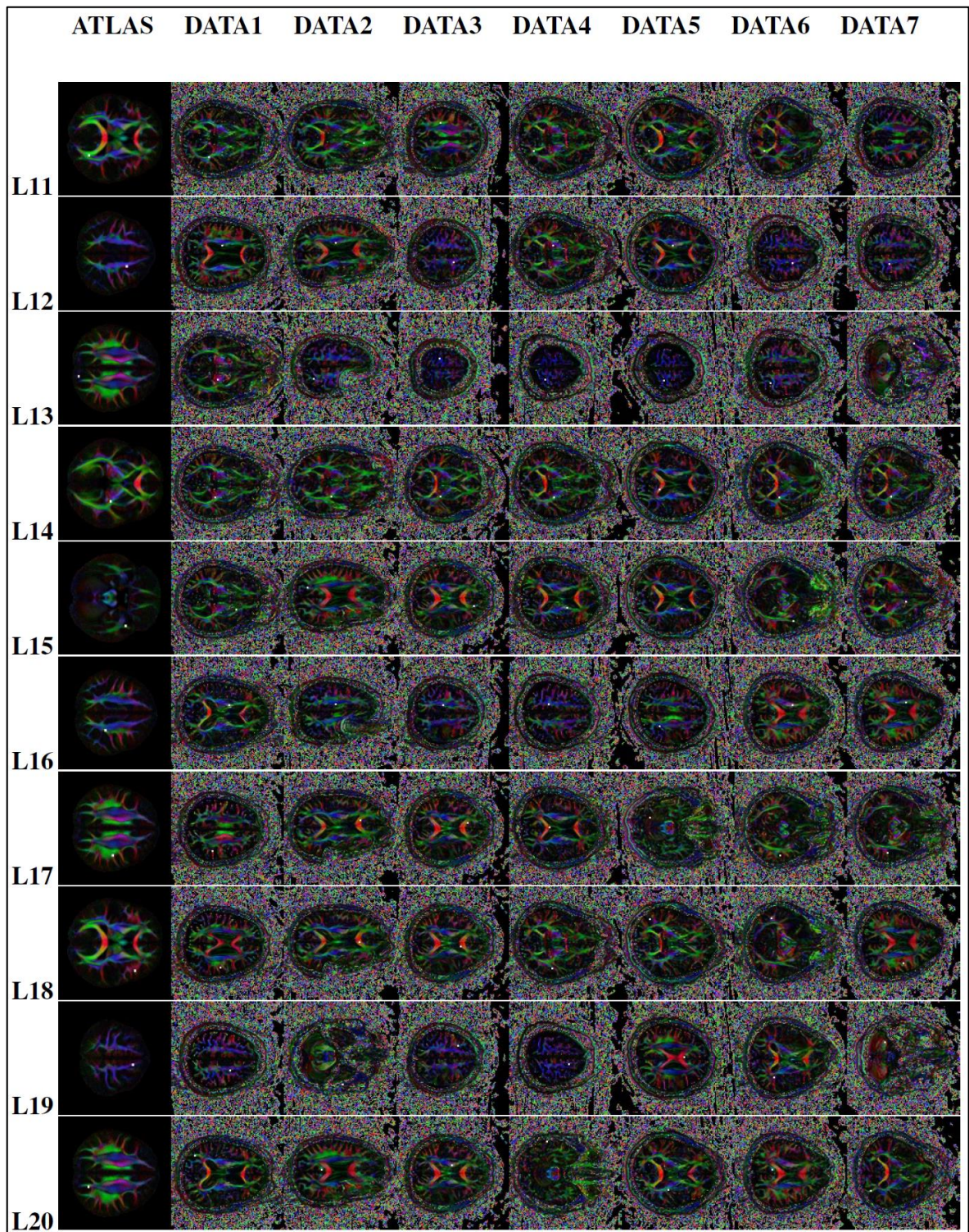


Figure 4.8. Predicted points of CNNs over 7 patient data for Landmark nu. 11-20.

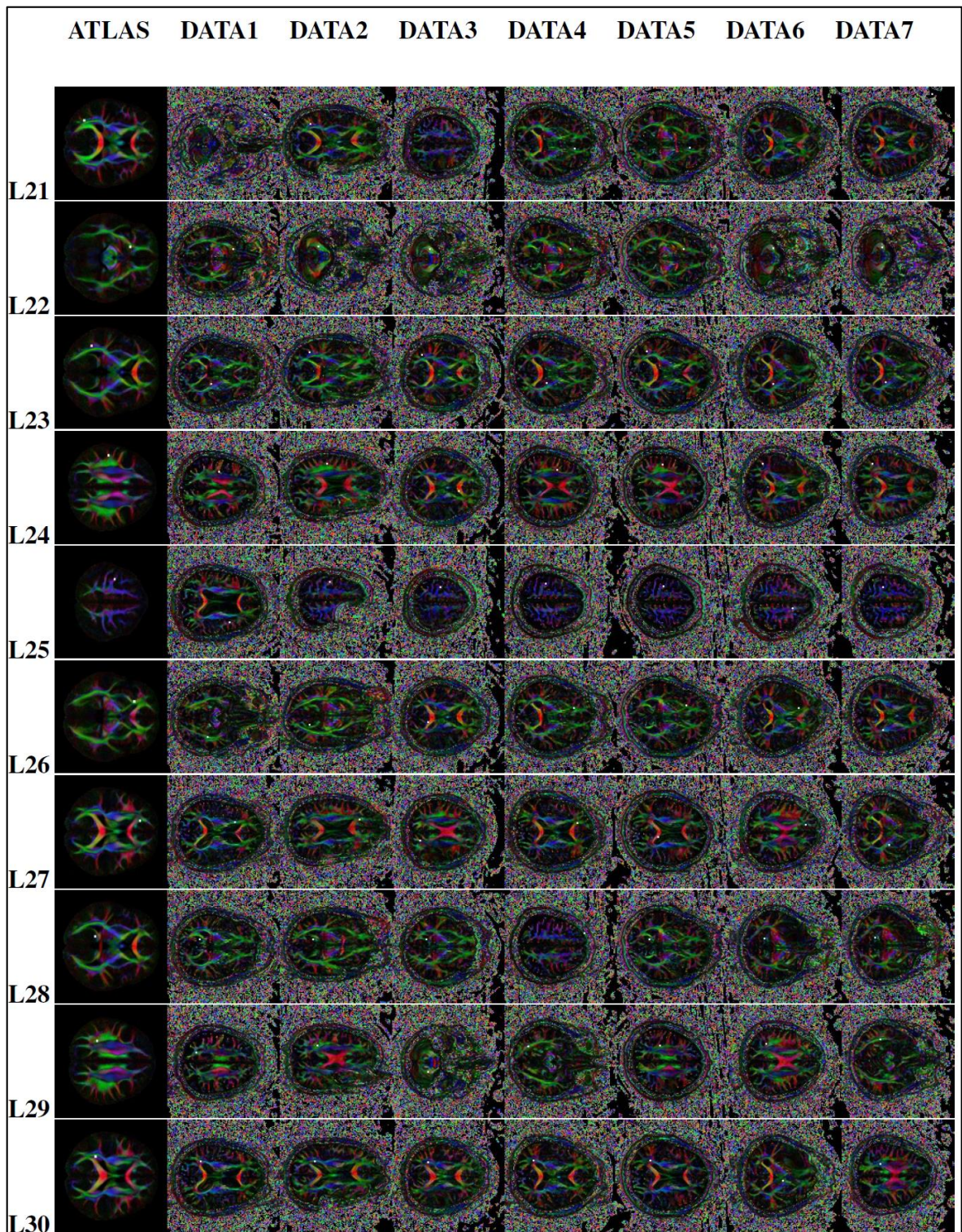


Figure 4.9. Predicted points of CNNs over 7 patient data for Landmark nu. 21-30.

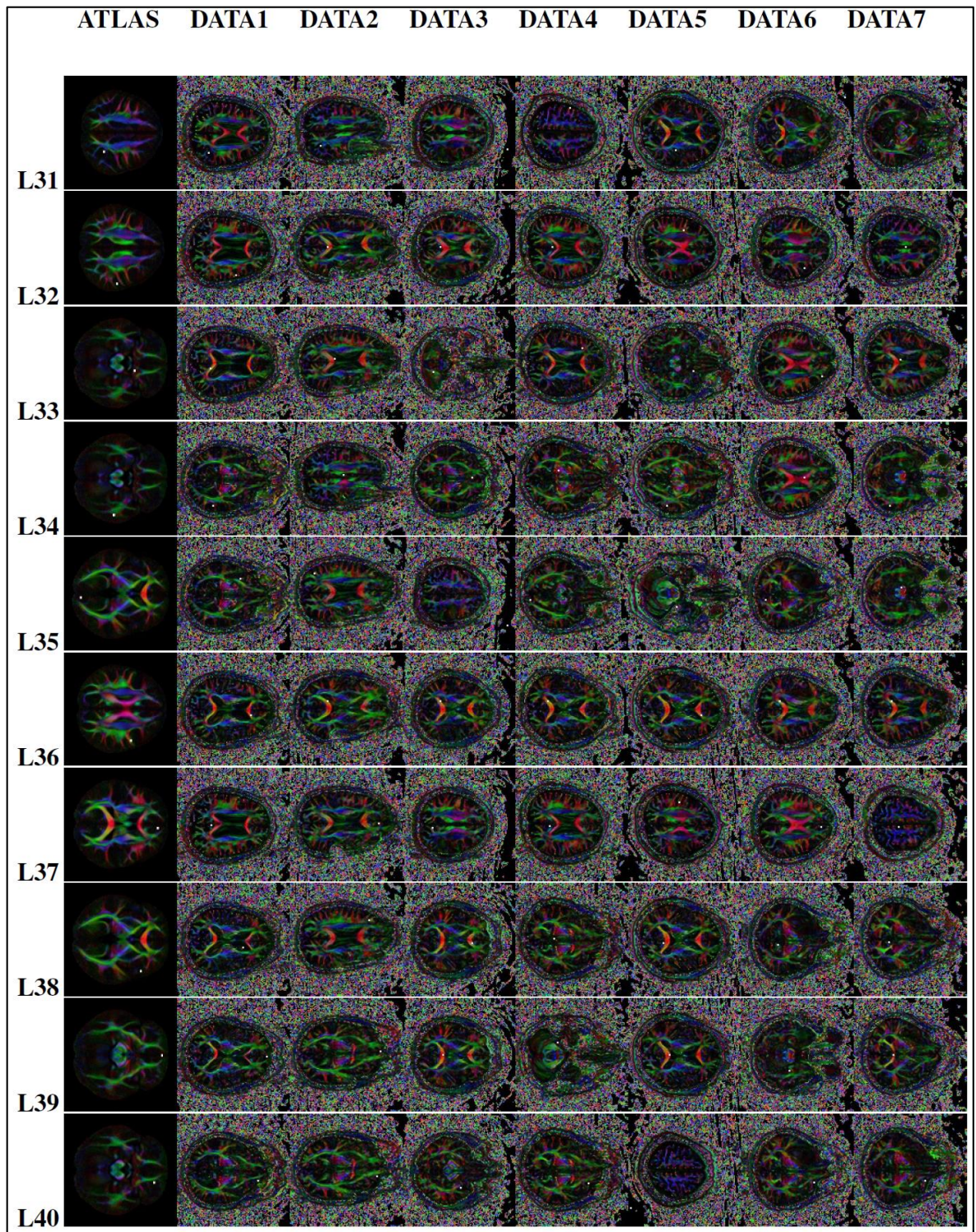


Figure 4.10. Predicted points of CNNs over 7 patient data for Landmark nu. 31-40.

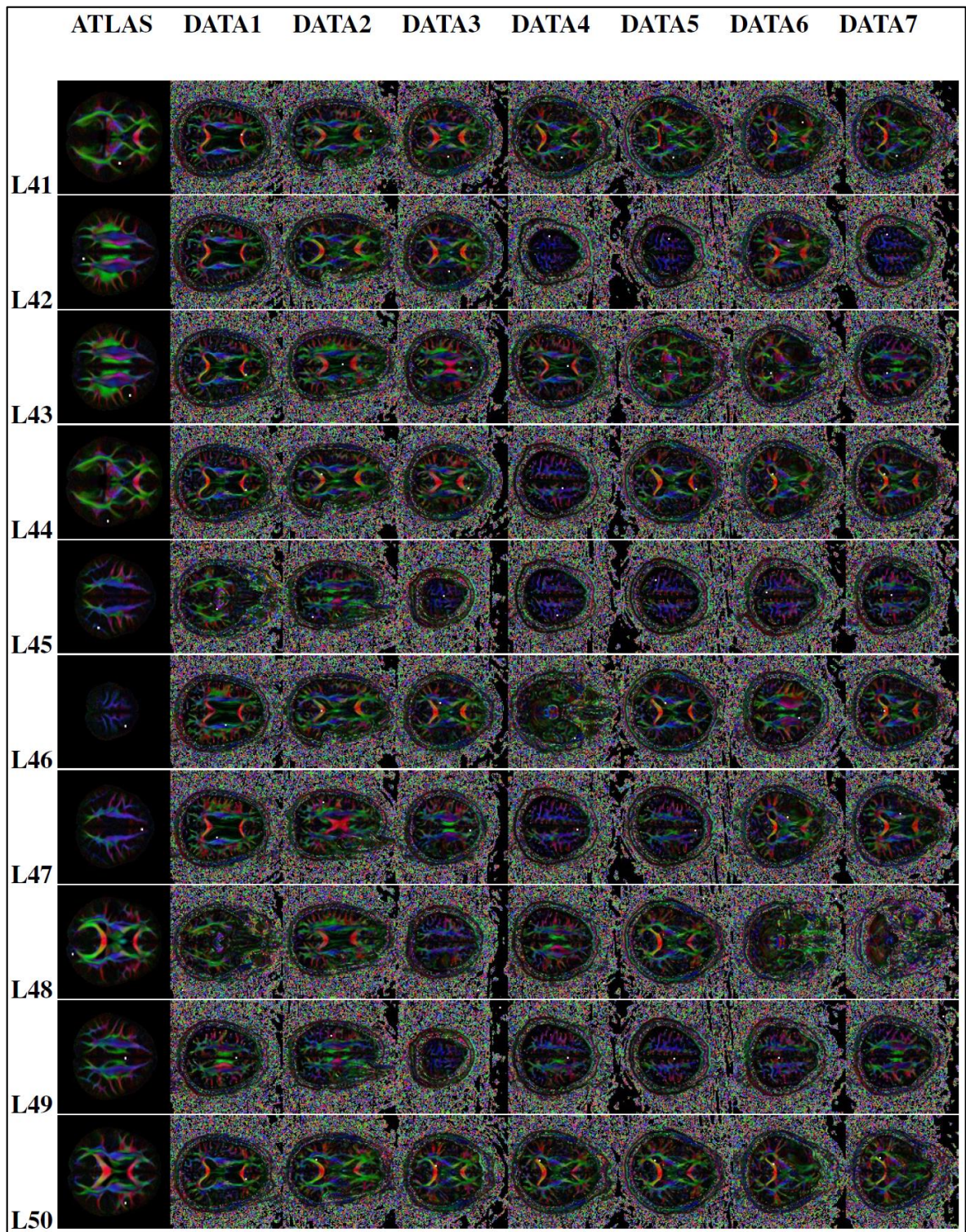


Figure 4.11. Predicted points of CNNs over 7 patient data for Landmark nu. 41-50.

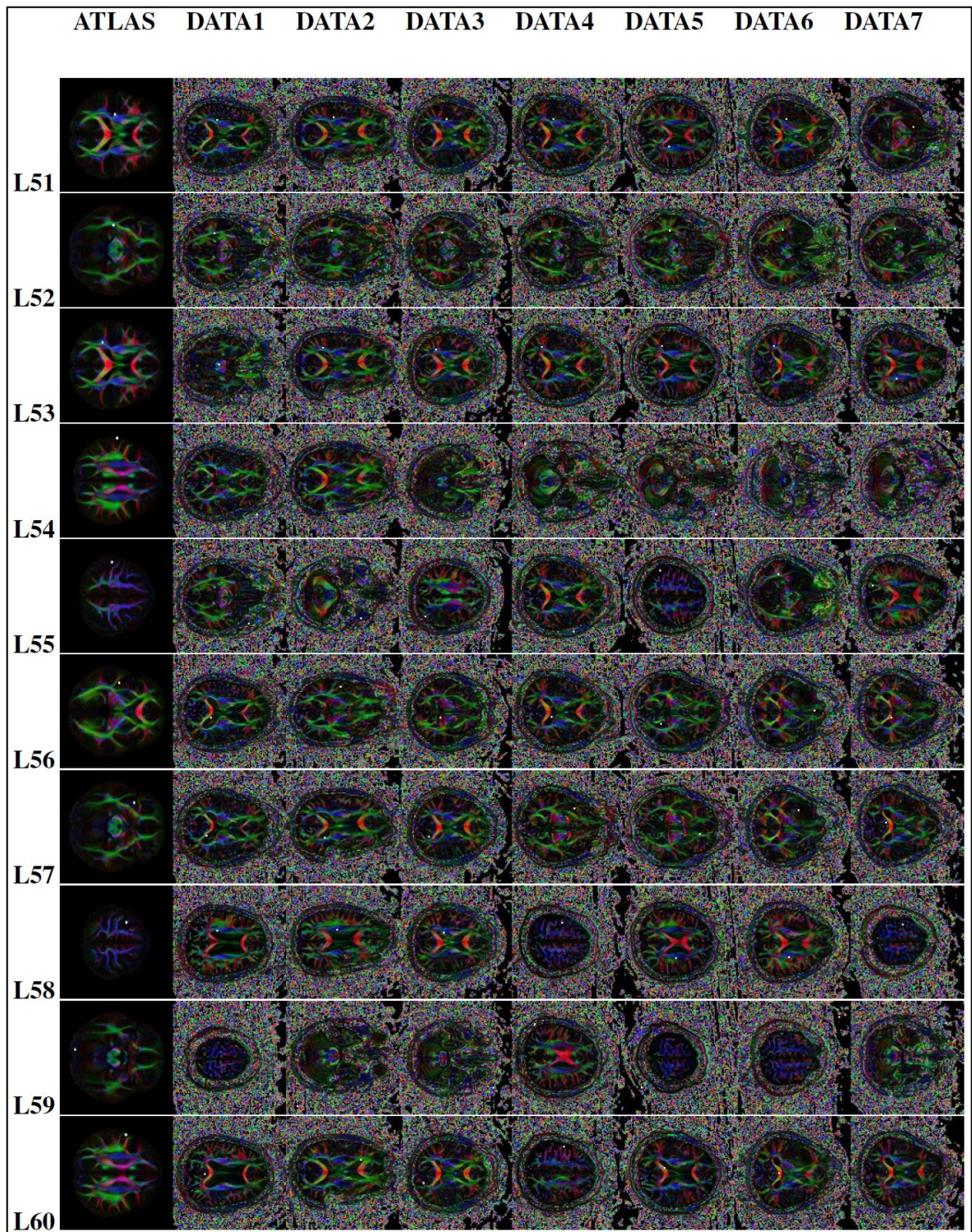


Figure 4.12. Predicted points of CNNs over 7 patient data for Landmark nu. 51-60.

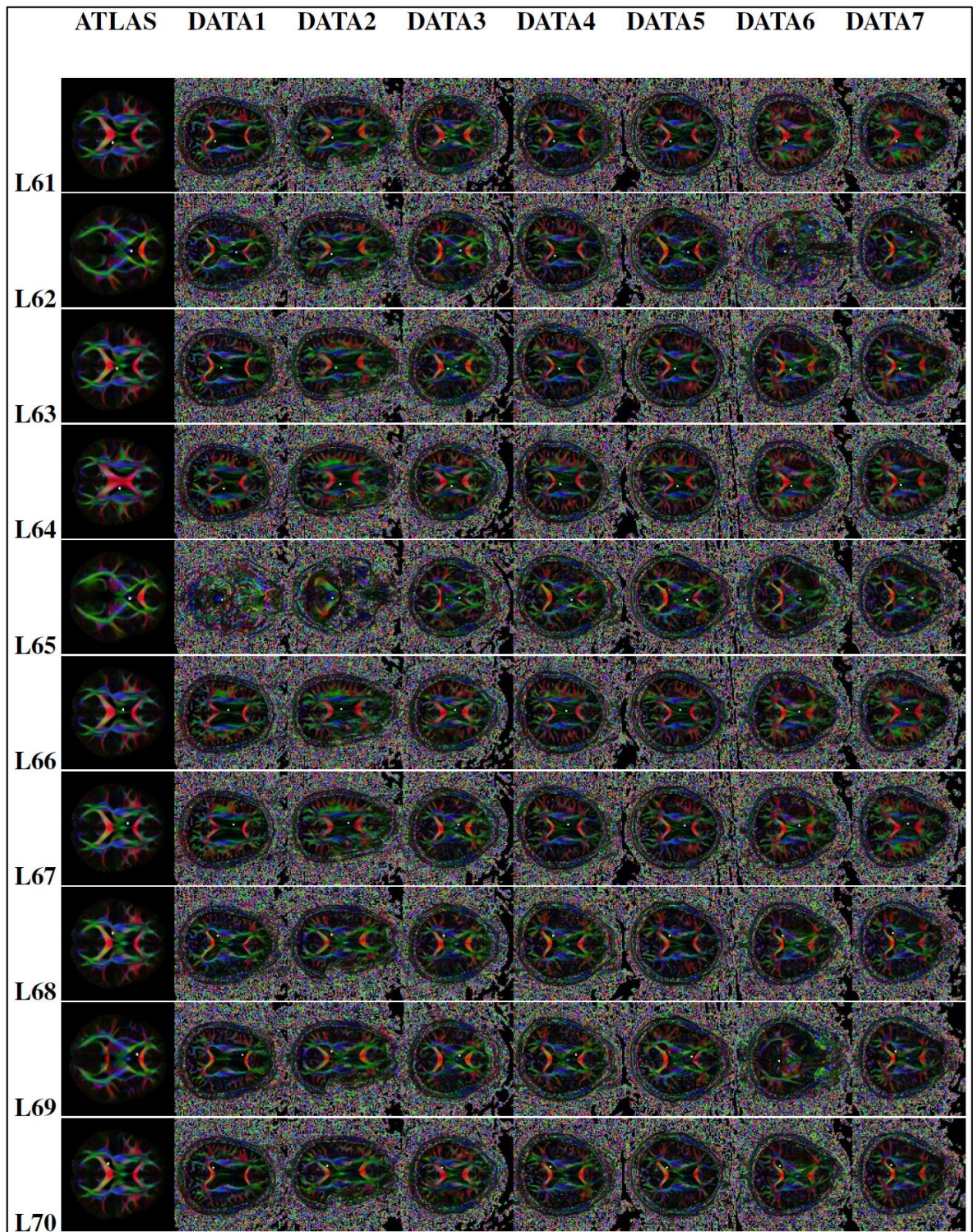


Figure 4.13. Predicted points of CNNs over 7 patient data for Landmark nu. 61-70.

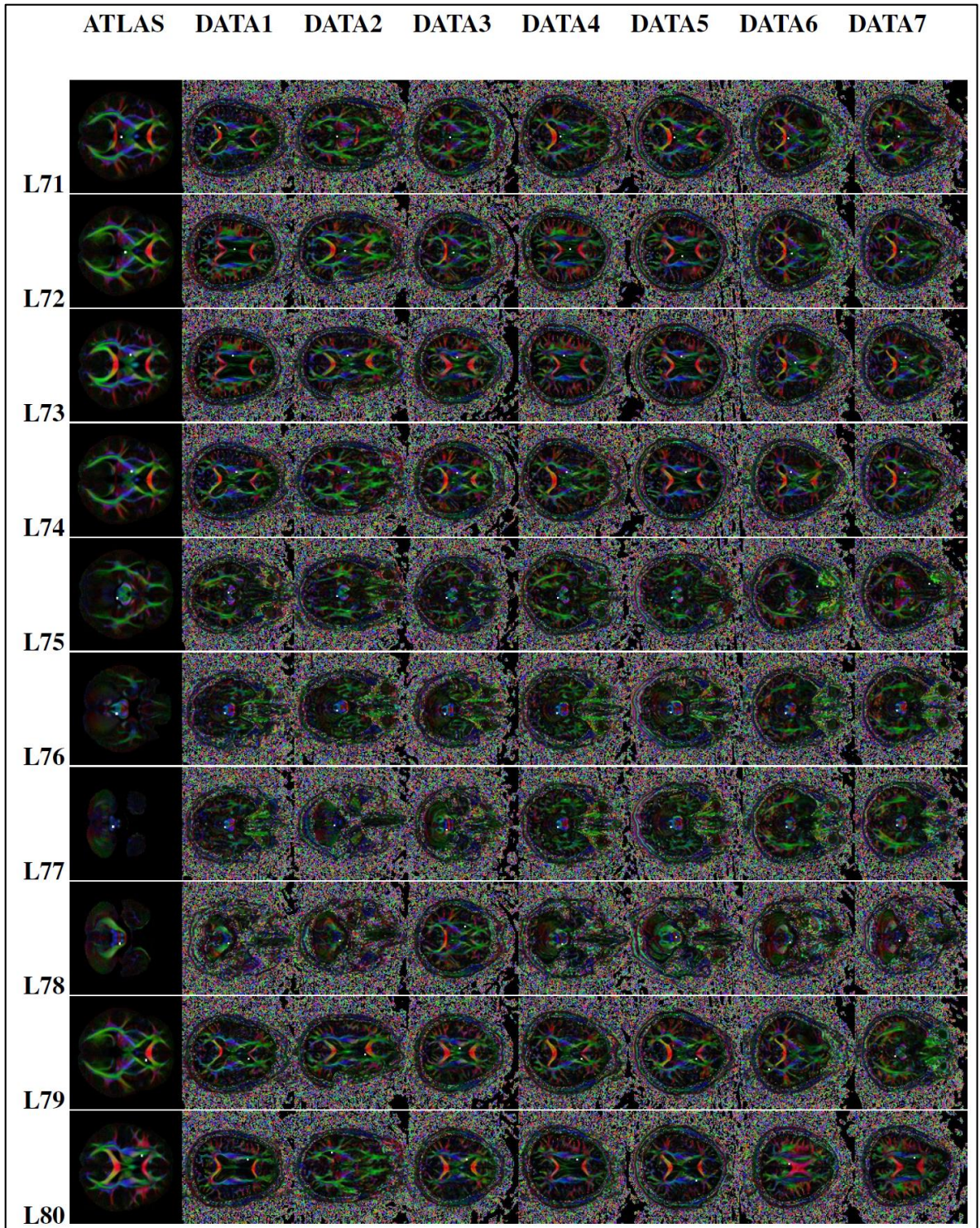


Figure 4.14. Predicted points of CNNs over 7 patient data for Landmark nu. 71-80.

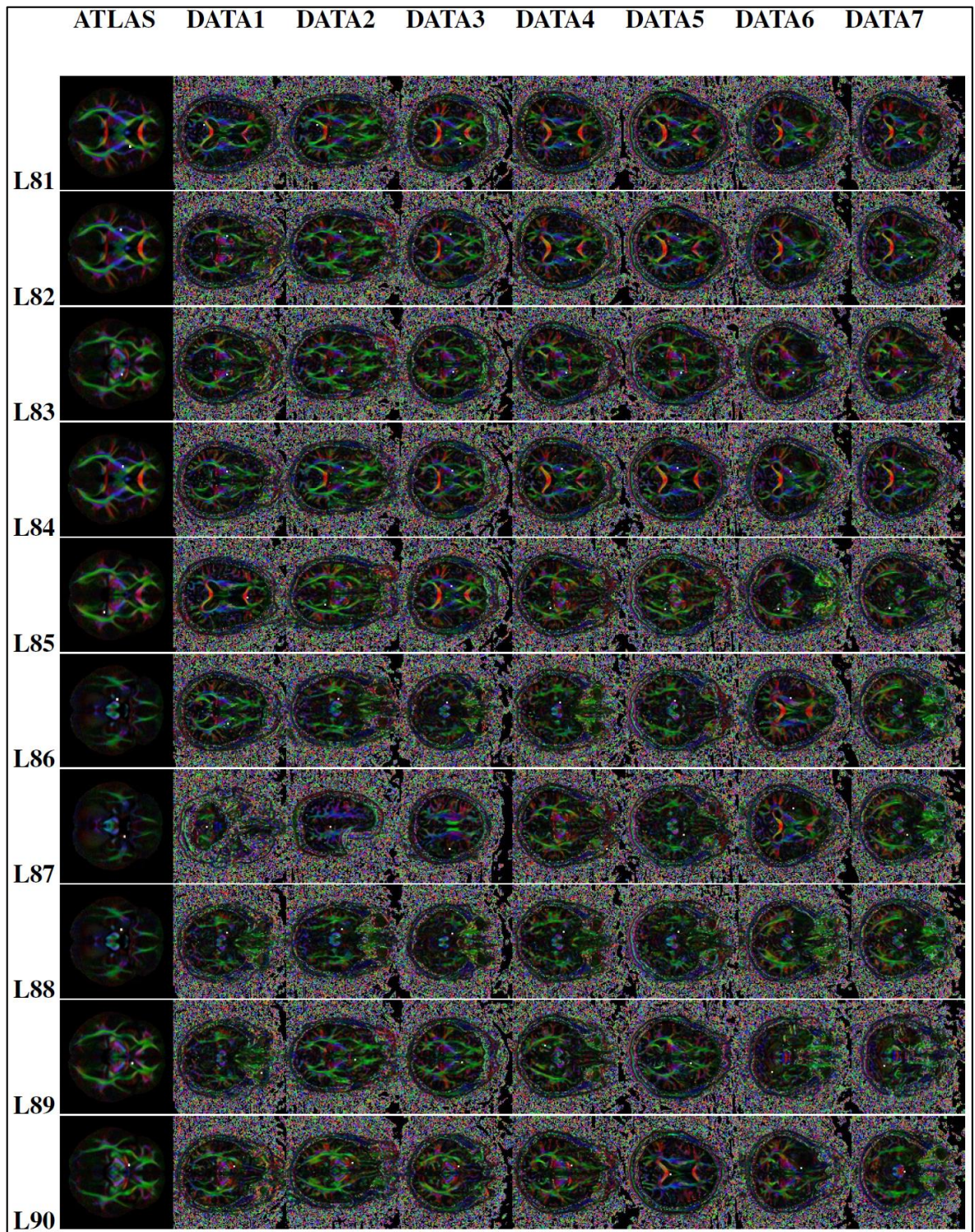


Figure 4.15. Predicted points of CNNs over 7 patient data for Landmark nu. 81-90.

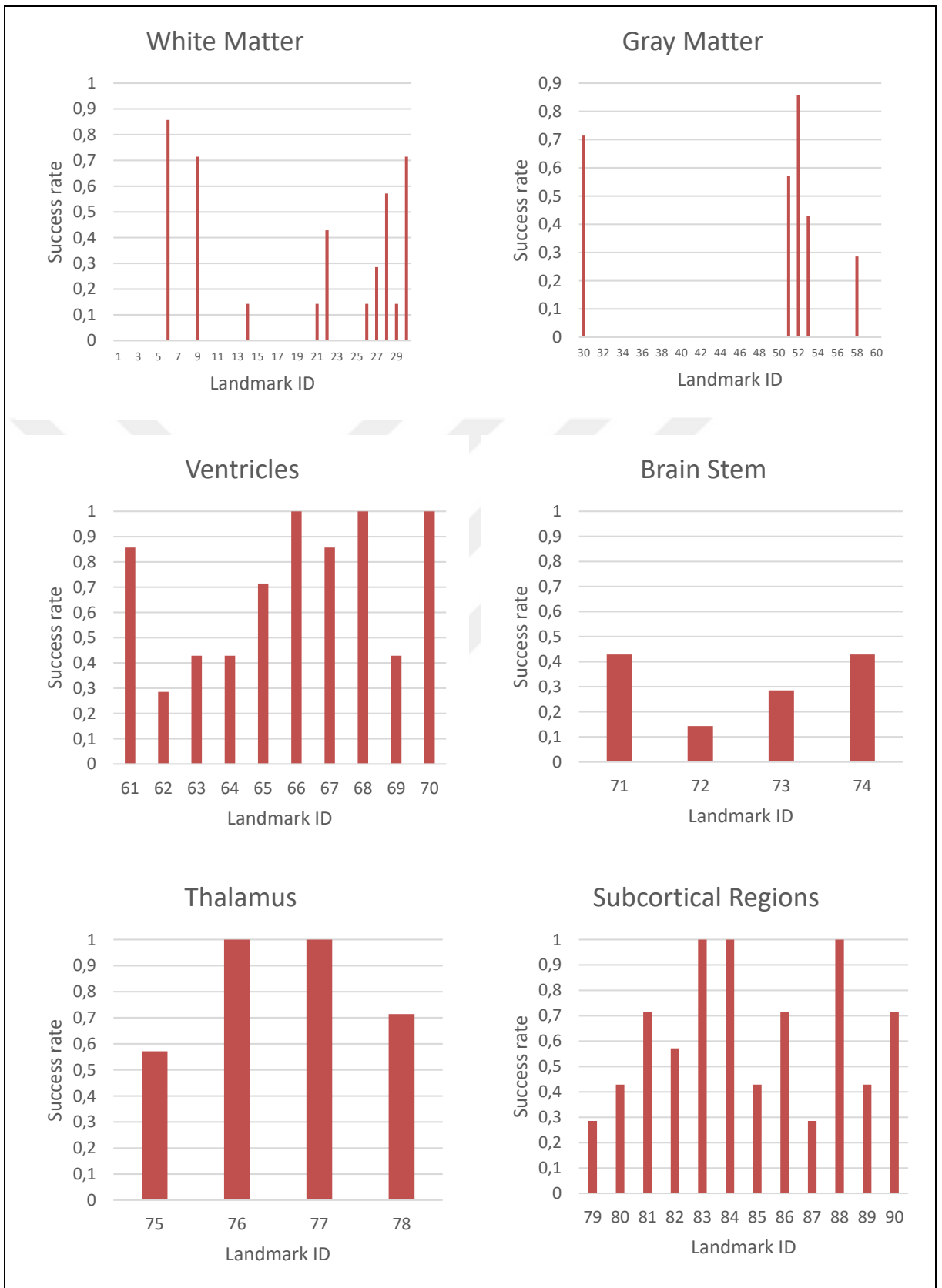


Figure 4.16. Regional averages of success rates of clinical data

In Figure 4.16, success rates of landmarks over 7 patients that are grouped based on regions are shown. As seen, the worst results were obtained on gray matter whereas the best results were obtained from thalamus.

Table 4.2. Average results based on regions of clinical data

Region Name	Success Rate
White Matter	0.14±0.25
Gray Matter	0.07±0.20
Ventricles	0.70±0.27
Brain Stem	0.57±0.11
Thalamus	0.93±0.23
Subcortical Regions	0.69±0.24

In table 4.2, overall results on clinical test data grouped by regions are shown.

5. DISCUSSION

In this dissertation, a CNN model is proposed that locates specified landmarks on the target image set. Diffusion MR images were chosen for applying landmark localization method for several key points. First, it provides richer information about connectivity in brain than other structural images. As an example of usefulness of DTI, pre-operative planning of tumour removal operation can be given. The task is to maximize the extent of tumor resection, but at the same time minimize the neurological damages. It's only possible if relationship with tumor and functional structures such as gray matter, white matter tracts is known. It is reported that DTI is a better option for cortical mapping than functional MRI, or electro-cortical stimulation methods [38].

There are notable studies that aim to locate structures on brain using connectivity. In [9], the aim is to precisely reproduce selected landmarks on brain using connectivity in the brain. Although the aim is similar, methods that are followed are quite different. Deep learning based approach is not preferred.

CNNs have been solidifying their positions in computer vision literature with a help of their good grasp of understanding the patterns in the data. Recently proposed improvements and CNN architectures, and have so much potential application areas besides the originally addressed problem. In that context, the proposed CNN model was originally inspired from hourglass CNN design which was proposed to estimate human pose in the wild from 2D images [30]. In [39], the same CNN structure was used as a starting point to develop a system that reconstructs 3D face from single 2D image.

Hourglass architecture, as known as encode-decode architectures has an advantage of overcoming problems related to transformations in image set [30]. It has been very popular approach. One of the well-known studies that uses similar architecture is U-Net, which aims segmentation of medical images by classifying each pixel [40]. On the other hand, aim of this study is to regress the location landmarks on diffusion MRI images, which is 4D. Therefore, modifications and simplifications are tailored to this specific problem.

CNN was trained with color-coded FA maps of the HCP dataset with identical spatial position, direction, and resolutions and showed different results in regions of the brain. The

worst results were obtained on the gray matter while the best results were obtained on subcortical regions. Landmarks located at the inner regions of the brain could be detected with a higher success rate. Despite the possibility that this is because inner regions have richer characteristics, landmarks on ventricles, which are observed homogeneous on color-coded FA maps, are also located with a high success rate. Even if the failure cases are removed, the distance between CNN and FNIRT predicted landmarks is higher in the outer regions of the brain. The reason for this low performance might be due to some sort of systematic error related to patches that are close to the edge of the brain, which needs to be investigated.

OVL results shows that CNN outperforms on white matter, brain stem, thalamus, subcortical regions which are rich in fiber structures, meanwhile FSL-FNIRT performs better on gray matter and lateral ventricles where diffusion tends to be isotropic. Although CNN training is performed by using the FNIRT labelled landmarks, the performance of CNN is higher in some regions, which is an interesting result. Therefore, CNN does not learn the FNIRT labelling process but the underlying diffusion pattern. It should also be noted that OVL metric on tensor field consistencies might result in an unbalanced comparison, for FNIRT method implements T1 weighted MR images as the data term.

CNN was trained on the HCP data which is high resolution, cubic voxel structured, pre-processed data. However, when making predictions over low resolution raw clinical data that was acquired directly from scanner was not edited, pre-processed, unmasked. Nonetheless, reproducible results were obtained for some landmarks that are mostly located at inner regions of brain or intersections of fibers. In addition, since the CNN model works on patches, it's guaranteed that CNN model does not learn location of landmarks, it learns characteristics of data. Results on clinical data, especially on data 6 show that the system has a certain level of robustness against tumor caused deformations, especially when inspecting landmark no.74. Also, regional results on HCP dataset and clinical dataset show significant correlations. High success rate on ventricles, which have isotropic diffusion characteristics shows that these landmarks are not memorized but learned by the CNN model. Nonetheless, this information has to be solidified by more comprehensive numerical tests.

In order to get better results, first step must be data augmentation. Since positions of patch centres were chosen so that they were uniformly distributed while generating training data, any additional data augmentation step was not considered. In further stages, while evaluating

results, it has been observed that starting to slide window from different voxel changed the results. Since there is no correlation between getting better results and selecting starting point closer or farther from centre of edge of image, more comprehensive test are not regarded necessary.



6. CONCLUSION

Nowadays, CNN provides the best results in computer vision problems. Henceforth it's the most promising approach to overcome frequently confronting problems such as translation, rotation or scaling objects in images, different types of noises, working on different resolutions. Well working network design could not only work on different sets of examples from different problems. Furthermore, besides adapting network designs to a different problem, opportunity transferring the learned representations to other networks enables getting results faster and better.

Getting rid of fully connected neural network layers and replacing them with convolutional layers reduced the computational cost. Nonetheless, total cost is still concerning. CNN based landmark localization frameworks could be more feasible to widen its area of use. Therefore computational cost and training time must be reduced. As stated in section 3.1, CNN model was trained jointly for each landmark. In other words, for each landmark, kernel weights were initialized randomly. Instead of repeating same procedure over and over again, starting with kernel weights that is tailored to extract features from our training dataset, and then fine-tune the CNN for specific landmark would be more efficient.

So as to increase success rate on landmarks located on gray matter, besides data augmentation, making adjustments on CNN design such as adding more skip layers or increasing the depth by adding more layers can be tried. Another suggestion is to train CNN model with different patch sizes. It would be beneficial since it enriches training dataset with positive samples. Patch size is very important variable that is worth optimizing. In this study, selecting the patch size 16 in 3 dimensions was an educated guess. Although different patch sizes were tried, none of them outperformed the CNN model that takes 16 x 16 x 16 x 3 patches as input. It's related to structure of features in dataset and may need update if dataset or problem changes.

Training with multiple patch sizes at once, and altering the design of CNN in that matter would be helpful. There are proposed CNN models that takes multiple input with different sizes or processing the same input with different sized kernels. Nonetheless, at this point, it is hard to say that this approach will meet the expectations, and the gained results will worth the increased computational cost. Instead, making pre-defined parameters such as patch size,

kernel size, number of kernels dynamic and turning it into learnable parameter might be both efficient and computationally more affordable. Yet, it's not only out of scope of this study, it is a comprehensive task that requires being inspected from different aspects.



REFERENCES

1. Zitova B, Flusser J. Image registration methods: a survey. *Image and Vision Computing*. 2003;21(11):977–1000.
2. Milletari F, Navab N, Ahmadi SA. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*; 2016: IEEE.
3. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A. Face recognition: a literature survey. *ACM Computing Surveys (CSUR)*. 2003;35(4):399–458.
4. Toshev A, Szegedy C. DeepPose: Human pose estimation via deep neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2014: IEEE.
5. Murphy-Chutorian E, Trivedi MM. Head pose estimation in computer vision: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2008;31(4):607–626.
6. Tompson J, Goroshin R, Jain A, LeCun Y, Bregler C. Efficient object localization using convolutional networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition*; 2015; IEEE.
7. Ghayoor A, Vaidya JG, Johnson HJ. Robust automated constellation-based landmark detection in human brain imaging. *Neuro Image*. 2018;170:471–481.
8. Payer C, Stern D, Bischof H, Urschler M. Regressing heatmaps for multiple landmark localization using CNNs. *2016 International Conference on Medical Image Computing and Computer-Assisted Intervention*; 2016: MICCAI.
9. Zhu D, Li K, Guo L, Jiang X, Zhang T, Zhang D, et al. DICCCOL: Dense individualized and common connectivity-Based cortical landmarks. *Cerebral Cortex*. 2013;23(4):786–800.
10. Calabrese E. Diffusion tractography in deep brain stimulation surgery: a review. *Frontiers in Neuroanatomy*. 2016;10(1):45-57.

11. Markelj P, Tomazevic D, Likar B, Pernus F. A review of 3D/2D registration methods for image-guided interventions. *Medical Image Analysis*. 2012;16(3):642–661.
12. Yetkin AE, Hamamci A. Region proposal networks in domain generalization of mr landmark detection. *2017 21st National Biomedical Engineering Meeting (BIYOMUT)*. IEEE; 2017.
13. League D. Interactive, image-guided, stereotactic neurosurgery systems. *AORN Journal*. 1995;61(2):360–370.
14. Mori S. *Introduction to diffusion tensor imaging*. Baltimore: Elsevier; 2007.
15. Irfanoglu MO, Nayak A, Jenkins J, Hutchinson EB, Sadeghi N, Thomas CP, et al. DR-TAMAS: Diffeomorphic registration for tensor accurate alignment of anatomical structures. *NeuroImage*. 2016;132:439 – 454.
16. Hebb DO. The organization of Behavior: A neuropsychological theory. *Psychology Press*; 2005.
17. Minsky M, Papert S. *Perceptrons: an introduction to computational geometry*. Massachusetts: MIT press; 2017.
18. Clevert DA, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:151107289*. 2015.
19. Gurney K. *An introduction to neural networks*. London: CRC press; 2014.
20. Bachman D. *Advanced calculus demystified*. New York: McGraw Hill; 2007.
21. Tieleman T, Hinton G. CSC321 Lecture notes 2014 [cited 2019 12 July]. Available from: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.
22. Hinton GE, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Computation*. 2006;18(7):1527–1554.
23. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*. 2014;15(1):1929–1958.

24. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998;86(11):2278–2324.
25. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*; 2012: NeurIPS
26. Smith SW, et al. *The scientist and engineer's guide to digital signal processing*. California: California Tech; 1997.
27. Goodfellow I, Bengio Y, Courville A. *Deep learning*. London: MIT Press; 2016.
28. Yetkin AE, Hamamcı A. Data augmentation for head pose estimation from mri surface. *2016 Tıp Teknolojileri Ulusal Kongresi (TIPTEKNO)*; 2016: IEEE.
29. Lin M, Chen Q, Yan S. Network in network. *arXiv preprint arXiv:13124400*. 2013.
30. Newell A, Yang K, Deng J. Stacked hourglass networks for human pose estimation. *European Conference on Computer Vision*; 2016: Springer.
31. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*; 2016: IEEE.
32. Chollet F, et al. Homepage of Keras API 2015 [cited 2019 12 July]. Available from <https://keras.io>.
33. Sotiropoulos SN, Jbabdi S, Xu J, Andersson JL, Moeller S, Auerbach EJ, et al. Advances in diffusion MRI acquisition and processing in the Human Connectome Project. *NeuroImage*. 2013;80(1):125 – 143.
34. Glasser MF, Sotiropoulos SN, Wilson JA, Coalson TS, Fischl B, Andersson JL, et al. The minimal preprocessing pipelines for the Human Connectome Project. *NeuroImage*. 2013;80(2):105–124.
35. Ioffe S, Szegedy C. Batch Normalization: accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*; 2015: ICML.

36. Basser PJ, Pajevic S, Pierpaoli C, Duda J, Aldroubi A. In vivo fiber tractography using DT-MRI data. *Magnetic Resonance in Medicine*. 2000;44(4):625–632.
37. Garyfallidis E, Brett M, Amirbekian B, Rokem A, Van Der Walt S, Descoteaux M, et al. Dipy, a library for the analysis of diffusion MRI data. *Frontiers in Neuroinformatics*. 2014;8(2):8-26.
38. Jellison, Brian J., et al. Diffusion tensor imaging of cerebral white matter: a pictorial review of physics, fiber tract anatomy, and tumor imaging patterns. *American Journal of Neuroradiology*. 2004;25(3): 356-369.
39. Jackson, Aaron S., et al. Large pose 3D face reconstruction from a single image via direct volumetric CNN regression. *Proceedings of the IEEE International Conference on Computer Vision*; 2017: IEEE.
40. Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. U-net: convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*; 2015: Springer.

APPENDIX A: ETHICAL APPROVAL FORM



T.C. YEDİTEPE ÜNİVERSİTESİ

Sayı : 37068608-6100-15-1173
Konu: Klinik Araştırmalar
 Etik kurul Başvurusu hk.

25/02/2016

İlgili Makama (Andaç Hamamcı)

Yeditepe Üniversitesi Biyomedikal Mühendisliği Bölümü Doç. Dr. Andaç Hamamcı'nın sorumlu olduğu "**Beyin Görüntülerinin Bağlantılılık Temelli Çakıştırılması**" isimli araştırma projesine ait Klinik Araştırmalar Etik Kurulu (KAEK) Başvuru Dosyası (**1183** kayıt Numaralı KAEK Başvuru Dosyası), Yeditepe Üniversitesi Klinik Araştırmalar Etik Kurulu tarafından **24.02.2016** tarihli toplantıda incelenmiştir.

Kurul tarafından yapılan inceleme sonucu, yukarıdaki isimi belirtilen çalışmanın yapılmasının etik ve bilimsel açıdan uygun olduğuna karar verilmiştir (**KAEK Karar No: 586**).

Prof. Dr. Turgay ÇELİK

Yeditepe Üniversitesi
 Klinik Araştırmalar Etik Kurulu Başkanı