

AUTOMATIC WHOLE-BRAIN MAPPER FOR LOCALIZATION AND REGISTRATION

A Thesis

by

SHEELABHADRA DEY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee, Zhangyang Wang
Committee Members, Theodora Chaspari
Xiaoning Qian
Head of Department, Dilma Da Silva

May 2019

Major Subject: Computer Science

Copyright 2019 Sheelabhadra Dey

ABSTRACT

Brain imaging is an indispensable tool in neuroscience research. In many cases, research in this field involves capturing images of thin slices of animal brains using powerful high-resolution microscopes. The brain consists of billions of neurons interconnected in a highly sophisticated manner. To understand the role of a group of neurons in animal behavior, target neurons are fluorescently labeled. Then the brain images are compared to standard brain atlases to identify the spatial regions in the brain where these target neurons are expressed. Brain atlases consist of structural outlines labeling the different regions in the brain. Deforming an atlas to align the regions in a brain image is also known as image registration. In this work, we explore techniques to automatically identify the atlas that a brain image belongs to, register the atlas to the brain image, and count the number of expressed neurons in the sub-regions of the brain. The final deliverable is a software application capable of performing all the tasks with minimal human supervision.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my thesis advisor Dr. Zhangyang (Atlas) Wang of the Department of Computer Science and Engineering at the Texas A&M University for providing me the opportunity to pursue a master's thesis on this topic. He always encouraged me to come up with independent ideas and instilled in me an aptitude for research, traits that I am sure will help me as I progress in my career. My sincere thanks to Dr. Theodora Chaspari of the Department of Computer Science and Engineering and Dr. Xiaoning Qian of the Department of Electrical and Computer Engineering for being on my thesis committee inspiring me to produce quality work.

I am indebted to Dr. Jun Wang and Dr. Lee Shapiro of the Department of Neuroscience & Experimental Therapeutics for their valuable input on every step of my work. Without their encouragement, support, and critique this work would not have reached its current form.

Heartfelt thanks go to Himanshu and Yifeng for clarifying my queries on their brain imaging experiments and providing me access to every kind of data whenever I needed them.

I am thankful to Maggie for helping me lay the groundwork for this study. She has an infectious enthusiasm and a keen observation, qualities I will always try to inculcate within me. I am also thankful to Mahin for handling the progress of the work while I was away from school doing an internship for a semester.

I will be failing in my duty if I don't thank the student researchers in Dr. Wang's and Dr. Shapiro's labs for their valuable feedback while testing the iterations of the neuron counting application.

Special thanks to Siddharth, Subodh, Nishant, Himanshu and all my friends who supported me through the highs and lows during my time at grad school.

Finally, but most importantly, I am thankful to my parents for providing me with unfailing support and continuous encouragement throughout my years of study. But for their blessings, I wouldn't have come this far.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Dr. Zhangyang (Atlas) Wang and Dr. Theodora Chaspari of the Department of Computer Science and Engineering and Dr. Xiaoning Qian of the Department of Electrical and Computer Engineering. Dr. Jun Wang and Dr. Lee Shapiro of the Department of Neuroscience & Experimental Therapeutics oversaw the progress of the work providing their invaluable input during the development of the neuron counting application.

Maggie von Ebers of the Department of Computer Science and Engineering investigated the suitability of various feature detectors and data pre-processing methods for our task. Himanshu Gangal of the Department of Neuroscience & Experimental Therapeutics created the atlas dataset and provided code snippets for neuron counting. Dr. Yifeng Cheng of the Department of Neuroscience & Experimental Therapeutics provided access to all the data used for analysis. Mahin Ramzani, an academic collaborator, implemented the manual control point based localization method and created version 1 of the graphical user interface.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

Graduate study was supported by funding from part-time student worker positions offered by the Department of Computer Science and Engineering and the Department of Neuroscience & Experimental Therapeutics at the Texas A&M University.

NOMENCLATURE

AP	Anteroposterior
CNN	Convolutional Neural Network
DNN	Deep Neural Network
SVM	Support Vector Machine
RF	Random Forest
MR	Magnetic Resonance
MSER	Maximally Stable Extremal Regions
SIFT	Scale Invariant Feature Transform
SURF	Speeded-up Robust Features
MI	Mutual Information
CC	Cross Correlation
HOG	Histogram of Oriented Gradients
CP	Control Point
TPS	Thin Plate Spline
MNIST	Modified National Institute of Standards and Technology
FCN	Fully Convolution Network
SSD	Sum of Squared Difference
VGG	Visual Geometry Group
k -NN	k -Nearest Neighbor
MATLAB	Matrix Laboratory

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGMENTS	iii
CONTRIBUTORS AND FUNDING SOURCES	iv
NOMENCLATURE	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	viii
1. INTRODUCTION.....	1
1.1 Brain Imaging	1
1.2 Challenges.....	5
1.3 Research Questions	5
1.4 Organization of the Thesis.....	7
2. RELATED WORK	8
2.1 Medical Image Retrieval.....	8
2.2 Medical Image Registration	8
3. BRAIN ATLAS LOCALIZATION.....	13
3.1 Manual Interest Point Placement	13
3.2 Data Pre-processing	16
3.3 Key-point Extraction.....	17
3.4 Deep Features	18
3.4.1 Global Deep Features	19
3.4.2 Local Deep Features	21
4. BRAIN ATLAS REGISTRATION	26
4.1 Geometric Image Transformation	26
4.2 Image Warping	28
4.3 Manual Control Point Placement.....	28
4.4 Automatic Control Point Placement	29
4.4.1 Harris and SURF Features-based Registration	32

4.4.2	Local Deep Features-based Registration	32
5.	NEURON COUNTING	37
5.1	Manual Neuron Identification	37
5.2	Automatic Neuron Counting	38
6.	EVALUATION	40
6.1	Total Percentage Counting Accuracy	40
6.2	Total Region-wise Error	40
7.	USER INTERFACE.....	43
8.	CONCLUSIONS AND FUTURE WORK.....	46
8.1	Findings	46
8.1.1	Findings having Positive Impact	46
8.1.2	Findings having Negative Impact	48
8.2	Strengths.....	50
8.3	Limitations	51
8.4	Future work	51
8.4.1	Next Steps	51
8.4.2	Long-term Research Scope	51
	REFERENCES	52

LIST OF FIGURES

FIGURE	Page
1.1 Schematic diagram of the mouse brain (reprinted from [1])	2
1.2 Naming convention for mouse brain slices (reprinted from [2]).....	2
1.3 Sample mice brain slice images	3
1.4 Manual atlas alignment	4
1.5 Sample brain image and atlas pairs	6
3.1 Candidate atlases for a given analysis image	14
3.2 SSD curve for atlas retrieval.....	15
3.3 Interesting regions in brain images.....	18
3.4 VGG16 architecture (reprinted from [3] ©2017 IEEE)	20
3.5 Confusion matrix for atlas retrieval using global features	22
3.6 Illustration of windows (yellow dashed lines) drawn around key-points (green dots).	23
3.7 Flow chart for atlas localization	25
3.8 Confusion matrix for atlas retrieval using the 2 stage approach	25
4.1 Control point selection tool.....	29
4.2 Control points placed using selection tool	30
4.3 Results of manual control points-based registration	31
4.4 Control points generated automatically	33
4.5 Flow chart for automatic registration	34
4.6 Results of automatic control points-based registration.....	35
4.7 Result of local deep features-based registration	36
5.1 Sample analysis image with expressed neurons	37

5.2	Illustration of atlas overlaid on a neuron image	38
5.3	Flow chart for neuron counting process	39
6.1	Total percentage accuracy	41
6.2	Total region-wise counting error	42
7.1	Analysis and neuron image input	43
7.2	Matching atlas selection	44
7.3	Control point selection tool.....	44
7.4	Selected control points.....	45
7.5	Atlas registered to analysis image.....	45

1. INTRODUCTION

1.1 Brain Imaging

Brain imaging or neuroimaging has been a powerful tool used in neuroscience research. It is the use of various techniques to either directly or indirectly image the structure, function, or pharmacology of the nervous system. This enables neuroscience researchers to observe the response of regions in the brain to different stimuli.

Researchers in the labs headed by Dr. Jun Wang and Dr. Lee Shapiro of the Department of Neuroscience & Experimental Therapeutics at the Texas A&M University analyze rodent brains. A schematic diagram of a mouse brain highlighting the major regions in the brain is shown in figure 1.1. The left and right sides of the brain image shown belong to the anterior and posterior portion of the brain respectively. The subjects used in their experiments are rats and mice. One of their research projects involves understanding drug addiction; alcohol addiction in particular. The dorso-medial striatum in the brain has been linked to reward based learning and addiction. Neural activity in this region has been shown to be upregulated during addiction in rodent models. To validate these results, different genetic tools are used to target neural populations that send synapses to a desired neural population. One such technique is rabies tracing which aims to label the target neurons using specific colors. The rabies virus is infused into a rodent and its brain neurons are visualized by sacrificing it a week after the infusion. Using a cryostat the brain of the rodent is cut into thin slices each having a width roughly equal to 50 microns. The naming convention for the slices that we followed in this study is shown in figure 1.2. The brain sections were cut along the plane perpendicular to the plane of the figure. The slices were named according to their distance from the bregma which is indicated by the axis at the top in the figure. The distance of the brain slice from the bregma is called anteroposterior¹ (AP) number or atlas ID. The brain slices are mounted on a slide to perform confocal microscopy which gives high resolution images of the brain slices. Examples of brain slice images obtained using this process is shown in figure

¹In anatomy, anteroposterior means from front to back.

1.3.

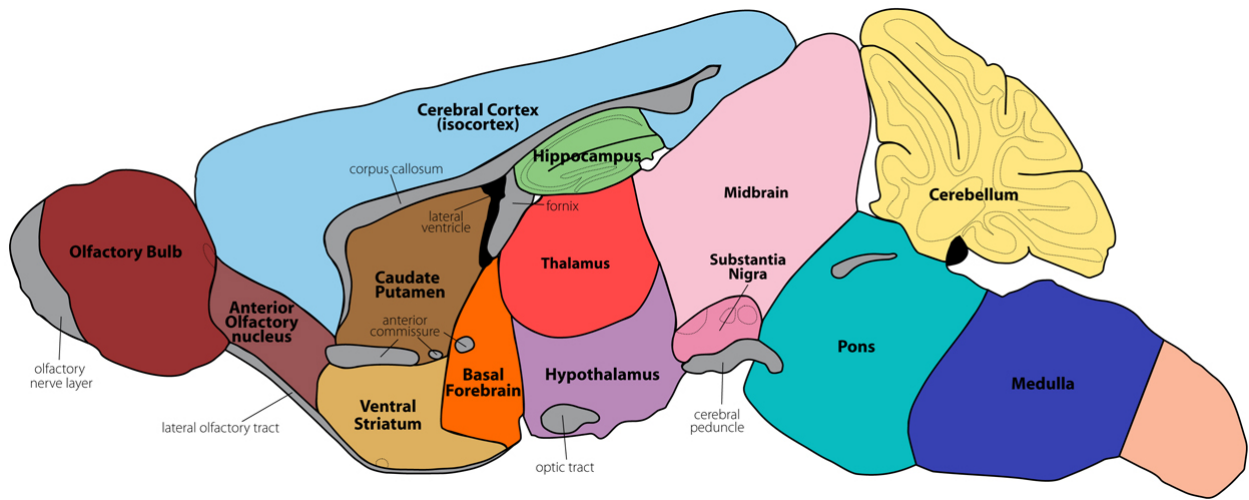


Figure 1.1: Schematic diagram of the mouse brain (reprinted from [1])

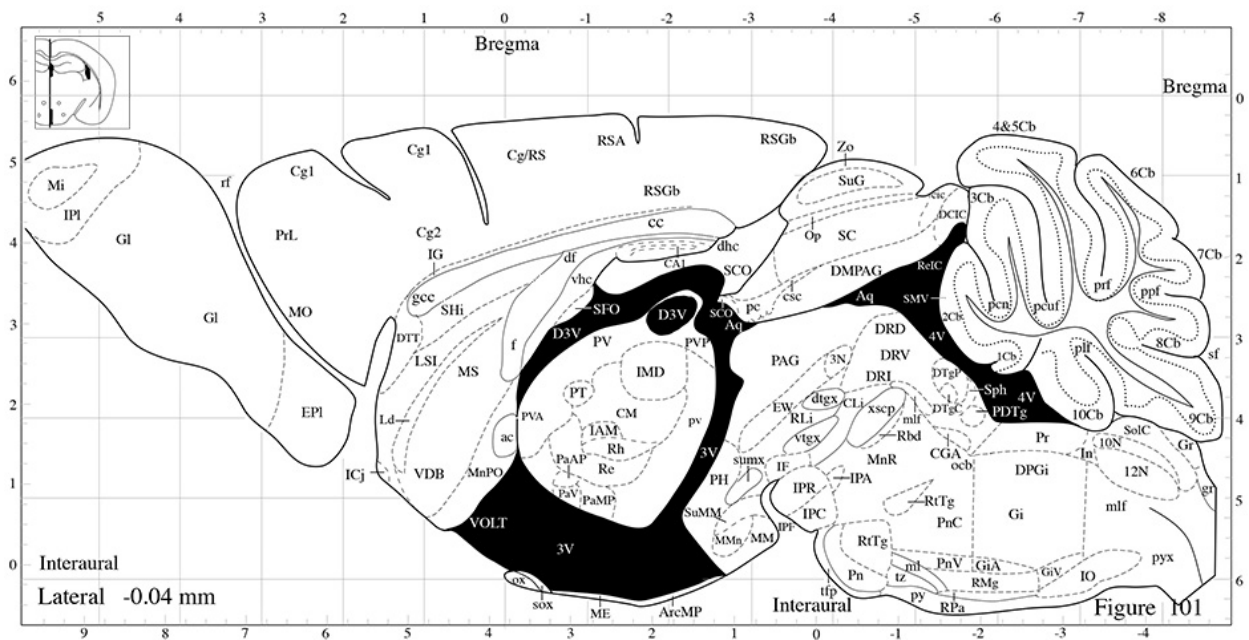
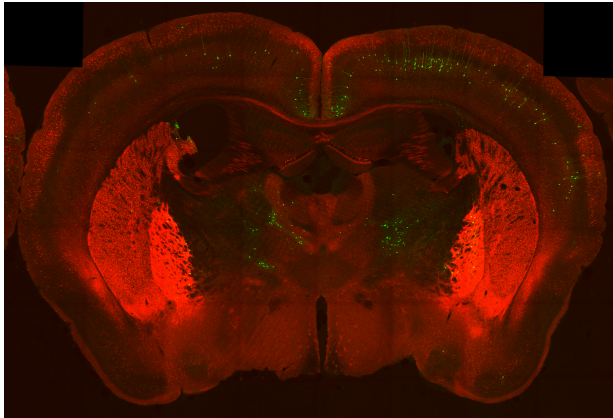
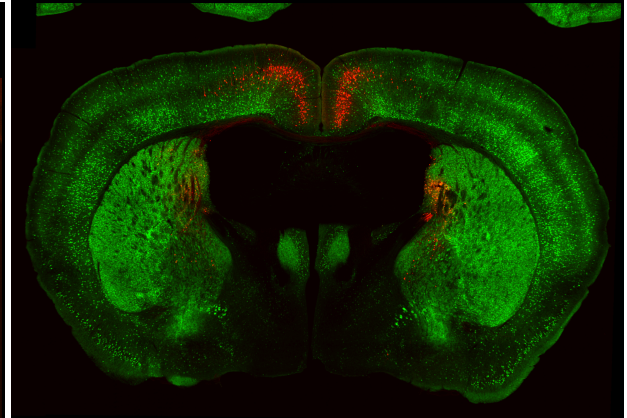


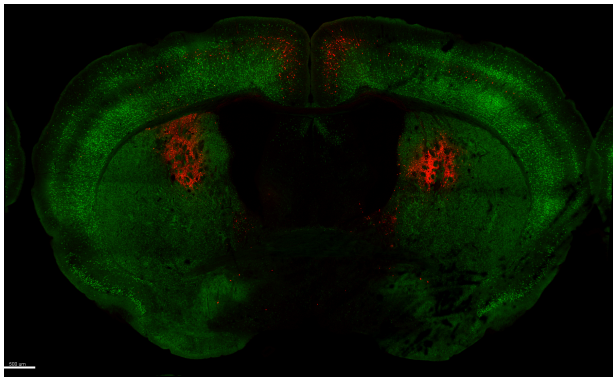
Figure 1.2: Naming convention for mouse brain slices (reprinted from [2])



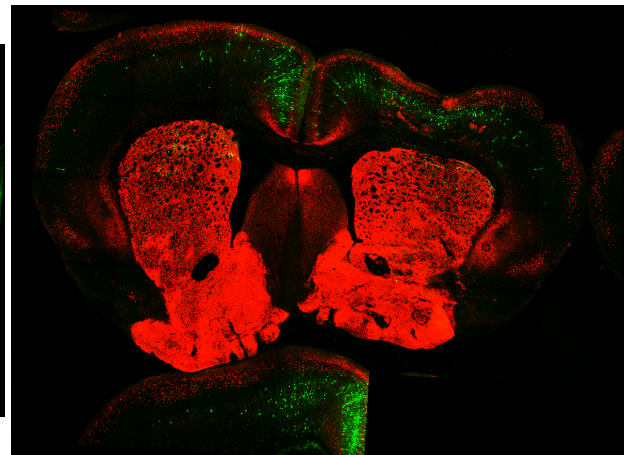
AP -0.94 mm



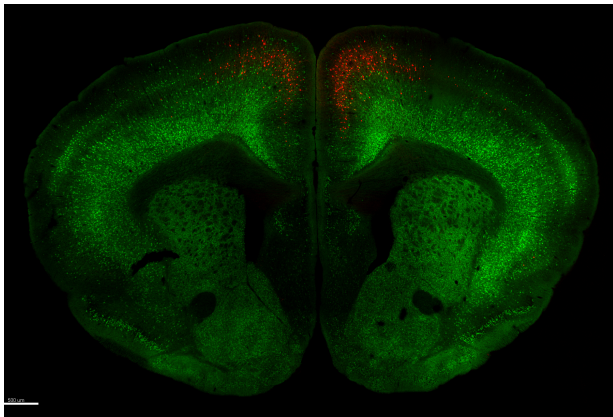
AP -0.22 mm



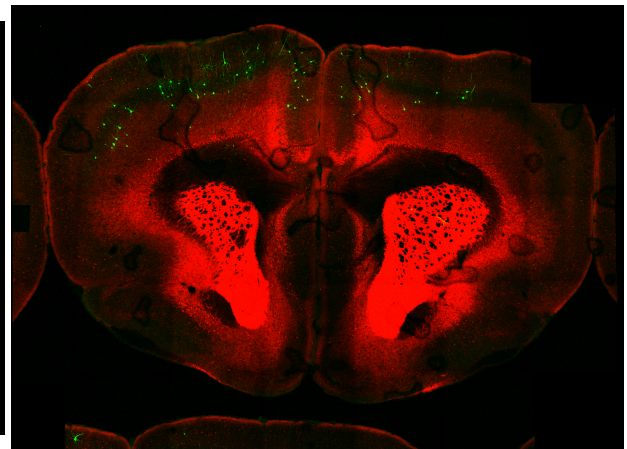
AP -0.22 mm



AP 0.86 mm



AP 1.54 mm



AP -1.54 mm

Figure 1.3: Sample mice brain slice images

The brain image slices with the labeled neurons are used for analysis. The major goal of this procedure is to identify the spatial regions in the brain where the labeled neurons are expressed. This is achieved by taking line drawings illustrating the sub-regions of the image. In neuroscience literature these are called atlases. These atlases are overlapped on the brain slice images to identify the spatial regions in the brain where the expressed neurons are present. To achieve this, researchers in both the laboratories overlay the atlases on their corresponding brain slice images using software applications such as Adobe Photoshop. In most of the cases the brain images and their corresponding atlases don't align perfectly. This could be attributed to the variation in the structure of the brain from one subject to another and deformation introduced during the brain cutting procedure. The solution currently being employed to overcome the challenge is to manually deform the atlas to align it with the outer contour and the internal structures of the brain slice images using Adobe Photoshop. The atlas generated after manual deformation is shown in Figure 1.4. Figure 1.5 shows a few sample brain images with their corresponding matching atlases.

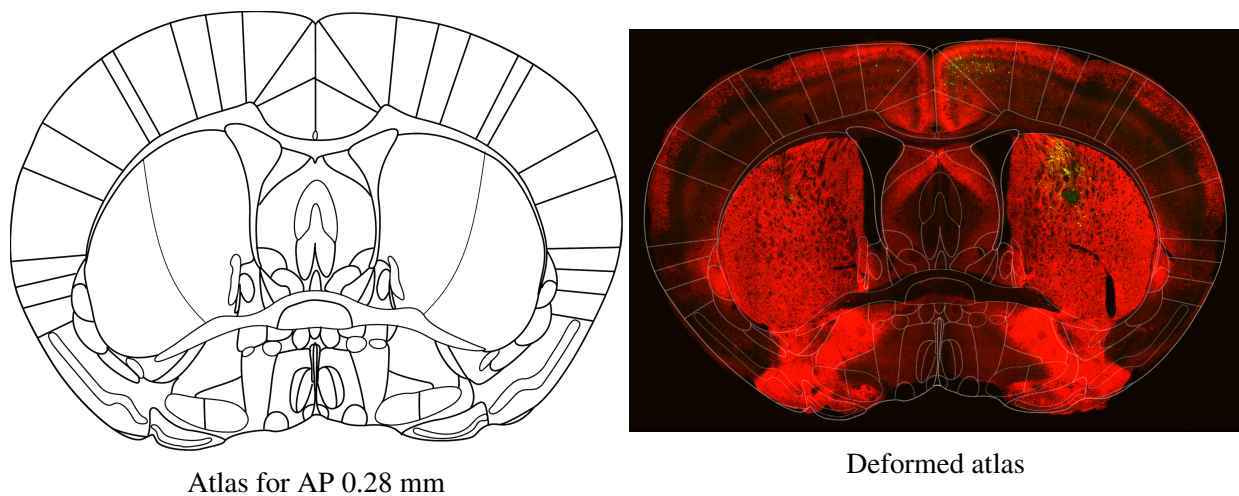


Figure 1.4: Manual atlas alignment

After aligning the atlas with the brain image, researchers manually mark the labeled colored neurons in the brain. This step requires an expert as it can be difficult for an algorithm to figure out

the number of neurons within a cluster of colored neurons. The number of neurons in each region of the deformed atlas is then counted manually. The number of marked neurons in a brain image can vary between 100 and 300.

1.2 Challenges

Although the process works well, it is extremely time-consuming and error-prone. It takes about 15 – 30 minutes to manually deform an atlas to align it with its corresponding brain slice image. Photoshop also doesn't allow users to make minute modifications to the fit as the spacing between the grid lines is fixed. This makes the task cumbersome and introduces error in the fit.

The large number of marked neurons makes the process of neuron counting extremely time consuming and error-prone. The error may be introduced due to human involvement in the counting process which is tedious taking close to 30 minutes. The error in the count of neurons can vary between 0 and 50 neurons.

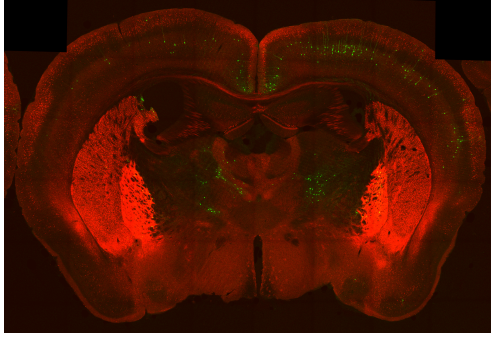
In this work we have addressed the following three major challenges faced by the researchers in the laboratories with regards to neuron counting:

1. Identifying the correct atlas for a given brain slice image.
2. Registering the atlas to the brain slice image.
3. Counting the number of expressed neurons in the sub-regions of the brain slice image.

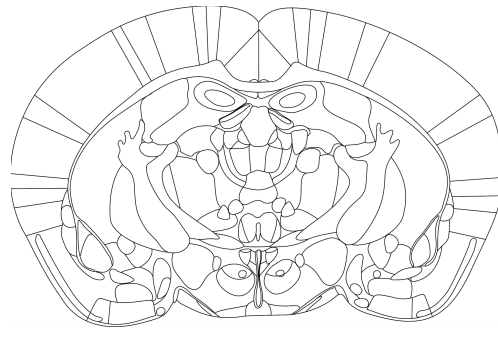
1.3 Research Questions

The purpose of this study is to develop a framework that assists neuroscientists by reducing the time and the discrepancy in neuron counting. Given a brain slice image for analysis, we formulated the research questions as listed below:

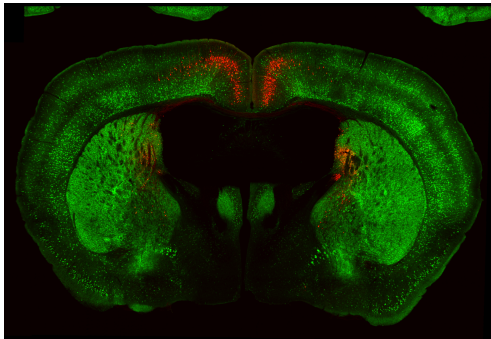
1. Which are the feature-based methods suitable for retrieving its matching atlas?
2. How can feature-based methods be used for registering the atlas to the image?
3. How can we automatically count the number of labeled neurons?



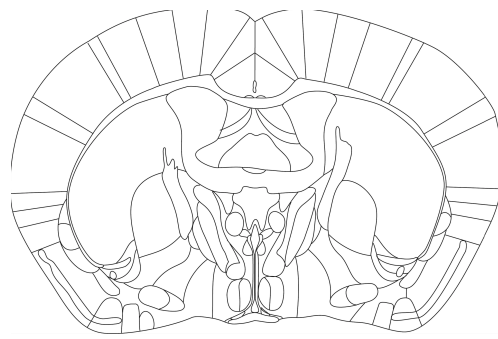
Brain image for AP -0.94 mm



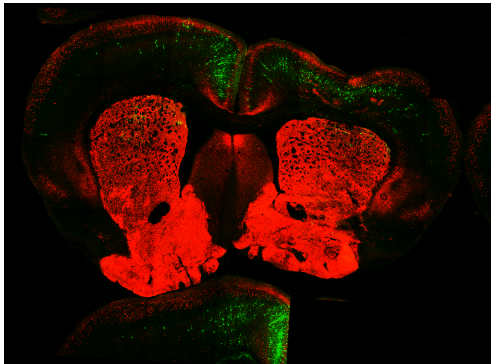
Atlas for AP -0.94 mm



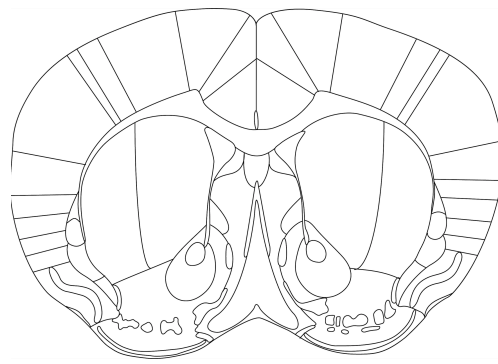
Brain image for AP -0.22 mm



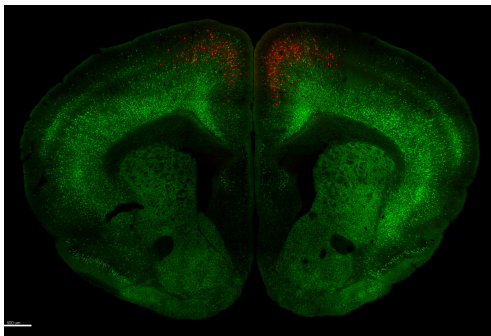
Atlas for AP -0.22 mm



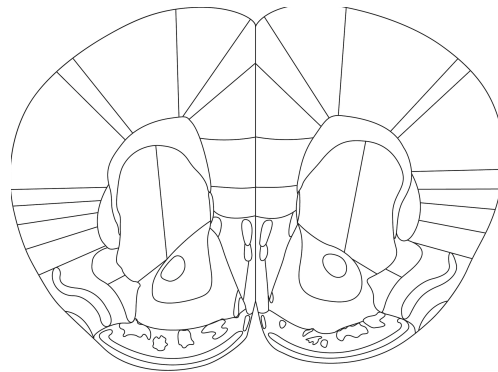
Brain image for AP 0.86 mm



Atlas for AP 0.86 mm



Brain image for AP 1.54 mm



Atlas for AP 1.54 mm

Figure 1.5: Sample brain image and atlas pairs

1.4 Organization of the Thesis

The remainder of the thesis is organized into 7 sections. Section 2 provides a bird's eye-view of the existing literature addressing medical image retrieval and medical image registration. Section 3 explains our approach for automatic brain image retrieval. Section 4 talks about the methods we adopted for automatic image registration. Section 5 is a brief discussion on how we performed the region-wise automatic counting. Section 6 describes our evaluation metrics. Section 7 contains snapshots of the user interface. Section 8 addresses in detail our findings during the course of the study, the strengths and limitations of our approach and presents possible future research directions.

2. RELATED WORK

This chapter discusses the existing work related to medical image retrieval and medical image registration.

2.1 Medical Image Retrieval

The retrieval techniques relevant to our problem are content based. Content based techniques can consist of global features or local features or both. In this section we look at content-based retrieval and classification methods pertinent to medical images.

[4], [5], [6] used the texture, color, and gray level features for content-based retrieval. Local and global features were employed in [7]. It used color histograms and Gabor filters that are used for texture classification for the image retrieval task.

Convolutional neural networks (CNNs) have been shown to achieve astonishing results on image classification problems as can be seen in [8], [9], [10], [11], [12]. The deep neural networks (DNNs) developed in these works obtained state-of-the-art classification results on the ImageNet [13] dataset. [14] used a modified pre-trained GoogLeNet [10] to determine if the orientation of the chest X-ray images belonged to the frontal or lateral view. This approach showed that CNNs pre-trained on a widely different dataset from medical images are robust enough to perform simple classification tasks related to medical images. [15] used a modified network introduced in [12] called CheXNet to classify 14 different types of diseases observed in chest x-rays. The network was trained on 112,000 images from the ChestXray14 dataset [16] achieving state-of-the-art performance on the test data. [17] used CNNs along with support vector machine (SVM) and random forest (RF) classifiers to determine whether lung nodules were benign or malignant. An extensive study on the use of DNNs for the analysis of medical images can be found in [18].

2.2 Medical Image Registration

Medical image registration is a field that has been thoroughly studied over the past few years. [19] used a hierarchical transformation approach for breast Magnetic Resonance Image (MRI)

registration. The MR images were contrast enhanced and normalized mutual information was used to measure the similarity between pixels of the images being registered. Affine transformation was used to capture the global motion between the images. B-splines were used to model the local motion. [20] added spatial information based on the gradients of the images to be registered to make the registration more robust.

When the images to be registered have been obtained in vastly different modalities, finding pixel-wise correspondences between them becomes challenging. In these cases, it is much more pragmatic to use interesting points and regions that are present in both the images. Then a transformation function mapping the set of points in both the images can be used for registration.

Region-based features are detected using segmentation methods. [21] presented maximally stable extremal regions (MSERs) based on homogeneity of image intensities. Elongated anatomic structures described in [22] were used as line features for medical images. Canny edge detector [23] and Laplacian of Gaussian [24], which are methods for edge detection were used for line feature detection. Point feature detectors are often used to find corners in images due to their invariance to imaging geometry. Harris corner detector [25] is a widely popular algorithm to detect corners in images. Scale invariant feature transform (SIFT) [26] is a local feature detector and descriptor that identifies key-points by finding the maxima and minima of the result of difference of Gaussians function applied at various scales to a series of smoothed and resampled images. Speeded-up robust features (SURF) [27] is another feature detector and descriptor that is based on similar principles and steps as SIFT. The authors claim that it is much faster than SIFT and more robust against various image transformations.

Area-based feature matching techniques such as mutual information (MI) are popular for multimodal image registration. Mutual information is a quantity that measures how much a random variable tells us about another random variable. It can be thought of as the reduction in uncertainty about one random variable given knowledge of another. High mutual information indicates a large reduction in uncertainty; low mutual information indicates a small reduction; and zero mutual information between two random variables indicates that the variables are independent.

More information about the intuition and math behind mutual information can be found here: http://www.scholarpedia.org/article/Mutual_information. [28] contains a thorough discussion on the use of mutual information for registration of medical images. [29], [30], [20], [31] are some of the most significant works that have used mutual information and its variants for image registration.

Feature-based matching methods involve finding correspondences between the set of features obtained on both the images to be registered. Invariant feature descriptors are useful in creating descriptors that are not sensitive to small deformations. [32] proposed a method to select features according to the reliability of their possible matches. [33] and [34] use feature descriptors based on the image intensity in the neighborhood of the feature points. The feature matching was done by calculating the cross-correlation (CC) between the feature descriptors. CC measures the similarity between the intensities in the pixels in 2 windows. A more robust metric is normalized CC [35] which is less sensitive to changes in intensities between the 2 windows. [36] used histogram of oriented gradients (HOG) [37] for creating feature descriptors. This made the feature descriptors less sensitive to changes in intensities and rather focused on the orientation of the edges around the feature point. SIFT feature descriptors are invariant to uniform scaling, orientation, illumination changes, and partially invariant to affine distortion. The descriptors are created by considering the magnitude and orientation values of samples within a window of size 16×16 around a key-point. SURF feature descriptors are also created by using a window around a key-point and Haar wavelet [38] responses are extracted from these windows.

In practice, points that are invariant and discriminative are employed as feature points. In conclusion, area-based feature matching is preferred when both the images to be registered do not contain many details and the distinctive information is obtained from the pixel intensities. On the other hand, feature-based matching methods are generally employed when local structural information dominates the image intensity information. They are also helpful in registering images that have entirely different modalities and contain local distortions between the images.

After the feature matching step the feature points are treated as the control points (CPs) for

finding the transformation or mapping function. The reference image is transformed and overlaid on the analysis image. The transformation function is determined using 2 methods: global transformation and local transformation. The global transformation function is found using a global model which uses all the CPs to estimate one set of parameters for the transformation function for the entire image. The local transformation function is determined by estimating parameters for small patches in the images. The patches are defined by the location of the CPs. Affine transforms, perspective transforms, and polynomial transforms and their variations were used in approaches covered in [39] to find the global transformation function. However, global transformations have a hard time handling images that contain local deformation. [40], [41], [42], [43] showed better registration performance using locally sensitive registration methods. These methods introduced slight variation to the process of global transformation estimation by adding weighted least square and weighted mean methods [40]. Thin plate splines (TPS) [44] and B-splines belong to the class of radial basis functions that have been shown to be effective for registration of images having local deformation. [45] discusses the application of TPS in medical image registration. [19], [46], [47] have shown that B-splines perform well in image registration. Recently, [36] has used a technique to find the brain slice cutting angle and have used bilinear model instead of B-splines for registration. The authors used a multi-level grid point placement strategy to handle large and small deformations.

Recent works have focused using deep learning techniques for medical image registration. Using deep learning techniques in our case is not possible as of now due to the lack of availability of a large dataset. For the sake of completeness, we will discuss some of the work here that have garnered significant attention from the medical imaging community. [48] introduced a CNN that can be trained end-to-end in an unsupervised manner for image registration. The dataset in this case consisted of unlabelled MNIST [49] and cardiac MR images. The CNN regressor maximized the normalized CC between the deformed image and the corresponding reference image. [50] used fully convolutional networks (FCNs) to learn the transformations between the images to be registered directly without knowing the transformation functions. It did so by estimating

the transformations between the images to be registered by maximizing an image-wise similarity metric between them. The authors in [51] trained a CNN for 3D medical image registration the parameters of which were used for registering a pair of new images.

Although, there are a plethora of research articles on image registration, the novelty in our case is the data that we are dealing with. As far as our knowledge, there haven't been any approaches that have aimed to solve the problem that we are aiming to tackle in this study. We combined approaches mentioned in this section and experimented with a few techniques of our own. The subsequent chapters will dive into the details of our approach and implementation.

3. BRAIN ATLAS LOCALIZATION

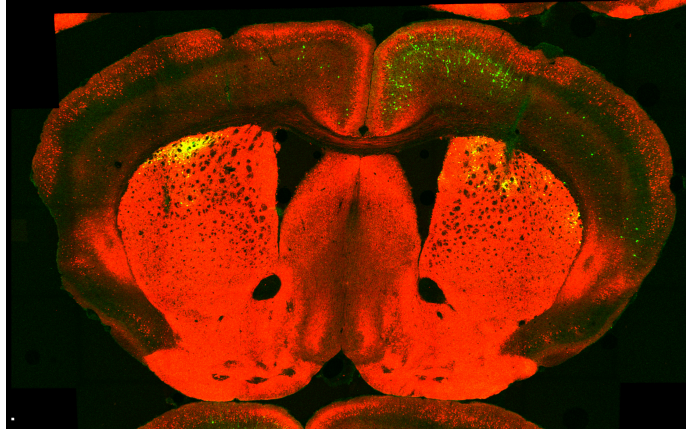
In this section we discuss the approaches we used for the automatic retrieval of brain atlases corresponding to the given brain slice images. The major motivation behind addressing this challenge was that it requires plenty of practice for a human to match a given analysis image to its atlas. This can be seen in figure 3.1, it can be extremely hard even for a human without sufficient amount of subject matter expertise to assign adjacent brain image slices to their exact atlases.

3.1 Manual Interest Point Placement

Initially for experimentation, we used a crude technique to find the matching atlas for a given query or analysis image. We manually placed 45 points of interest on the images of the atlas and saved them in a database. The user was asked to place the same number of points of interest on the analysis image. We then calculated the sum of squared distance (SSD) between the points placed on the analysis image and the points placed on all the atlases in the database. SSD is a metric used to find the similarity between two arrays. It is proportional to the Euclidean distance between the two arrays. It is calculated by taking the sum of squares of differences of corresponding values in the two arrays. If x and y represent the two arrays both containing N elements, then the mathematical expression for SSD is given by 3.1.

$$SSD = \sum_{i=1}^N (x_i - y_i)^2 \quad (3.1)$$

The AP number for which the SSD was the lowest among all the atlases was considered as the best matching AP number for the analysis image. Figure 3.2 shows the SSD curve for all the atlases with respect to a random analysis image. In this example, the analysis image belongs to AP 0.98 mm. The horizontal axis contains the list of AP numbers present in the database while the vertical axis has the SSD values for each of the AP numbers. In this example SSD is the lowest for AP 1.18 mm followed by the correct AP 0.98 mm. But, at the same time it can be seen that there are multiple troughs or local minima obtained at other AP numbers too. This could be major concern



Analysis image

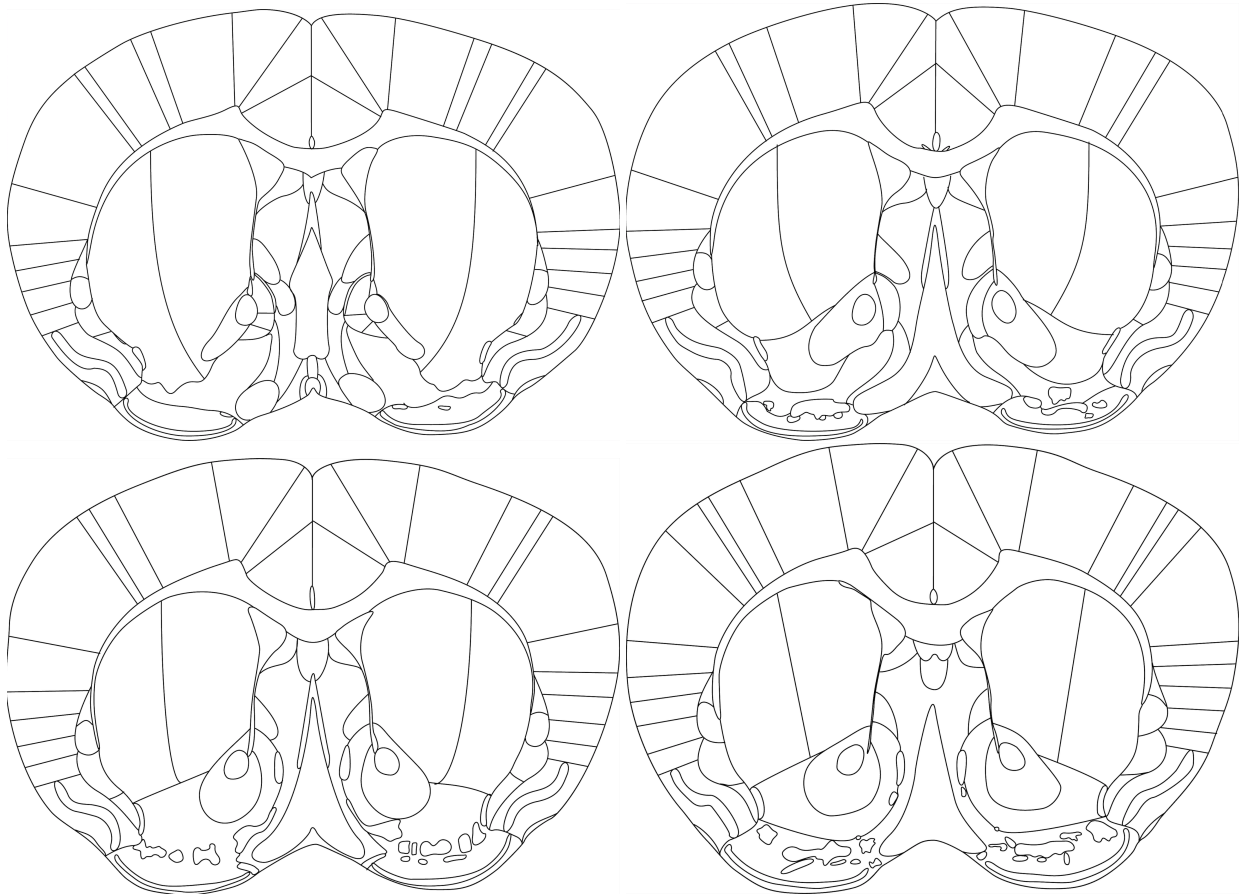


Figure 3.1: Candidate atlases for a given analysis image

if there are major deformations while cutting the brain slices. In these cases there could be local minima obtained at multiple Ap numbers and the AP number for which the global minimum occurs might be far away from the correct match. Handling deformations was a major important aspect

that had to be taken into consideration since obtaining perfect brain slices during the brain cutting process is not always possible. Also, manually placing points of interest on the analysis image is tedious and time-consuming. It is also subjective as the placement of the points of interest depends on the discretion of the user. This made the results inconsistent making it vary from user to user.

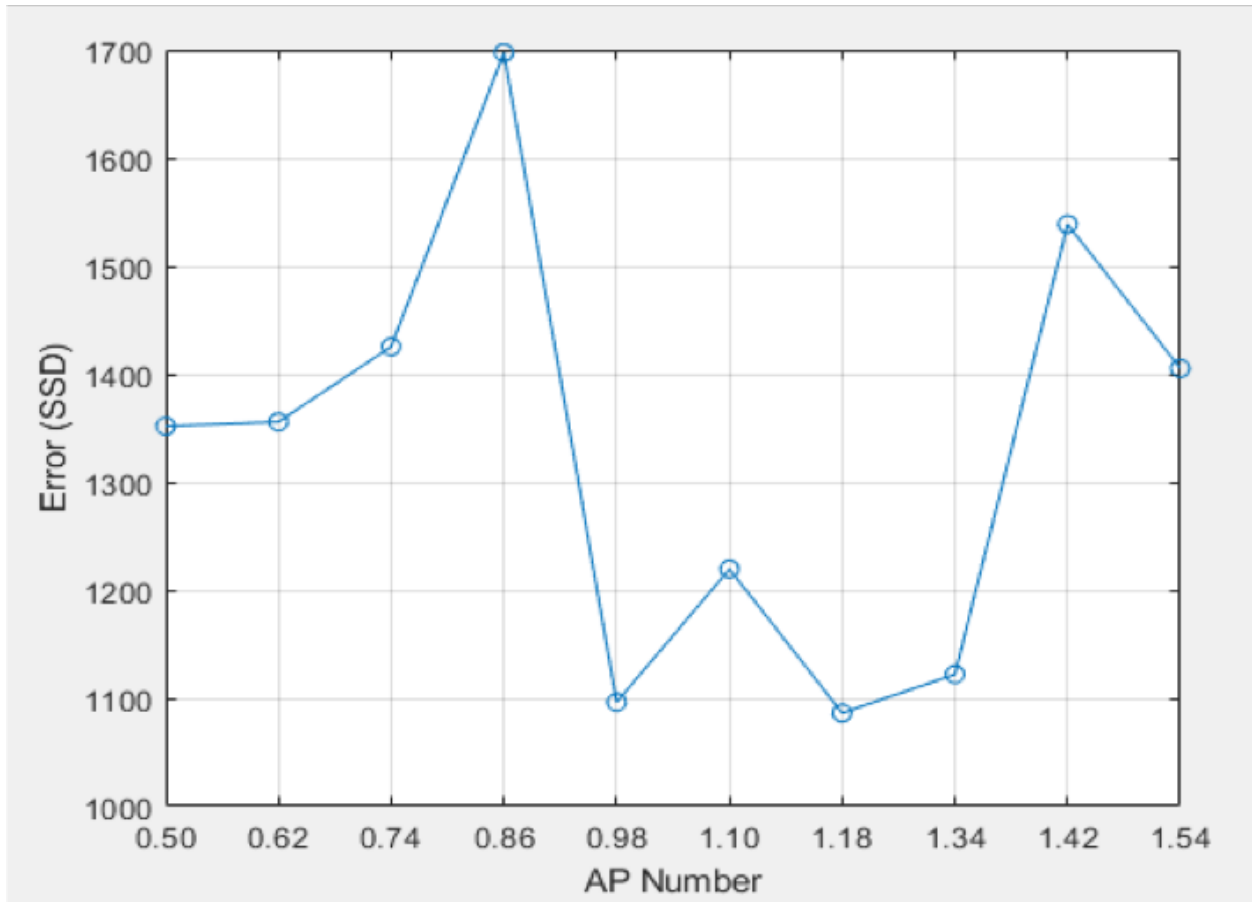


Figure 3.2: SSD curve for atlas retrieval

The limitations of the approach mentioned in this section led us to explore methods that are more robust to deformations. One of our major goals was also to minimize the amount of human effort involved in the atlas retrieval process. The rest of the sections in this chapter explain our approach to make the brain atlas retrieval process fully automatic.

3.2 Data Pre-processing

The task of directly identifying the correct matching atlas for a given an analysis image is not very straightforward. Part of the reason is that analysis images look very different from atlas images. The analysis images in our case are brain slice images stained in different colors. Figure 1.3 contains examples of a few analysis images. As can be observed, these images vary a lot in the way they are stained. The level of staining and contrast in the images is not consistent. This heavily depends on the goal of the experiments conducted by neuroscience researchers.

We performed a few data cleaning and pre-processing operations on the images to make them more suitable for image retrieval. First of all, instead of directly identifying the matching AP number for a given analysis image we assigned a few brain slice images belonging to the AP number and referred to them as reference images. Secondly, we removed all the spurious images from the background that weren't of any interest to us. As explained in the previous paragraph, color in our case doesn't provide any additional information. In some cases color may even obfuscate algorithms since similar images may have different colors depending on the staining. So, we converted all the brain images into single channel grayscale images. In the next step, we performed contrast limited adaptive histogram equalization (CLAHE) on all the grayscale images to improve their contrast. CLAHE is a variant of adaptive histogram equalization (AHE) in which the contrast amplification is limited to reduce the problem of noise amplification. Histogram equalization is technique in which the contrast of an image is improved by distributing the intensity values of the pixels over a wider range of intensity values. The method aims to transform the intensity values at each pixel location in such a manner that the distribution of the intensity values in the intensity histogram is as close to a uniform distribution. AHE computes several histograms on smaller sections of the image instead of the whole image and then redistributes the intensity values of the image. This makes AHE suitable for improving the contrast over a local region. The downside of AHE is that it has a tendency to overamplify noise in relatively homogeneous regions of an image. CLAHE addresses this issue by limiting the contrast amplification by clipping the intensity histogram at a predefined value. The dataset consisting of the contrast enhanced images are then

split into training and test sets. The training set consisted of 80% of the examples and the test set included the remaining 20% of the examples. Specifically we had 8 images per AP number in the training dataset and 2 images per AP number in the test dataset. We applied heavy data augmentation to the training and test dataset. The data augmentation included randomly applying varying levels of brightness, lateral flipping, and non-rigid deformations. This was achieved using an open source implementation for applying data augmentation on images which can be accessed here: <https://github.com/mdbloice/Augmentor>. Our final dataset consisted of 50 examples in the training dataset and 10 examples in the test dataset.

3.3 Key-point Extraction

Images with rich content contain interesting points that can be used as one of the metrics in comparing similar images. Extracting these interesting points have been shown to be helpful in image retrieval tasks. As can be seen in the brain images we are working with, there are not a lot of interesting points that can be identified as key-points. Barring the presence of ventricles, anterior commissure, and the external capsule, all the brain images have the same texture. Figure 3.3 highlights the interesting regions and locations in 2 sample brain images.

Harris corner detector and SIFT are popular feature detectors widely used to identify interesting points in images. Harris corner detector identifies corners in an image. Corners are regions in the image with large variation in intensity in all the directions. Recently SIFT has been patented and is not available in MATLAB. Hence, we used the SURF detector instead to identify interesting points. SURF is in general less accurate but much faster than SIFT. Although few in number, we obtained matching key-points on the analysis image and the reference brain images in the database. This was obtained by constructing feature descriptors around the points of interest (key-points). Reference images with more feature descriptor matches with the analysis image were further used to narrow down the search region for the matching reference image. The locations of the feature point matches were stored to be used in the subsequent steps.

In the next section we explain in detail the final step of our approach in the image retrieval process.

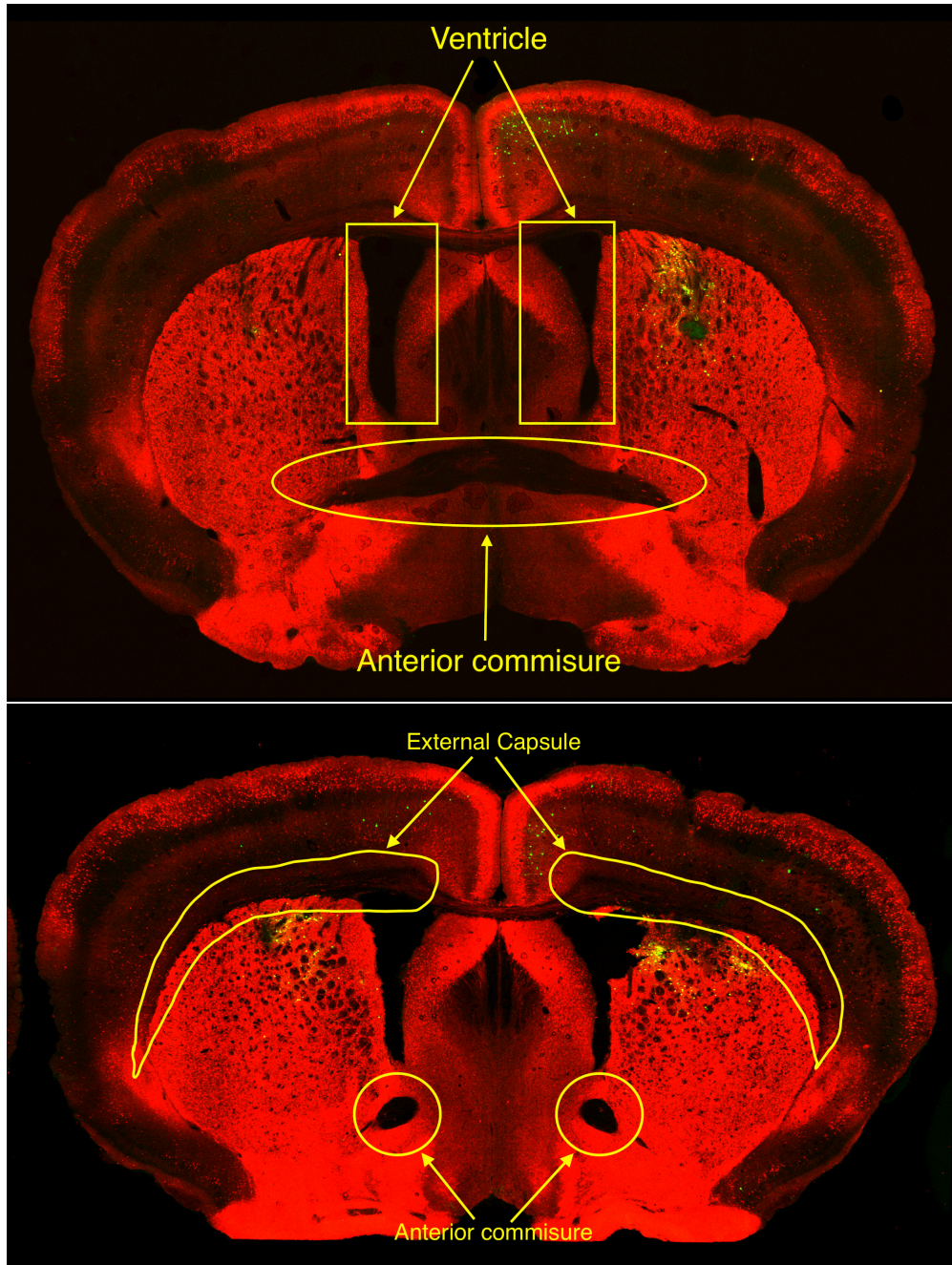


Figure 3.3: Interesting regions in brain images

3.4 Deep Features

CNNs have achieved state-of-the-art results on various image classification tasks. Although our problem can also be termed as an image classification problem, the variations in our images

are much more subtle. This is especially true if the thickness of the brain image slices is small. This makes it very tough for CNNs to match the analysis image to its exact reference image by only looking globally at the images.

A major drawback in our case that heavily limited the applicability of applying CNNs to the brain images is the lack of availability of a large similar dataset. This reduced our scope of using deep learning methods to solve our problem.

Despite the challenges outlined above, we experimented with a few approaches using pre-trained CNNs. Specifically, we used a VGG-16 CNN [9] pre-trained on the ImageNet [13] dataset. ImageNet is a publicly available large dataset that contains images of objects usually encountered by us in everyday life. It contains over 1.2 million images belonging to 1000 categories. It is widely used in academia as the de facto benchmark used to report the performance of computer vision classification algorithms. VGG-16 achieved 92.7% top-5 test accuracy on the ImageNet dataset. The architecture of VGG-16 is shown in figure. The default size of the input to a VGG-16 network is a 3-channel 224×224 RGB image. It consists of 16 weight layers using convolutional filters of dimensions 3×3 pixels. The convolutional stride used in all the layers is 1 pixel. Padding is applied after each convolutional layer to preserve the spatial resolution of the input. 5 max-pooling layers perform spatial pooling. The window size used for spatial pooling has a size of 2×2 pixels and a stride of 2 pixels. The convolutional layers are followed by 3 fully connected layers which are responsible for classification. The final fully connected layer is the softmax layer which gives the probability scores for each class. All the hidden layers contain the ReLU [52] activation function which introduces non-linearity into the model.

3.4.1 Global Deep Features

To make the brain slice images suitable for use with VGG-16, we first converted all the grayscale single channel images to 3 channel images. This was done by stacking together 3 replicas of a grayscale image. We resorted to this approach only to match the input requirements of VGG-16 as the network expects 3 channel images as input. This was followed by resizing the images to a size of $224 \times 224 \times 3$ which is the input image dimension expected by VGG-16. Since

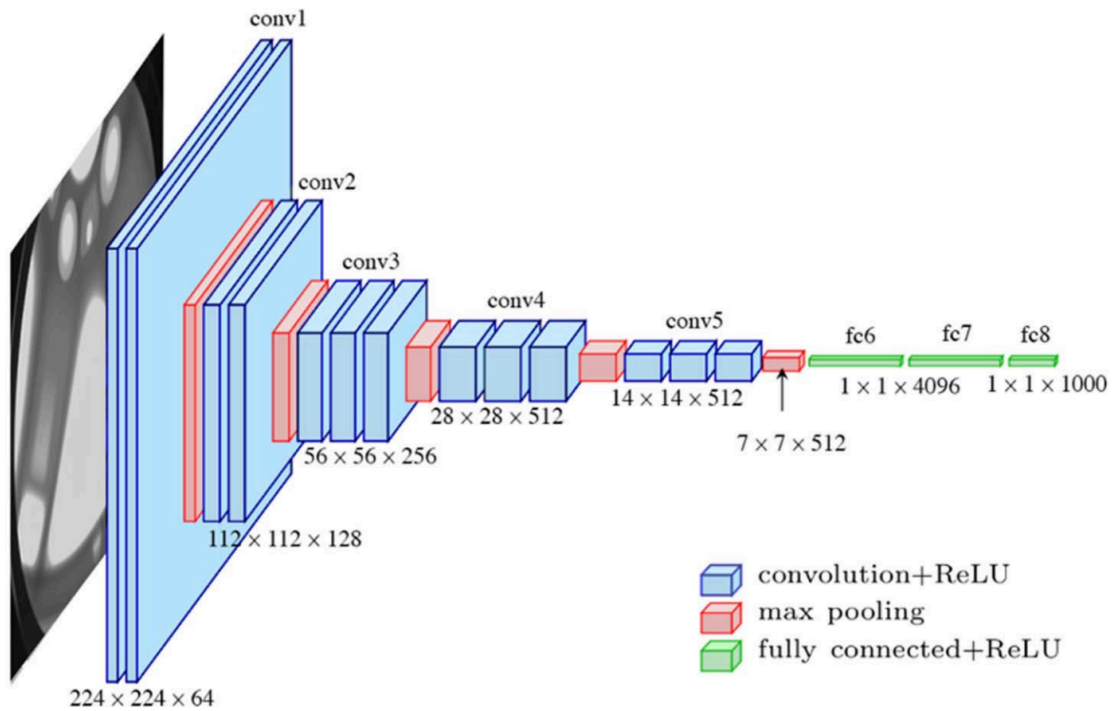


Figure 3.4: VGG16 architecture (reprinted from [3] ©2017 IEEE)

our goal was to only extract the feature descriptor instead of obtaining classification results, we discarded the fully connected layers of the network. The activations obtained from the last convolutional layer i.e the 5^{th} max-pooling layer was used as the global feature descriptors for the input image. The output of the 5^{th} max-pooling layer was 512 feature maps of size 5×5 . We flattened all the feature maps and concatenated them to obtain 1-dimensional feature vectors of size 12800. We extracted these feature vectors for all the images in our training data. Given an analysis image, we extracted a feature descriptor with 12800 elements and then compared it with all the feature descriptors obtained on the database of brain images. The analysis image is selected at random from a database of test images for which feature descriptors haven't been generated previously.

Treating the task of finding the matching atlas corresponding to a given analysis image as an image retrieval problem, we applied the k -nearest neighbor (k -NN) algorithm to find the matching reference image for the query image. The k -NN algorithm used the SSD error metric to find the

closest feature vectors in the database to the feature vector of the analysis image. Empirically, we found that $k = 3$ gave decent results. Figure 3.5 shows the results obtained using just the global features in the form of a confusion matrix. The vertical axis represents the actual AP number that the analysis image belongs to while the horizontal axis represents the AP numbers predicted by the k -NN algorithm on the global VGG16 feature descriptors. We can clearly observe that VGG-16 feature descriptors are powerful enough to differentiate images that look very different from each other. Analysis images from the in the anterior portion of the brain are generally assigned to AP numbers belonging to the anterior portion of the brain and a similar observation can be made about images obtained from the posterior portion as well. Results are poor for the images which look very similar, especially belonging to the medial portion of the brain. This shows that the VGG-16 features descriptors obtained globally on the images are not discriminative enough to tackle our problem.

To address this short-coming of the global features, we created feature vectors using a pre-trained VGG-16 network from patches in the image. We called these feature vectors local deep features since they captured the local and regional details in the brain image. In the next section we discuss in detail about our intuition behind the approach, the implementation details and the results we obtained using the local deep features.

3.4.2 Local Deep Features

The idea behind using deep features was very straight forward. While extracting feature points and matching them, we made an observation that the given analysis image had particularly more number of feature matches with reference images that closely resembled the analysis image. This might seem obvious because these reference images would have interesting regions similar to those on the analysis image. But, at the same time solely considering the number of feature matches to find the correct matching reference image did not turn out to be good idea. It all boiled down to how similar the analysis image looked to the reference images. In many cases there were local regions in the reference images that looked similar to the analysis image due to similar contrast and edge profiles. The feature descriptors created by the inbuilt feature extractors in MATLAB are

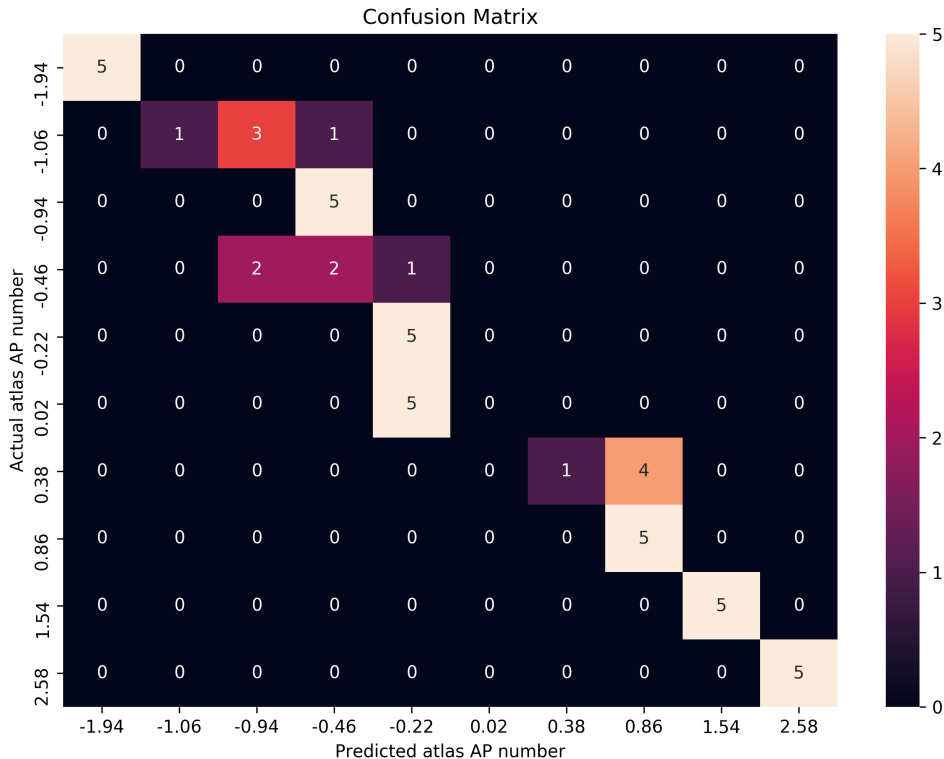


Figure 3.5: Confusion matrix for atlas retrieval using global features

not robust enough to these variations. This led to many incorrect feature matches with many of the matching points located far away from each other in the images to be registered.

So as discussed above, we observed that the feature points identified by conventional feature detection algorithms are robust but the feature vectors created for them are not. To tackle this challenge we resorted to using a CNN instead to create feature vectors. The idea behind the approach was that CNNs take into consideration both the structures and the intensities inside a region. To make the feature vectors robust, we used the pre-trained VGG-16 that we used to extract global features to create the feature vectors from around the feature match locations. We called these features deep local features. These features were obtained by creating patches around the locations of feature points. Specifically we used a patch of size 64×64 pixels at the locations where feature points were obtained. An illustration of the same is shown in figure 3.6. These

patches were fed as input to the VGG-16 network and feature vectors were obtained for each patch. This is exactly similar to what we did in the global feature extraction feature, the only difference being that the size of the input in that case was 224×224 .

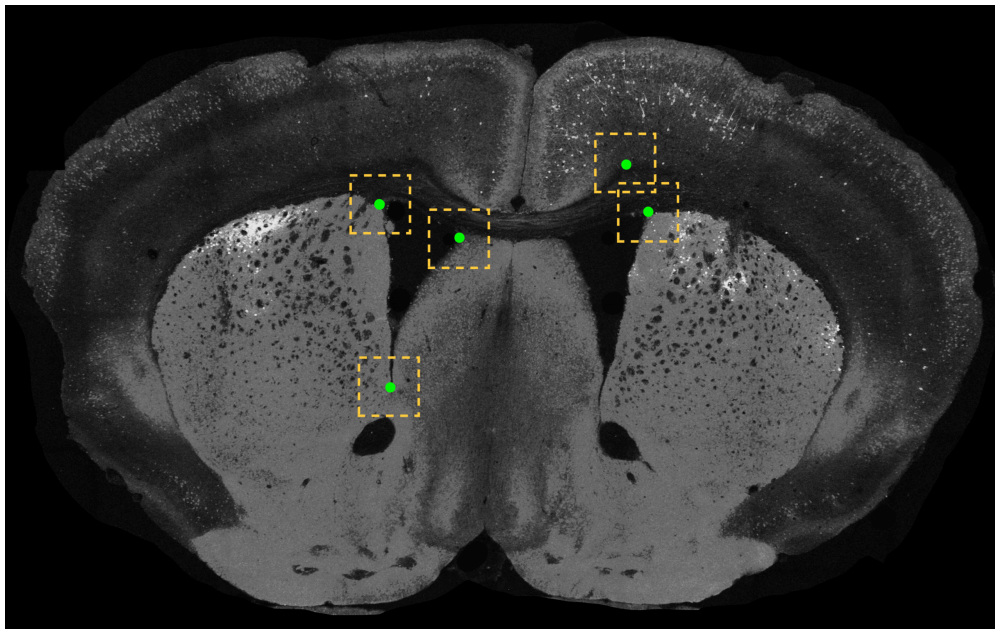


Figure 3.6: Illustration of windows (yellow dashed lines) drawn around key-points (green dots)

We employed a two-stage process in which we first created the global feature vector for a given analysis image after resizing it to 224×224 pixels and then using the k -nearest neighbors algorithm found the reference image whose feature vector resembled closest to the global deep feature vector of the analysis image. This was made possible by creating a database of global deep feature vectors obtained from all the reference images beforehand. These feature vectors were created by resizing the reference images to 224×224 pixels and then feeding them to the pre-trained VGG-16 network. Specifically there were 50 reference images for each atlas AP which we had created from 5 original images using a few data augmentation tricks. Experimentally we observed that $k = 3$ for the k -nearest neighbors algorithm performed well. Once we obtained the closest matching reference image, we predicted the AP number for the analysis image. As explained in the

previous section, global features worked well for brain sections belonging to the anterior and the posterior regions of the brain. So, if the predicted AP number lied below -1.00 mm and above 1.00 mm, we considered the predicted AP number as our final prediction. Otherwise, if the predicted AP number was between -1.00 mm and 1.00 mm, we made use of the local deep features. We extracted feature locations using the Harris corner and SURF feature detectors. This was followed by feature matching to obtain the matching locations on both the analysis image and the reference image. Instead of using the conventional feature vectors used for feature matching, we obtained local deep features from patches around these locations. Then we appended the features obtained from all these patches to create a single feature vector for the image. This procedure was done on the analysis image and all the reference images belonging to AP numbers between -1.00 mm and 1.00 mm. We calculated the cosine similarity between the local deep feature vector obtained from the analysis image and the local deep feature vectors obtained from all the reference images. The AP number for which the average cosine similarity was the highest was the one predicted for the analysis image. This approach helped us gain better results on analysis images belonging to the medial region of the brain as can be seen in 3.8. Overall the localization approach provided an improvement over the global deep features approach. Figure 3.7 shows our two-stage approach for automatic atlas localization.

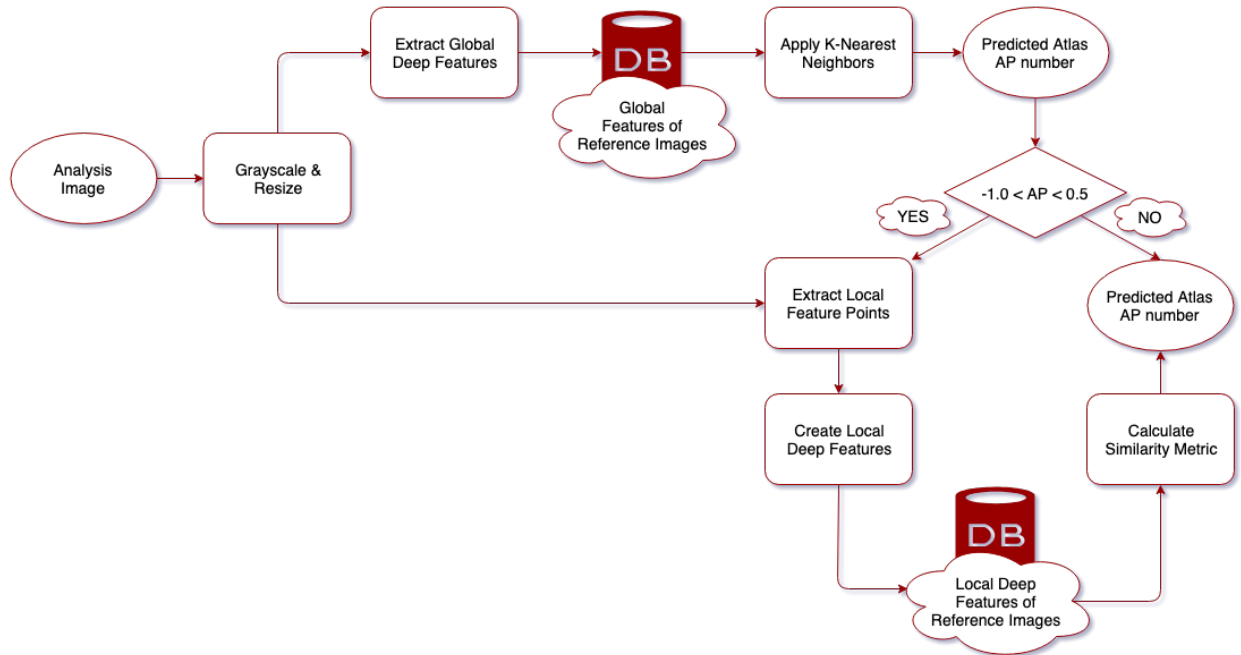


Figure 3.7: Flow chart for atlas localization

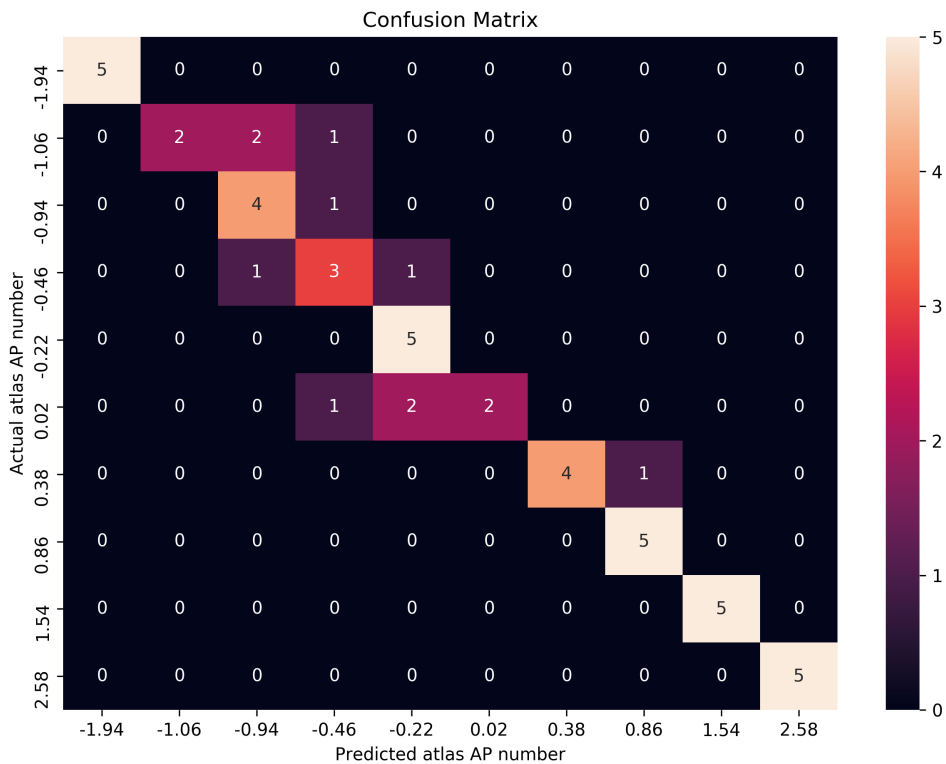


Figure 3.8: Confusion matrix for atlas retrieval using the 2 stage approach

4. BRAIN ATLAS REGISTRATION

In this section we discuss the approaches we used for the semi-automatic and automatic registration of brain atlases to the analysis images.

4.1 Geometric Image Transformation

Image transformation refers to finding similar corresponding points in two images and then matching them so that the rest of the regions in the 2 images can be aligned. The idea behind this approach is that if a relationship can be established between a few selected number of points in one image to corresponding points in another image, then the same the same relationship can be applied to the rest of the points in both the images. This would help in transforming one image into another. This relationship can be captured in the form of a transformation function. Concretely, let (x, y) be the set of selected points on the the moving image and let (u, v) be the set of corresponding selected points on the fixed image. In our case the atlas is the moving image and the query image or the analysis image is the fixed image. This choice is straightforward since we wish to transform the atlas to align it with the regions in the analysis image. So, the relationship can be given by

$$g(u, v) = h(f(x, y)) \quad (4.1)$$

where $g(\cdot)$ is the fixed image, $f(\cdot)$ is the moving image, and $h(\cdot)$ is transformation function. The simplest form of transformation that we used at first is a global transformation. The general equation for a global affine transformation in matrix form is given by

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (4.2)$$

This is the homogeneous coordinates form of the matrix that is helpful in easier representa-

tion. Empirically we observed that affine transformation wasn't a good choice in our case. This is because affine transformations map lines to lines. But our images also consisted of regional variations which was difficult to capture using only an affine transformation function. So, we experimented with the polynomial transformation function. Affine transformations are first-order polynomial transformations. Higher-order polynomials have the ability to bend lines. In particular we found that the quadratic transformations provided much improved results. The general form for polynomial equations is given by

$$\begin{aligned} u &= \sum_k \sum_l a_{kl} x^k y^l \\ v &= \sum_k \sum_l b_{kl} x^k y^l \end{aligned} \quad (4.3)$$

So, for a 2^{nd} -order polynomial i.e. a quadratic polynomial, the equations in 4.3 can be written as

$$\begin{aligned} u &= a_{20}x^2 + a_{02}y^2 + a_{11}xy + a_{10}x + a_{01}y + a_{00} \\ v &= b_{20}x^2 + b_{02}y^2 + b_{11}xy + b_{10}x + b_{01}y + b_{00} \end{aligned} \quad (4.4)$$

In matrix form equations in 4.4 can be written as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_{20} & a_{02} & a_{11} & a_{10} & a_{01} & a_{00} \\ b_{20} & b_{02} & b_{11} & b_{10} & b_{01} & b_{00} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \\ x \\ y \\ 1 \end{bmatrix} \quad (4.5)$$

In MATLAB, the function "fitgeotrans(.)" provides the option to use the polynomial transformation. For a quadratic polynomial the minimum number of points that need to be selected is 6. This would generate the 12 set of equations to solve the 12 unknown parameters a_{kl} and b_{kl} as shown in equations 4.4 and 4.5. In general, more number of points leads to better fit. This leads

to solving an over-determined set of equations and the values for the parameters a_{kl} and b_{kl} using least-squares approximation [53].

4.2 Image Warping

The primary objective of image registration is to apply warps to a standard reference image to align it as closely as possible to a given analysis image. After finding the transformation function, we applied warping on the moving image to align it with the fixed image. We used the "imwarp(.)" function in MATLAB for this purpose. "imwarp(.)" takes as input the image to be transformed and the geometric transformation.

4.3 Manual Control Point Placement

Our first approach for image registration involved allowing users to manually place control points on the analysis image and the corresponding atlas. The major factors that were taken into consideration while aligning the atlas with the analysis image is the outer boundaries of both the images. To achieve this, the user placed a number of control points along the outer edges of the query image and the atlas. The recommended number of corresponding control points that should be placed along the outer boundaries of both the images is 15 – 20.

In the next major step we focused on the inner details and artifacts in the analysis brain image. The idea here was to line up the ventricles, the striatums, and the anterior commissures of the analysis image with the inner outlines of the atlas. To take this into account we placed control points at locations which should ideally match after deforming the atlas. The number of control points in the interior regions of the analysis image and the atlas was subjective. In general, placing more number of control points led to better visual matching.

The manual placement of control points was achieved using MATLAB's Control Point selection tool which is included in the image processing toolbox. This tool can be accessed using the "cpselect(.)" function. The function opens up a window as shown in figure 4.1. It allows the user to place corresponding control points on the fixed image and the moving image. The coordinates of the control points selected on the fixed image and the moving image are stored in arrays. Figure

4.2 shows the tool with the marked control points on both the analysis image and the atlas. These corresponding control points are then used to obtain the transformation function using the "fitgeotrans()" function in MATLAB. "Fitgeotrans()" takes as input the arrays of control points selected on the analysis image and the atlas. The type of transformation to be used for calculating the transformation matrix can be specified by the user. In our case the degree 2 polynomial transformation function provided good results.

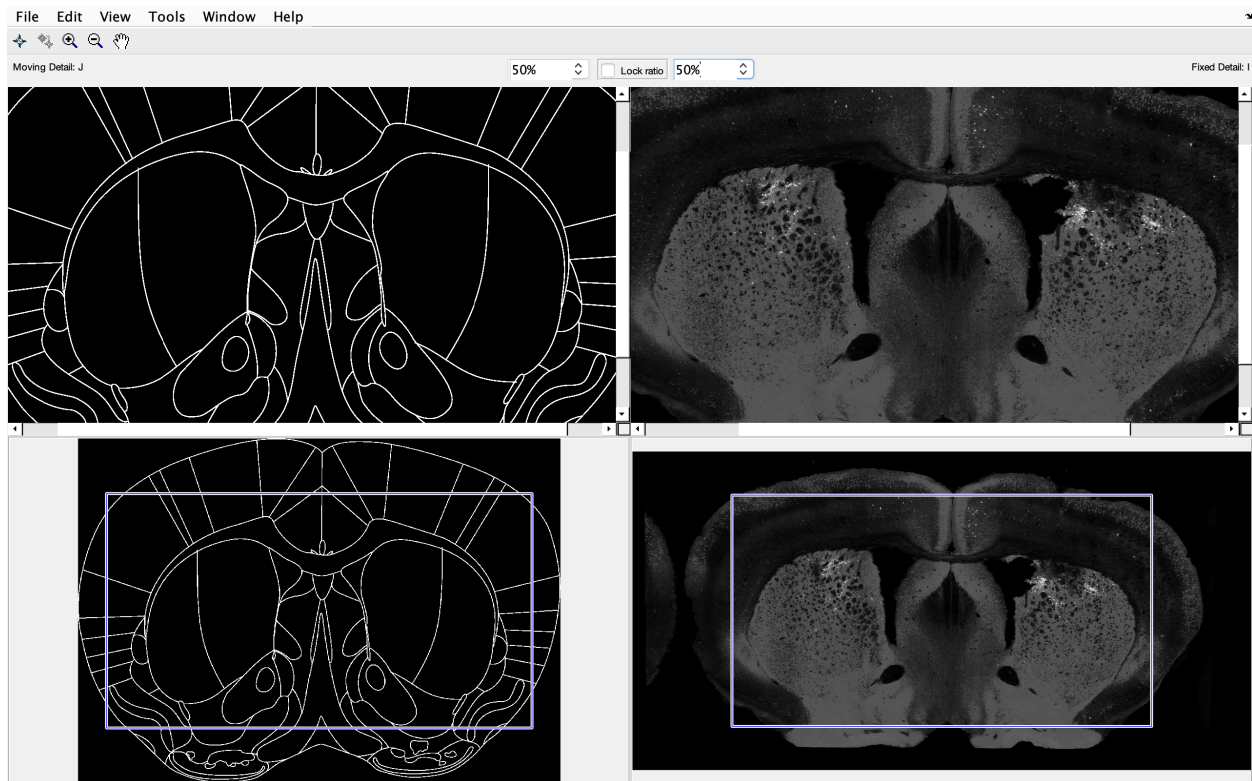


Figure 4.1: Control point selection tool

Figure 4.3 shows the results on few analysis images using the semi-automatic approach.

4.4 Automatic Control Point Placement

This section describes in detail the strategy we used to place control points on the analysis image and the corresponding reference image automatically. Brain images corresponding to each AP

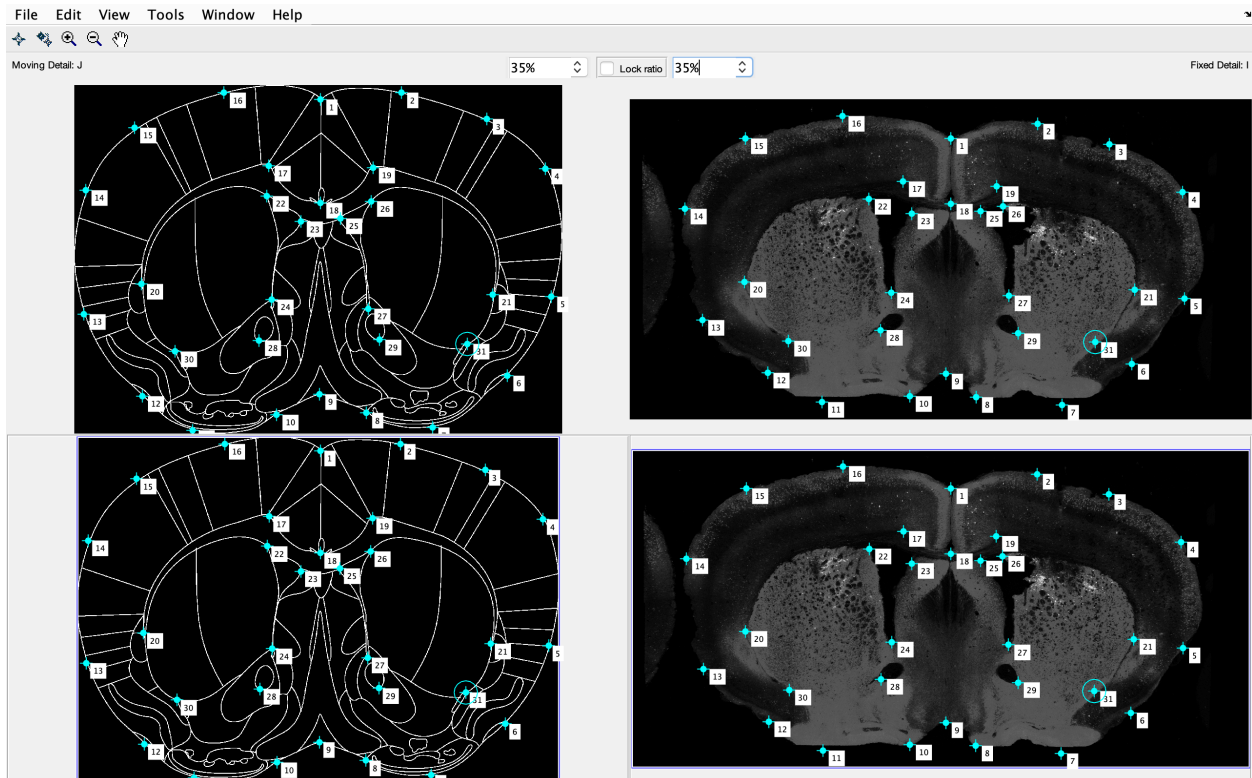
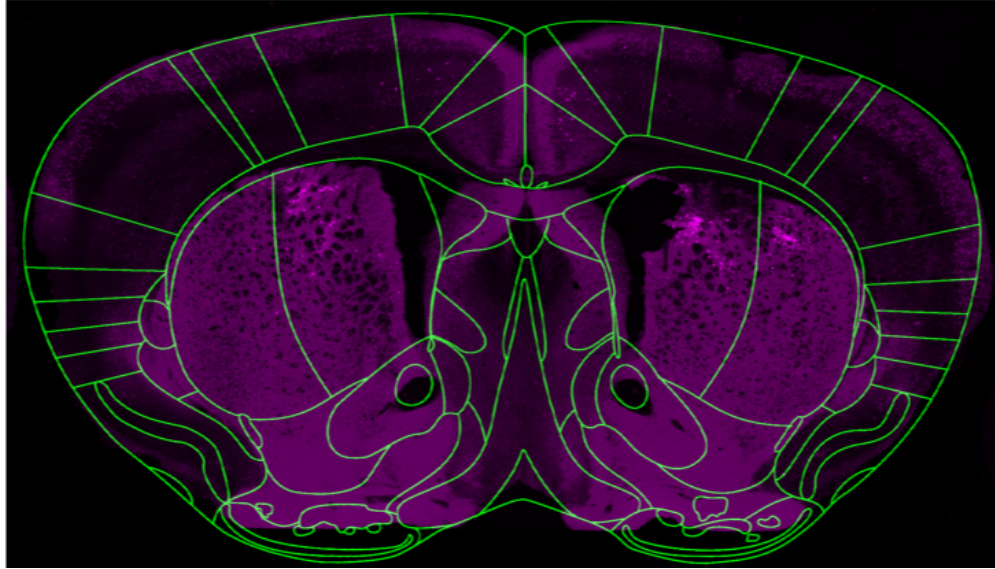


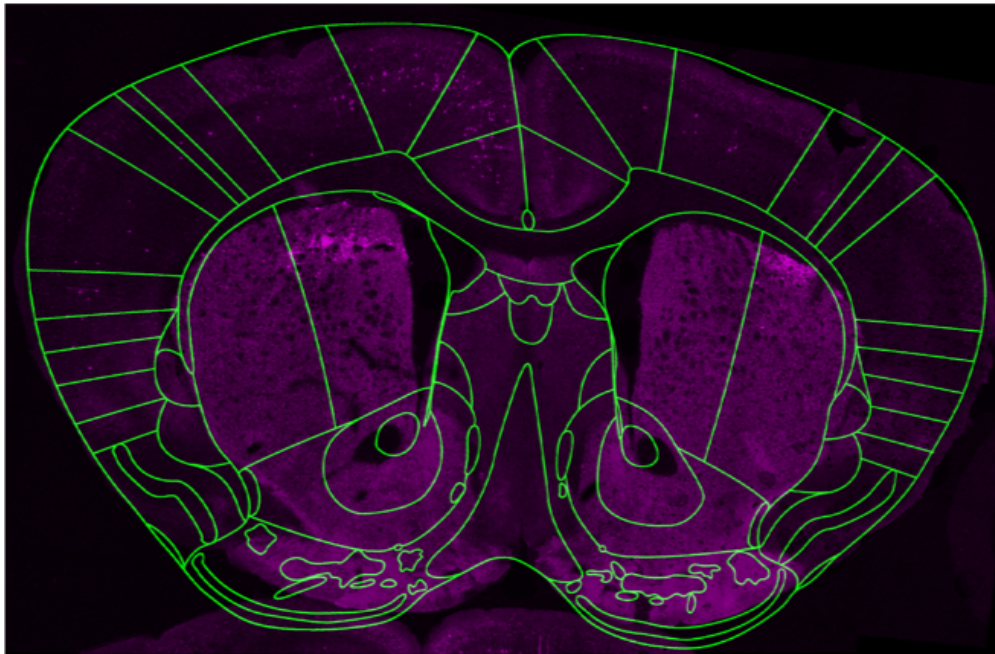
Figure 4.2: Control points placed using selection tool

number that have been cut with near perfection were added to a the database of reference images. Each reference image was then warped to fit its corresponding atlas as perfectly as possible. This was achieved by either manually warping the atlas using a photo editing software such as Adobe Photoshop or using the manual control point placement process explained in the previous section. This was necessary to ensure that the reference images and their corresponding atlases could be used interchangeably.

In the first step of this method, we placed control points uniformly around the boundaries of the analysis image and its corresponding reference image. This is owing to the reasoning explained in section 4.3 that the outer boundaries of the analysis image and the reference image should be aligned first. Extracting the outer boundaries from the analysis image and the reference images is relatively straightforward. We converted the images to black and white images to separate the brain image from the background. This was followed by identifying the structure with the largest



Registration result for AP 0.74 mm



Registration result for AP 0.98 mm

Figure 4.3: Results of manual control points-based registration

contour area in the black and white image. The contour with the largest area corresponded to the outer boundary of the brain image. We then extracted the boundaries of these contours after filling all the holes inside the brain image. All these steps were implemented in MATLAB as the image processing toolbox provides a suite of methods which facilitates these functionalities. These

steps were performed on both the analysis image and the reference image. The process of outer boundary extraction provided us the locations of all the pixels in the analysis and reference images which corresponded to the outer boundaries of the images. We used an algorithm to place control points along the outer boundary. We placed 16 control points along the outer boundaries of the analysis image and the reference image. In this method we first placed 4 control points at the top, bottom, left, and right extremes of the query image and the reference image. In the subsequent step, control points are placed mid-way between the control points placed in the previous step. This is achieved by first calculating the mid-point of the line segment joining 2 adjacent control points. Then the point on the boundary closest to the calculated mid-point is added to the list of control points. This led to increasing the number of control points on the boundaries to 8. In a similar manner we generated another 8 more points to place 16 corresponding control points along the boundaries. This method can be extended to place more control points on the boundaries, although experimentally we found that placing more than 16 control points did not improve the registration results significantly.

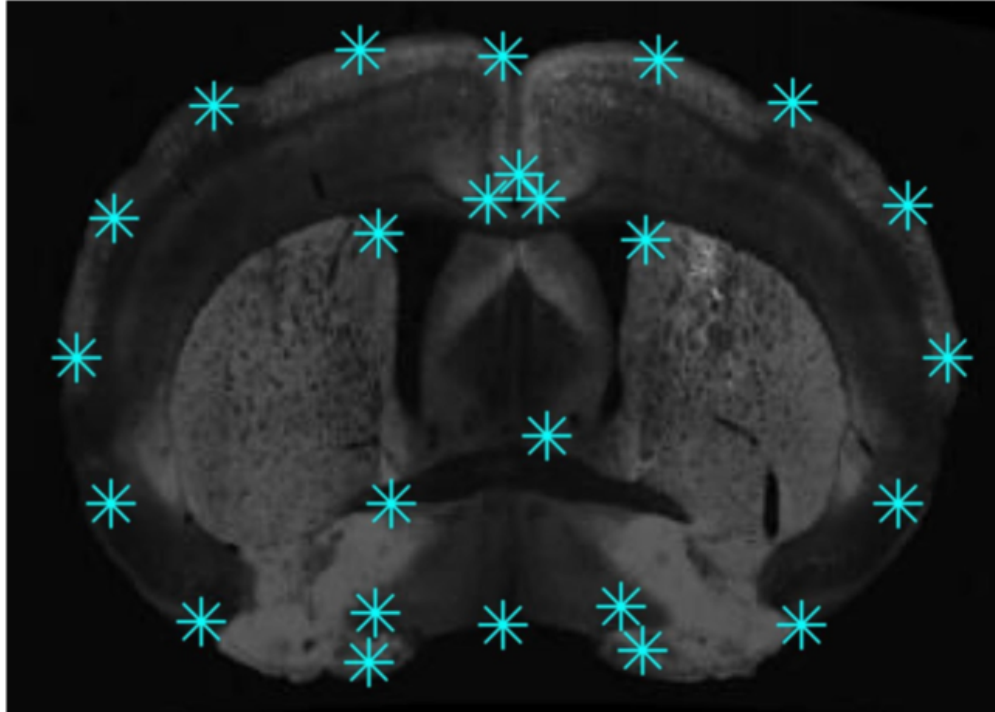
4.4.1 Harris and SURF Features-based Registration

In the next step, we ran feature detection on the analysis and reference images using the Harris corners and SURF feature detectors. We then performed feature matching to get feature point matches in both the images. This step is exactly similar to obtaining feature points for creating the local deep features as explained in section 3.4.2. These points were then added to the list of control points used for registration. Approximately a total of at least 20 control points were obtained after the feature matching step. Figure 4.4 shows the automatically placed control points on the analysis and the reference images. Figure 4.5 shows our overall approach used for automatic registration.

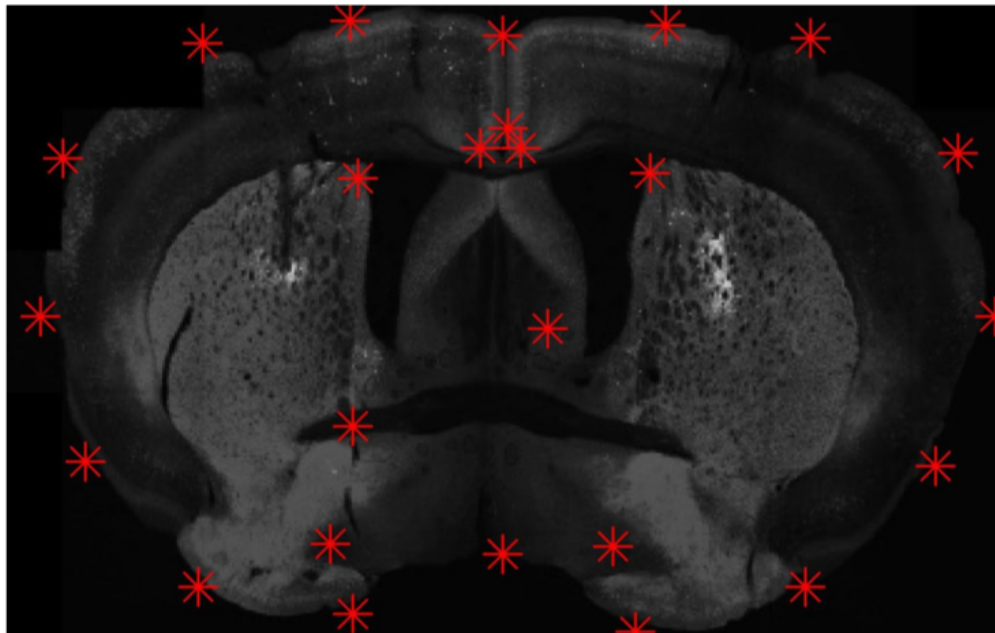
Figure 4.6 shows the results of automatic registration for a few analysis images.

4.4.2 Local Deep Features-based Registration

We also experimented with the approach of using local deep features obtained from the locations returned by the feature detectors on both the analysis and the reference images. For this step



Reference Image



Analysis Image

Figure 4.4: Control points generated automatically

we extracted the 50 feature points having the strongest Harris corners response and the 50 feature points having the strongest SURF features from both the analysis and the reference images. Local

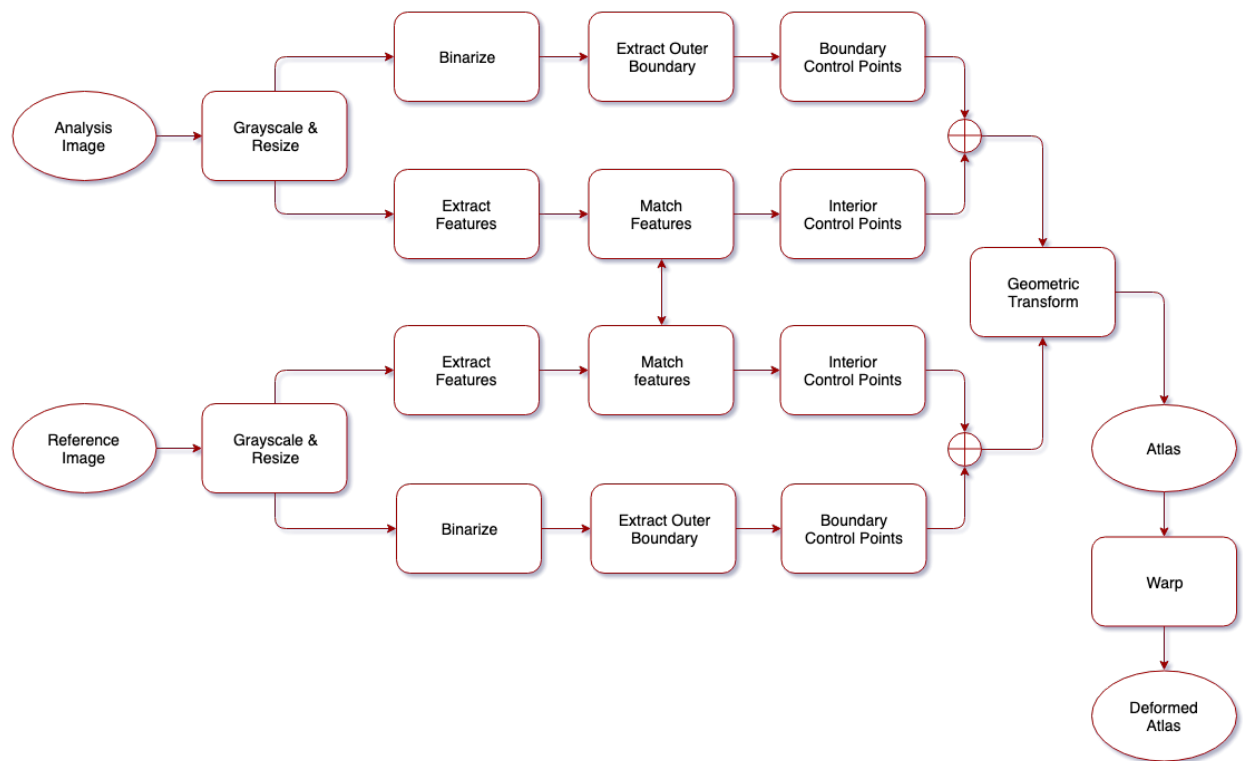
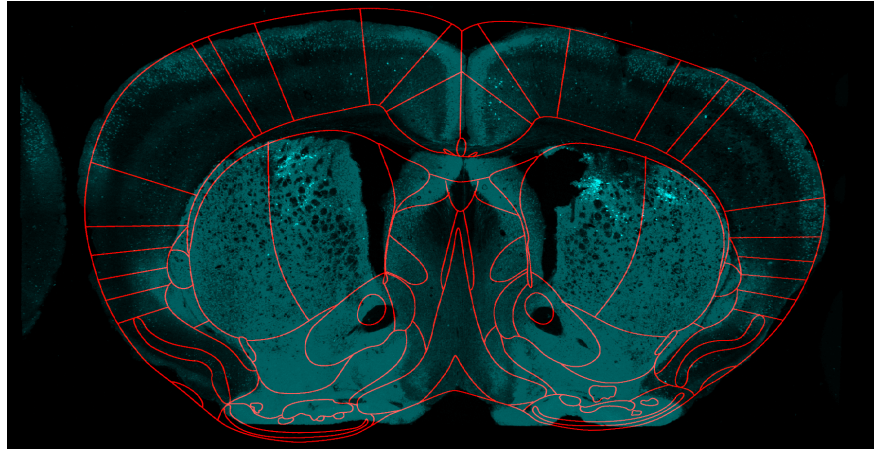
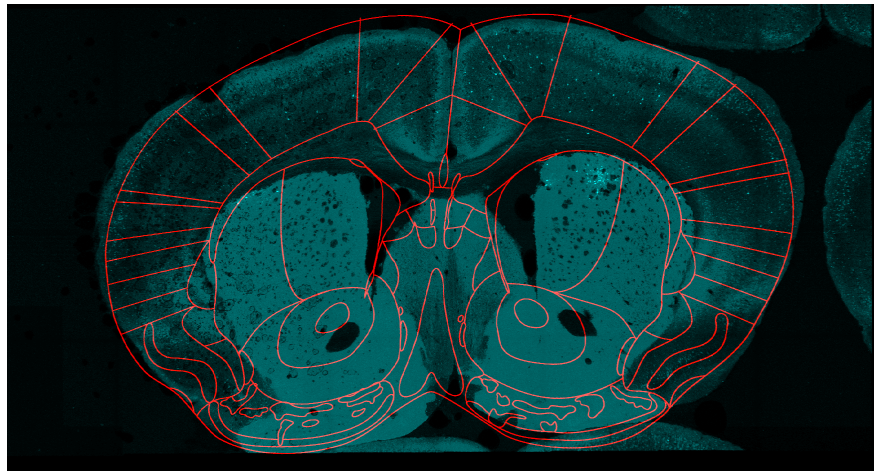


Figure 4.5: Flow chart for automatic registration

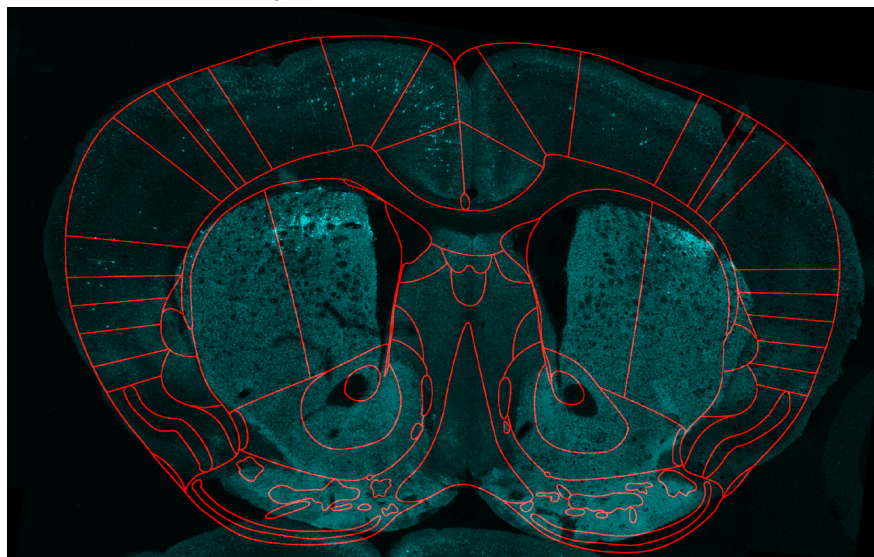
deep features were then extracted from patches of size 64×64 pixels from around these locations. These local deep feature vectors were then used for feature matching instead of the conventional feature descriptors. Our intuition behind this approach was that since convolutional features have more representational power than conventional features, they could be better for feature matching. Figure 4.7 shows the result on a sample analysis image. Although the result doesn't look good, we are still fine-tuning the local deep-feature extraction process.



Registration result for AP 0.74 mm



Registration result for AP 1.18 mm



Registration result for AP 0.98 mm

Figure 4.6: Results of automatic control points-based registration

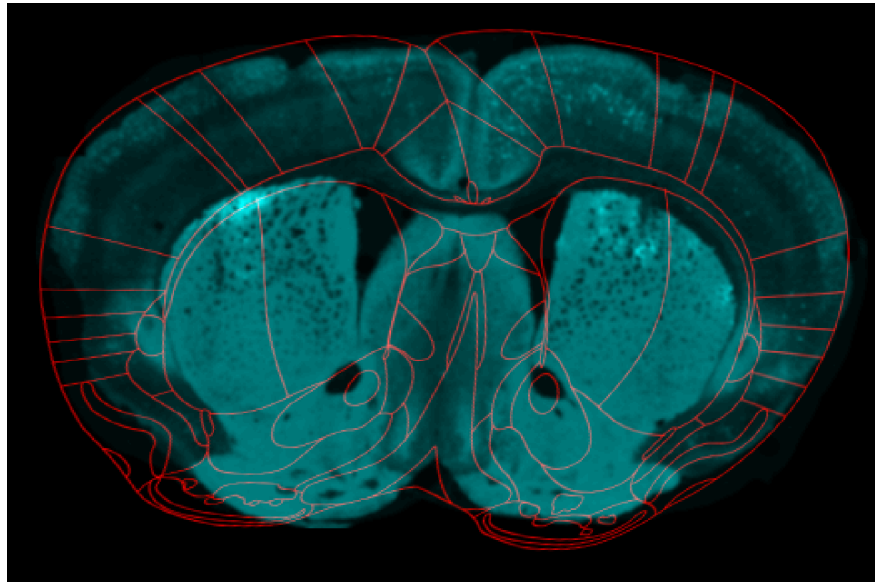


Figure 4.7: Result of local deep features-based registration

5. NEURON COUNTING

In this section we discuss the approach we used for the automatic counting of neurons in the sub-regions of the brain.

5.1 Manual Neuron Identification

Figure 5.1 shows a sample analysis image in which the expressed neurons can be seen. The bright red spots correspond to the expressed neurons.

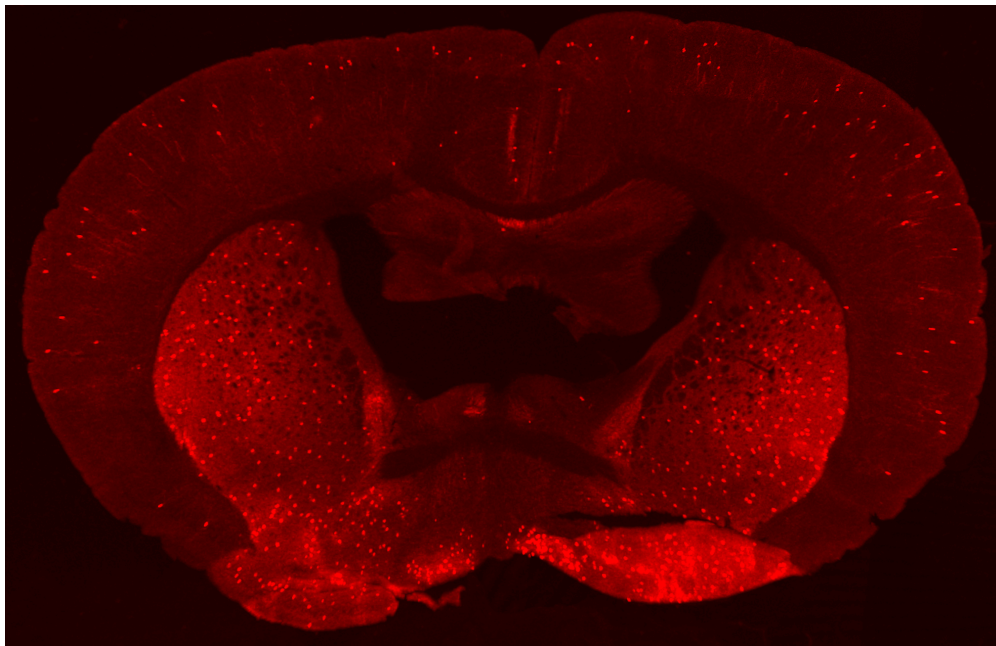


Figure 5.1: Sample analysis image with expressed neurons

Researchers in Dr. Wang's and Dr. Shapiro's labs manually place dots on the expressed neurons using a software application called Imaris. Each dot corresponds to 1 neuron. This is a time-consuming process and requires subject matter expertise. The image filled with dots is referred to as a neuron image. Figure 5.2 is an illustration of an atlas overlaid on a neuron image.

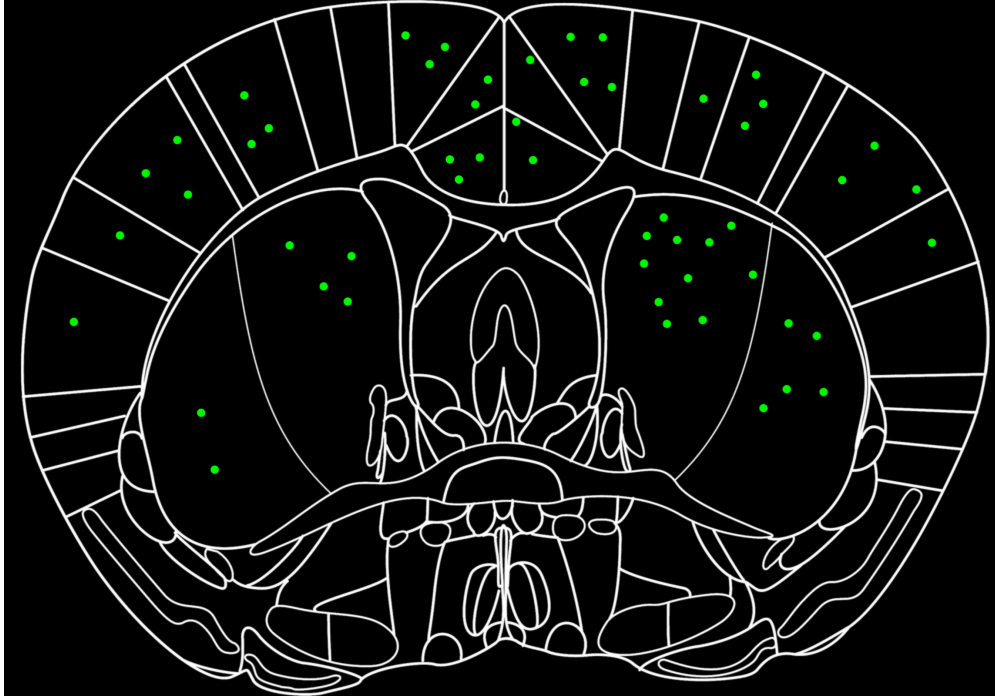


Figure 5.2: Illustration of atlas overlaid on a neuron image

5.2 Automatic Neuron Counting

We first extracted the boundaries of the deformed atlas. For each sub-region we used the "bwboundaries(.)" function in MATLAB to first arrange the coordinates of the boundary pixels in the correct order. Then we used the "inpolygon(.)" function in MATLAB to count the number of dots inside the sub-region.

The count of neurons in each sub-region was then saved in the form of an excel sheet. Figure 5.3 shows the overall approach used for neuron counting.

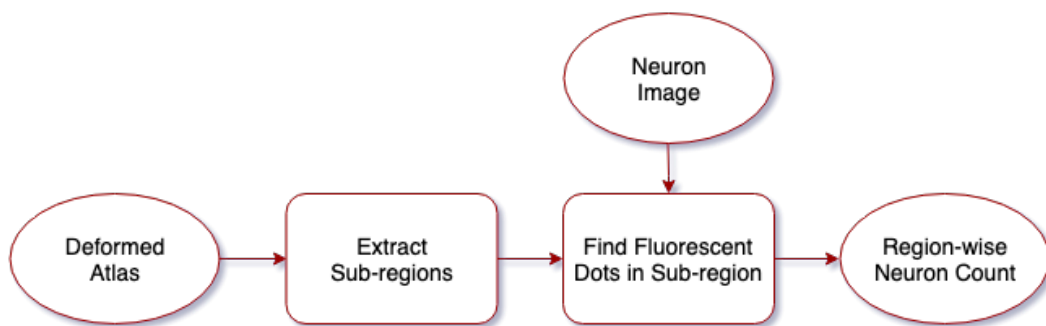


Figure 5.3: Flow chart for neuron counting process

6. EVALUATION

In this section we discuss the evaluation metrics that we used to measure the performance of our methods.

6.1 Total Percentage Counting Accuracy

One of our major objectives to check the performance of our methods was to ensure that all the neurons were actually inside the automatically deformed atlas. This was easily verified by summing the count of neurons present in each of the sub-regions in the atlas. To compute the accuracy with which our methods achieved this objective, we used the total percentage counting accuracy. This metric calculates the percentage of total number of neurons present inside the automatically deformed atlas with respect to the total number of neurons present inside the manually deformed atlas. We made a logical assumption here that all the neurons present in the neuron image lied inside the manually deformed atlas.

$$\text{Total Percentage Accuracy} = \frac{\sum_{r=1}^N \text{Auto}(r)}{\sum_{r=1}^N \text{Manual}(r)} \times 100 \quad (6.1)$$

Figure 6.1 shows the total percentage counting accuracy for 3 analysis images belonging to 3 different atlases.

6.2 Total Region-wise Error

Our next major objective was also to correctly count the number of neurons in each sub-region. To measure the performance of our methods in this regard, we compared the number of neurons obtained for each sub-region using the automatically deformed atlas and the manually deformed atlas with the neurons manually counted. To obtain the error due to the mismatch in region-wise counting, we used an error metric which we called the total region-wise counting error. This metric calculates the sum of the absolute difference in the region-wise count of neurons between the automatically deformed atlas and the manually deformed atlas.

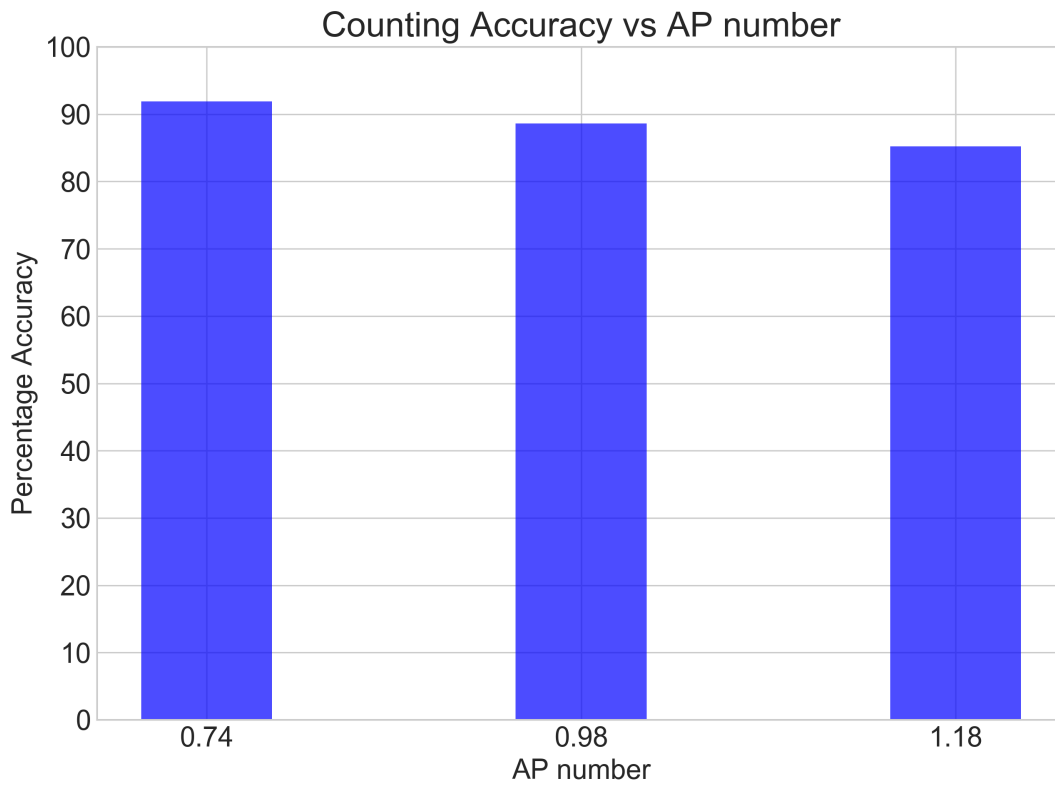


Figure 6.1: Total percentage accuracy

$$Total\ Regionwise\ Error = \sum_{r=1}^N |Auto(r) - Manual(r)| \quad (6.2)$$

Figure 6.2 shows the total region-wise counting error for 3 analysis images belonging to 3 different atlases.

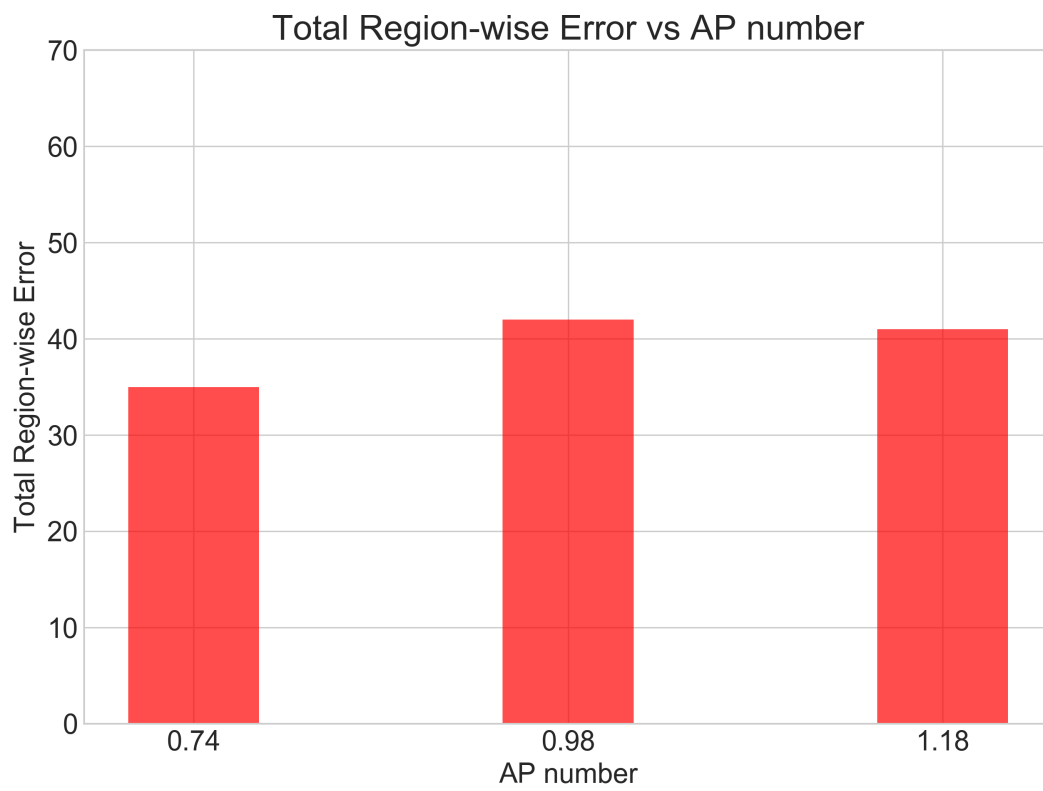


Figure 6.2: Total region-wise counting error

7. USER INTERFACE

This section contains snapshots of the graphical user interface which was one of the desired outcomes of this study. Figure 7.1 shows the panel that allows the user to specify the path to the analysis image and its corresponding neuron image. Figure 7.2 illustrates the panel in which the user can manually specify the AP number to which the analysis image belongs. Figure 7.3 contains the control point selection toolbox which allows the user to manually select control points on both the analysis image and the atlas. Figure 7.4 depicts an example of a set of control points placed both on the analysis image and the atlas. Figure 7.5 is the deformed atlas overlaid on the analysis image after warping the atlas using the transformation function obtained from the control points.

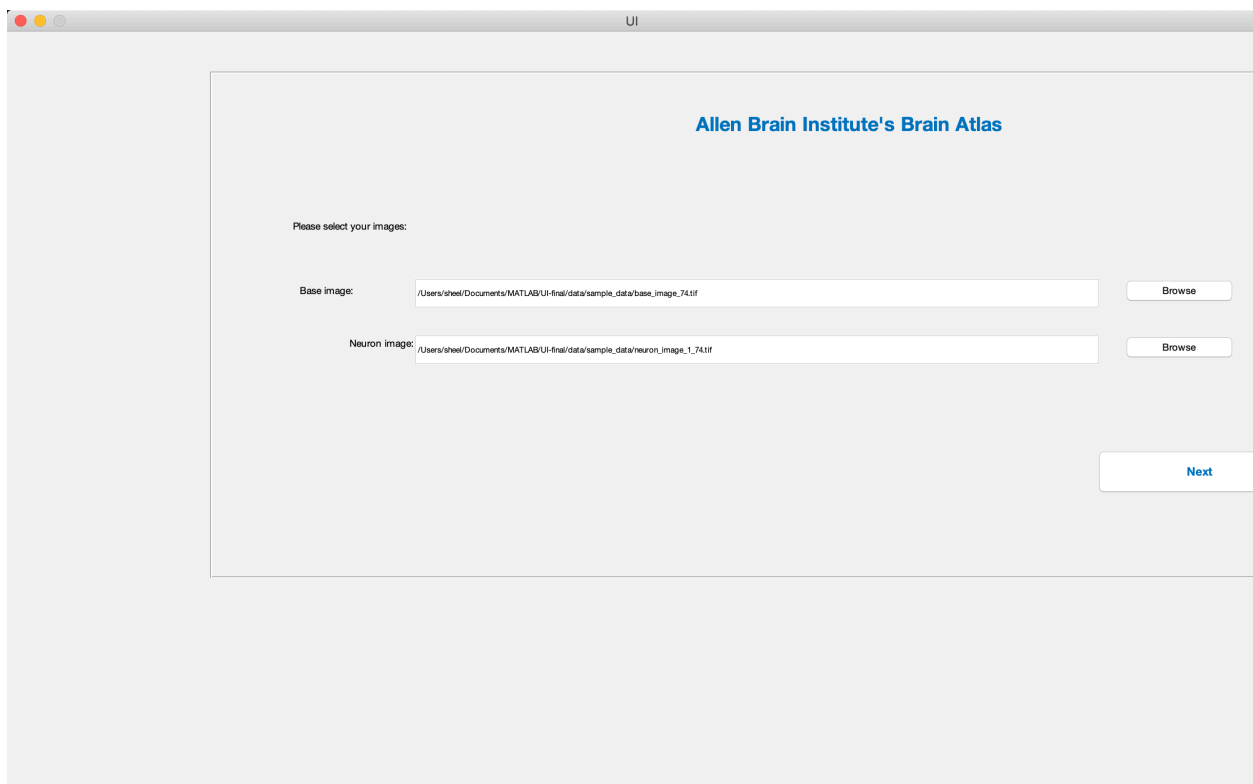


Figure 7.1: Analysis and neuron image input

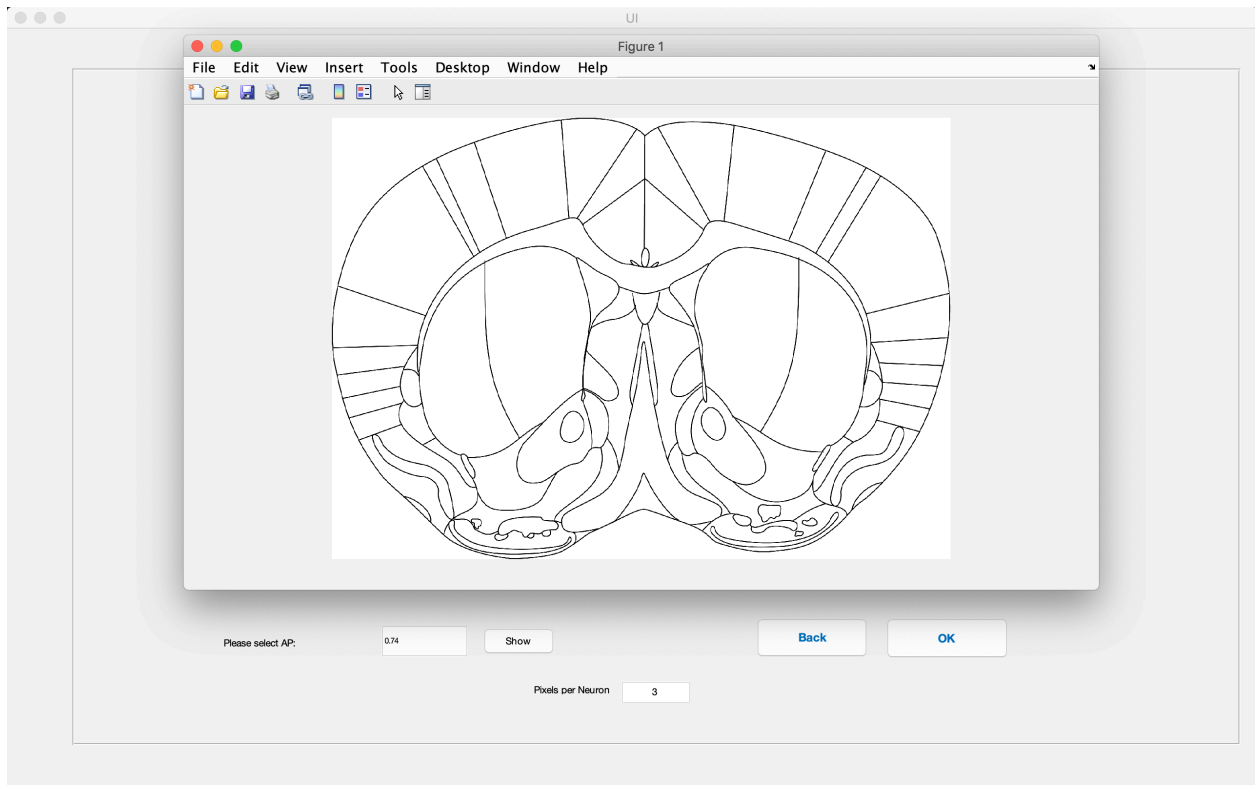


Figure 7.2: Matching atlas selection

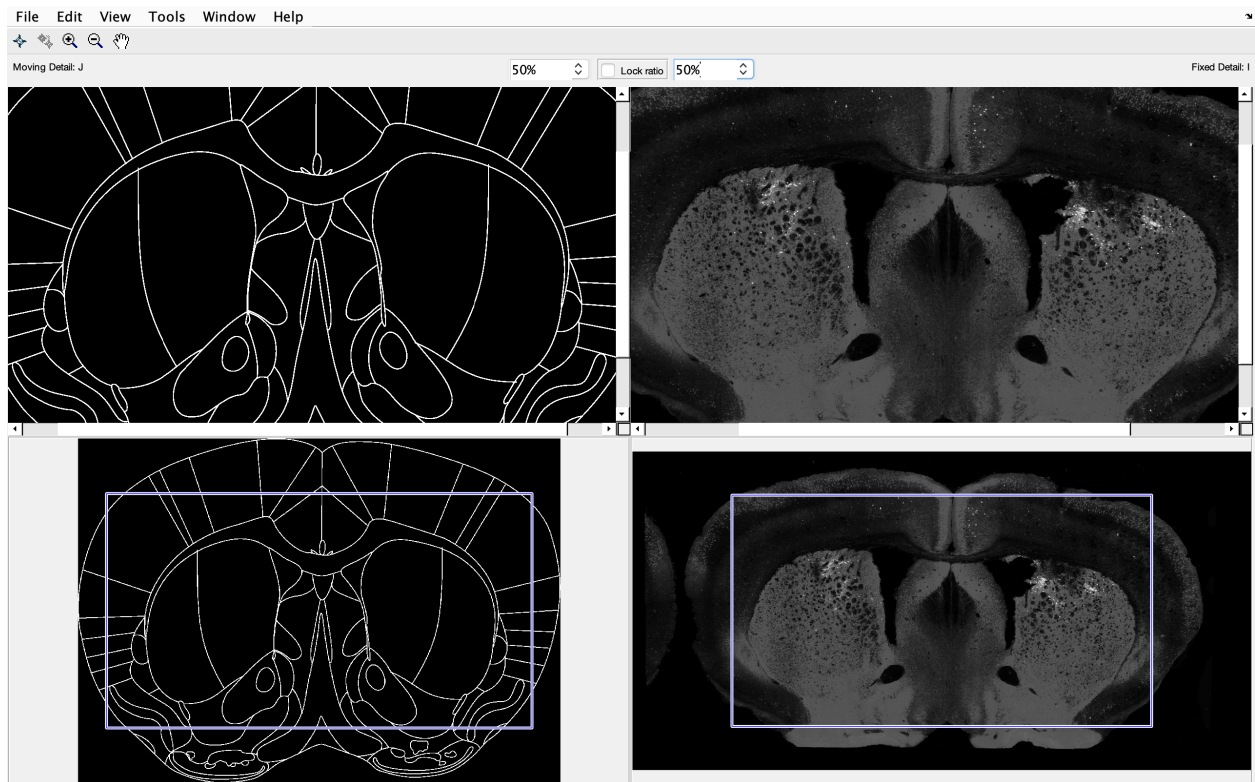


Figure 7.3: Control point selection tool

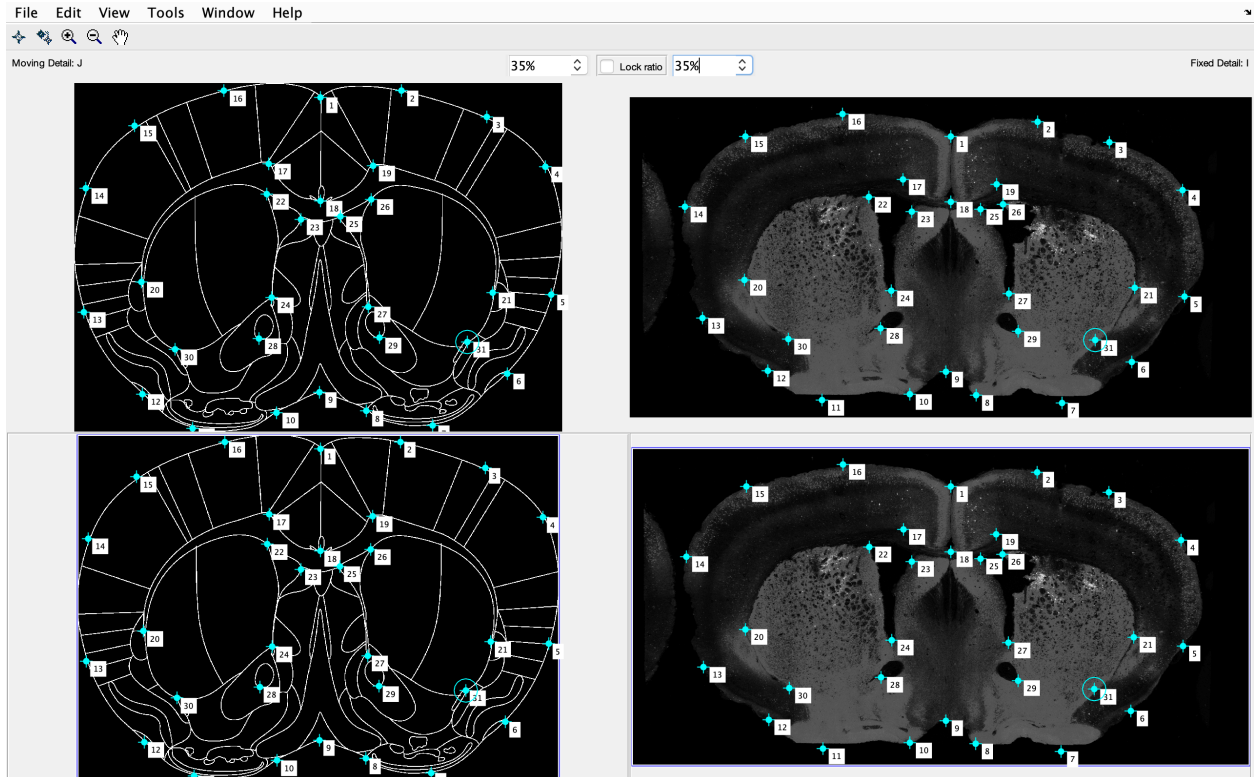


Figure 7.4: Selected control points

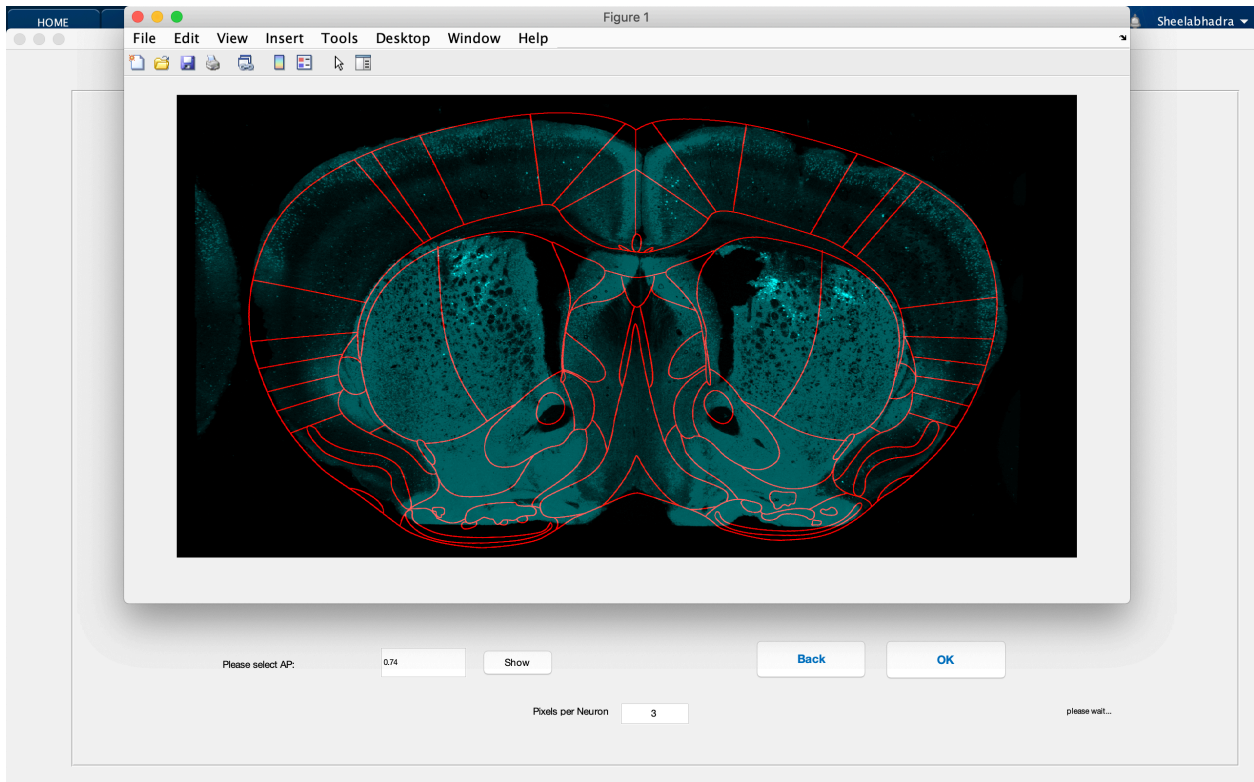


Figure 7.5: Atlas registered to analysis image

8. CONCLUSIONS AND FUTURE WORK

8.1 Findings

8.1.1 Findings having Positive Impact

The following observations made during performing the experiments helped us in obtaining better results.

1. Deep features vs conventional features

- *More representational power*: The feature vectors created after running the pre-trained VGG-16 DNN on the images resulted in features that captured their representation better. To make the feature vectors obtained from the Harris corners and SURF feature detectors robust to variations in intensity, we used the HOG feature extractor. HOG only takes the variation in intensity or the gradient into account. But from our experiments we found that feature descriptors created using the activations of the convolutional layers had more representation power. They were able to better capture the structure and the overall distribution of the pixel intensities in the images.
- *Better matches*: The better representational power of the CNN feature descriptors led to more number of matches in the local feature matching step. Instead of using the HOG-based features obtained from the Harris corners and SURF feature points, we created small patches around the locations where Harris corners and SURF feature points were obtained. The size of the patch selected around the points was 64×64 pixels. We obtained feature vectors for these patches. This resulted in higher quality of feature vectors and hence more matches. This was particularly evident in the brain atlas localization process.

2. Robust global transformation

- *Polynomial warping works well for small deformations*: After obtaining corresponding

matching points on the analysis image and the atlas, we applied a 2^{nd} -degree polynomial transformation on the control points placed on the atlas to warp the atlas to align it with the analysis image. The polynomial transformation is a step-up over the linear transformation as it allows the bending of straight lines to form curves which might exist in the final atlas. With the help of experiments we observed that although polynomial transformation is a global transformation, they performed well on images with small deformations. These transformations were able to capture the overall shape and structure present in the analysis image. This resulted in the atlas aligning well with the boundary and the internal regions of the analysis image.

3. Deep features-based localization

- *Global deep features work well for anterior and posterior sections:* We extracted global features from an entire image by running the image through the pre-trained VGG-16 network. This was done after resizing the image to 224×224 pixels, which is the input size required by the VGG-16 network. The feature maps so obtained from the last convolutional layer were used as feature descriptors for an image. In our case, we obtained 512 feature maps of size 5×5 which we flattened to obtain 1-dimensional feature vectors of size 12800. Brain sections belonging to the anterior and the posterior regions of the brain have lesser number of details than those belonging to the medial region. This makes the global feature vectors good enough for identifying brain image slices belonging to the anterior and posterior part of the brain.
- *Local deep features improve accuracy in medial sections:* For brain images belonging to the medial part of the brain, we created feature vectors from patches of the image using the pre-trained VGG-16 network. We used a patch size of 64×64 to obtain the feature vectors from the patches which we refer to as local deep features. We concatenated the feature vector obtained from all the patches to create a single 1-dimensional feature vector. This was used for the localization of brain slice images belonging to the

medial part of the brain. We observed that it led to a higher accuracy of assigning a given brain slice image belonging the medial region of the brain to its correct matching atlas. We believe that it was mainly because of the better representational power of CNNs that led to the creation of rich feature descriptors.

4. Deep features-based registration

- *Improve the quantity of Control Points:* Using the pre-trained VGG-16 network for creating feature descriptors led to more number of control point matches. The quality of matches were not good as our preliminary experiments suggested. This was evident by the registration results obtained using the deep feature based matching. A possible solution would be to use a network pre-trained on brain images to create the feature descriptors. This would reduce the number of poor feature matches and provide a more accurate location of corresponding control points.

5. Boundary control points

- *Control Points on the boundary are effective for overall registration:* We observed that placing control points along the outermost boundaries of the images to be registered played a key role in capturing the overall global deformation of the atlas. So, we placed 16 control points uniformly along the outermost boundaries of the analysis image and the atlas.

8.1.2 Findings having Negative Impact

The following factors reduced the effectiveness of our approach.

1. Interpretability of features points

- *Feature points may not be prone to deformation:* The control points obtained using feature matching are the points that the feature detectors determine to be interesting. It might so happen that these points are not prone to deformation. In these cases, the

transformation function obtained using these points did not capture the overall transformation function involved in the warping. In many cases the feature detector selected control points that wouldn't be selected by an expert and didn't select control points that would be selected by an expert. This created low interpretability of the control points selected using the feature detectors.

2. Local warping

- *Global transforms don't perform well for large deformations:* When the deformation in the brain slice image was large, the global transforms were not able to capture the deformation over the entire image. This was due to the deformations introduced during cutting the brain slice.
- *Local warping is necessary:* There were slight variations in the inner structures of the brain from one subject to another. To capture these local variations global transforms weren't sufficient. In these cases local warping became necessary.

3. Inconsistent contrast levels

- *Adversely affects feature extraction and matching:* The feature detectors used for feature extraction namely Harris corners and SURF use the intensities of the neighboring pixels to create the feature descriptors. These feature descriptors are then used for feature matching to obtain control points on the analysis image and its corresponding reference image. We observed that the feature matches were adversely affected if the contrast levels of the analysis image and its reference image were very different. Using HOG features to build the feature descriptors instead of just the intensities gave slightly better results. But overall the quality and the quantity of feature matches were adversely affected.
- *Intensity-based registration is not an option:* Due to the reason mentioned in the previous paragraph, intensity-based registration methods are not very suitable for our task.

Besides, intensity-based registration methods did not perform well on brain slice images with large deformations.

4. Treating entire brain image as a single entity

- *Registration is poor for large deformations:* For cases where there was large deformation, we did not get good registration results. A major reason for this issue was that we treated the whole brain image as a single entity. The transformation function had a tough time taking into account the control points over the entire image.
- *Possible solution is to treat right and left hemispheres separately:* A pragmatic solution to address the issue mentioned in the previous paragraph is to divide the brain image into left and right hemispheres. We can then perform registration on the left and right halves separately. We believe that this will reduce the registration error for large deformations by a large extent.

8.2 Strengths

Although, still far from perfect, our software application using the approaches mentioned in this thesis boasts of the strengths listed below.

- It is currently being tested by students in the neuroscience labs.
- It has the ability to handle brains with minor deformations.
- It works on images with varying contrast levels.
- It finds at least 25 control points automatically.
- It is fast; the whole process for 1 brain slice image takes $\sim 5 - 10$ minutes.
 - automatic localization: $\sim 2 - 4$ minutes
 - automatic registration: ~ 1 minute
 - automatic counting: ~ 3 minutes
 - semi-automatic registration: $\sim 3 - 5$ minutes

8.3 Limitations

The following aspects of our algorithms still require improvements listed below.

- The localization accuracy is still far from perfect.
- Large deformations adversely affect the registration and counting results.
- It requires some amount of data preparation such as removing spurious images from the periphery of the brain image under consideration.

8.4 Future work

8.4.1 Next Steps

The immediate next steps that could help make minor improvements in algorithm are as listed below:

- In case of large deformations it would help splitting the analysis brain image into left and right halves. Each half could then be treated separately and we could apply the techniques that we have been using on the 2 parts separately.
- We are creating a database of reference images which would in the long run be helpful in training DNNs to improve the localization results.
- To handle local deformation we could apply spline-based warping. Specifically, B-splines could be used since they are computationally less expensive than thin plate splines.

8.4.2 Long-term Research Scope

In the long term, especially after a large enough dataset has been created, the following research directions could be explored.

- A DNN could be trained for localization.
- The trained DNN could be used for creating feature descriptors.
- A separate DNN could also be trained end-to-end for registration.

REFERENCES

- [1] The Gene Expression Nervous System Atlas (GENSAT) Project, NINDS Contracts N01NS02331 & HHSN271200723701C to The Rockefeller University (New York, NY).
- [2] G. Paxinos and C. Watson, *The Rat Brain in Stereotaxic Coordinates*. Academic Press, 6 ed., 2007.
- [3] M. Ferguson, R. Ak, Y.-T. T. Lee, and K. H. Law, “Automatic localization of casting defects with convolutional neural networks,” in *2017 IEEE International Conference on Big Data (Big Data)*, pp. 1726–1735, IEEE, 2017.
- [4] A. Mojsilovic and J. Gomes, “Semantic based categorization, browsing and retrieval in medical image databases,” in *Proceedings. International Conference on Image Processing*, vol. 3, pp. III–III, IEEE, 2002.
- [5] S.-K. Chang, “Active index for content-based medical image retrieval,” *Computerized medical imaging and graphics*, vol. 20, no. 4, pp. 219–229, 1996.
- [6] R. Chbeir, Y. Amghar, and A. Flory, “Mims: A prototype for medical image retrieval,” in *Content-Based Multimedia Information Access-Volume 1*, pp. 846–861, LE CENTRE DE HAUTES ETUDES INTERNATIONALES D’INFORMATIQUE DOCUMENTAIRE, 2000.
- [7] H. Müller, A. Rosset, J.-P. Vallée, and A. Geissbühler, “Integrating content-based visual access methods into a medical case database,” *Studies in health technology and informatics*, pp. 480–485, 2003.
- [8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

- [9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.
- [10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *CoRR*, vol. abs/1409.4842, 2014.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015.
- [12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.
- [14] A. Rajkomar, S. Lingam, A. G. Taylor, M. Blum, and J. Mongan, “High-throughput classification of radiographs using deep convolutional neural networks,” *Journal of digital imaging*, vol. 30, no. 1, pp. 95–101, 2017.
- [15] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, *et al.*, “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,” *arXiv preprint arXiv:1711.05225*, 2017.
- [16] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2097–2106, 2017.
- [17] W. Shen, M. Zhou, F. Yang, C. Yang, and J. Tian, “Multi-scale convolutional neural networks for lung nodule classification,” in *International Conference on Information Processing in Medical Imaging*, pp. 588–599, Springer, 2015.
- [18] J. Ker, L. Wang, J. Rao, and T. Lim, “Deep learning applications in medical image analysis,” *Ieee Access*, vol. 6, pp. 9375–9389, 2018.

- [19] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. Hill, M. O. Leach, and D. J. Hawkes, “Nonrigid registration using free-form deformations: application to breast mr images,” *IEEE transactions on medical imaging*, vol. 18, no. 8, pp. 712–721, 1999.
- [20] J. P. Pluim, J. A. Maintz, and M. A. Viergever, “Image registration by maximization of combined mutual information and gradient information,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 452–461, Springer, 2000.
- [21] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.
- [22] N. Vujovic and D. Brzakovic, “Establishing the correspondence between control points in pairs of mammographic images,” *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1388–1399, 1997.
- [23] J. Canny, “A computational approach to edge detection,” in *Readings in computer vision*, pp. 184–203, Elsevier, 1987.
- [24] D. Marr and E. Hildreth, “Theory of edge detection,” *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
- [25] C. G. Harris, M. Stephens, *et al.*, “A combined corner and edge detector,” Citeseer, 1988.
- [26] D. G. Lowe *et al.*, “Object recognition from local scale-invariant features,” in *iccv*, vol. 99, pp. 1150–1157, 1999.
- [27] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*, pp. 404–417, Springer, 2006.
- [28] J. P. Pluim, J. A. Maintz, and M. A. Viergever, “Mutual-information-based registration of medical images: a survey,” *IEEE transactions on medical imaging*, vol. 22, no. 8, pp. 986–1004, 2003.

- [29] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE transactions on Medical Imaging*, vol. 16, no. 2, pp. 187–198, 1997.
- [30] W. M. Wells, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Medical image analysis*, vol. 1, no. 1, pp. 35–51, 1996.
- [31] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, "Automated multi-modality image registration based on information theory," in *Information processing in medical imaging*, vol. 3, pp. 263–274, 1995.
- [32] E. Guest, E. Berry, R. A. Baldock, M. Fidrich, and M. A. Smith, "Robust point correspondence applied to two-and three-dimensional image registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 165–179, 2001.
- [33] S. Abdelsayed, D. Ionescu, and D. Goodenough, "Matching and registration method for remote sensing images," in *1995 International Geoscience and Remote Sensing Symposium, IGARSS'95. Quantitative Remote Sensing for Science and Applications*, vol. 2, pp. 1029–1031, IEEE, 1995.
- [34] T. M. Lehmann, "A two-stage algorithm for model-based registration of medical images," in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170)*, vol. 1, pp. 344–351, IEEE, 1998.
- [35] W. K. Pratt, *Introduction to digital image processing*. CRC press, 2013.
- [36] J. Xiong, J. Ren, L. Luo, and M. Horowitz, "Mapping histological slice sequences to the allen mouse brain atlas without 3d reconstruction," *Frontiers in neuroinformatics*, vol. 12, p. 93, 2018.
- [37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893 vol. 1, June 2005.

- [38] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, 1910.
- [39] E. R. Denton, L. I. Sonoda, D. Rueckert, S. C. Rankin, C. Hayes, M. O. Leach, D. L. Hill, and D. J. Hawkes, "Comparison and evaluation of rigid, affine, and nonrigid registration of breast mr images," *Journal of computer assisted tomography*, vol. 23, no. 5, pp. 800–805, 1999.
- [40] A. Goshtasby, "Image registration by local approximation methods," *Image and Vision Computing*, vol. 6, no. 4, pp. 255–261, 1988.
- [41] M. Ehlers and D. N. Fogel, "High-precision geometric correction of airborne remote sensing revisited: the multiquadric interpolation," in *Image and Signal Processing for Remote Sensing*, vol. 2315, pp. 814–825, International Society for Optics and Photonics, 1994.
- [42] R. Wiemker, K. Rohr, L. Binder, R. Sprengel, and H. S. Stiehl, "Application of elastic registration to imagery from airborne scanners," *International Archives of Photogrammetry and Remote Sensing*, vol. 31, pp. 949–954, 1996.
- [43] J. Flusser, "An adaptive method for image registration," *Pattern Recognition*, vol. 25, no. 1, pp. 45–54, 1992.
- [44] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 11, no. 6, pp. 567–585, 1989.
- [45] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. Kuhn, "Landmark-based elastic registration using approximating thin-plate splines," *IEEE Transactions on medical imaging*, vol. 20, no. 6, pp. 526–534, 2001.
- [46] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers, "Diffeomorphic registration using b-splines," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 702–709, Springer, 2006.

- [47] S. Klein, M. Staring, and J. P. Pluim, “Evaluation of optimization methods for nonrigid medical image registration using mutual information and b-splines,” *IEEE transactions on image processing*, vol. 16, no. 12, pp. 2879–2890, 2007.
- [48] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, “End-to-end unsupervised deformable image registration with a convolutional neural network,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 204–212, Springer, 2017.
- [49] Y. LeCun and C. Cortes, “MNIST handwritten digit database,” 2010.
- [50] H. Li and Y. Fan, “Non-rigid image registration using fully convolutional networks with deep self-supervision,” *arXiv preprint arXiv:1709.00799*, 2017.
- [51] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, “An unsupervised learning model for deformable medical image registration,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9252–9260, 2018.
- [52] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.
- [53] T. Strutz, *Data fitting and uncertainty: A practical introduction to weighted least squares and beyond*. Vieweg and Teubner, 2010.