CONVOLUTIONAL NEURAL NETWORKS FOR AUTOMATED CELL DETECTION IN MAGNETIC RESONANCE IMAGING DATA

By

Muhammad Jamal Afridi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science — Doctor of Philosophy

2017

ABSTRACT

CONVOLUTIONAL NEURAL NETWORKS FOR AUTOMATED CELL DETECTION IN MAGNETIC RESONANCE IMAGING DATA

By

Muhammad Jamal Afridi

Cell-based therapy (CBT) is emerging as a promising solution for a large number of serious health issues such as brain injuries and cancer. Recent advances in CBT, has heightened interest in the non-invasive monitoring of transplanted cells in in vivo MRI (Magnetic Resonance Imaging) data. These cells appear as dark spots in MRI scans. However, to date, these spots are manually labeled by experts, which is an extremely tedious and a time consuming process. This limits the ability to conduct large scale spot analysis that is necessary for the long term success of CBT. To address this gap, we develop methods to automate the spot detection task. In this regard we (a) assemble an annotated MRI database for spot detection in MRI; (b) present a superpixel based strategy to extract regions of interest from MRI; (c) design a convolutional neural network (CNN) architecture for automatically characterizing and classifying spots in MRI; (d) propose a transfer learning approach to circumvent the issue of limited training data, and (e) propose a new CNN framework that exploits labeling behavior of the expert in the learning process. Extensive experiments convey the benefits of the proposed methods.

To my parents and siblings for their support and encouragement.

ACKNOWLEDGMENTS

Working towards my PhD has been a rewarding and an enriching experience. Looking back, there are many who shaped my journey.

- I have been privileged to have three co-advisors for my PhD. I am thankful to all of them for guiding me on this journey. I am especially thankful to Dr. Erik M. Shapiro for his constant support over the years. My work would not have been possible without his guidance. I have been very fortunate to have an advisor who has also been a good friend. He helped me realize the importance of intelligent automation in molecular imaging and radiology. I have learned a lot from Erik over these years. Erik always encouraged me to attend the top conferences and present my work. This helped me to interact with a number of other researchers who work on similar topics. In 2015, he sent me to present our paper in MICCAI which was held in Germany and in 2016, he introduced me to many other researchers and professionals at the WMIC where I was presenting our work. Thanks Erik!

- I must also express my gratitude for my PhD advisor in the computer science and engineering department, Dr. Arun Ross. His guidance has a key role in shaping my journey. It is Dr. Ross, with whom I have been thoroughly discussing the details of all the technical content presented in this thesis. Dr. Ross has always been polite, humble, motivating, and knowledgeable. I have learned a lot from our regular weekly meetings, his feedback on papers and presentations and even from his approach towards dealing non-technical matters. Dr. Ross has also been highly encouraging on sending me to technical conferences. In 2015, he sent me to attend ICCV in Chile. I enjoyed the trip and met a number of fellow researchers. Thank you Dr. Ross!

- I am also very thankful to Dr. Xiaoming Liu for his guidance as an advisor, especially, during the initial years of my PhD. I have learned a lot from Dr. Liu and his guidance and support has been of great value to me.

- Dr. Hayder Radha is also on my PhD committee and I would thank him for all the academic discussions I have had with him. To me, he is also an excellent teacher.

- I am grateful to all the lab members including Dorela Shuboni, Shatadru Chakrvarty, Barbara Blanco, Christiane Mallett, Laura Szkolar, Steven Hoffmann, Thomas Swearingin, Denton Bobeldyk, Eric Ding, Amin Jourabloo, Yousef Atoum, Joseph Roth and Xi Yin.

- I am also thankful to Margaret Benniwitz for remotely working with us on our journal paper.

- Further, I am thankful to the Computer Science and Enginering department at MSU for providing me with the TA and RA opportunities. Thanks to the department of Radiology at MSU and to the NIH for supporting this research!

- I will also like to thank Muhammad Shahzad and Zubair Shafiq for a number of academic discussions I had with them over these years.

- I will like to thank my family and friends for their support and encouragement along the way.

- I must also appreciate the role of OISS at MSU. Thank you for all your help and for making MSU a great place to study and work at!

- I am also thankful to Katherine Trinklein, Courtney Kosloski, Linda Moore, Cathy Davison, and Debbie Kruch for their administrative assistance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Background

Cell-based therapies are poised to make a significant impact across a broad spectrum of medical scenarios. In regenerative medicine, stem cell transplants are in various stages of clinical trials for treating or slowing a myriad of diseases, including Parkinson's disease [4, 5], rheumatoid arthritis [6, 7] and multiple sclerosis [8, 9]. Cell-based therapy in the form of cancer immunotherapy is also being tested in clinical trials [10, 11]. It is well acknowledged that imaging the *location* of transplanted cells, both immediately and serially after delivery, will be a crucial component for monitoring the success of the treatment. Two important applications for imaging transplanted cells are:

1. to non-invasively *quantify the number of cells* that were delivered or that homed to a particular location, and

2. to serially determine if there are cells that are leaving desirable or intended locations and entering undesirable locations.

For multiple reasons, including image resolution, lack of radiation, and established safety and imaging versatility, magnetic resonance imaging or MRI has emerged as the most popular and perhaps most promising modality for tracking cells *in vivo* following transplant or delivery. In general, MRI-based detection of cells is accomplished by first labeling cells with superparamagnetic iron

oxide nano- or microparticles, though some cell types can be labeled directly *in vivo*, such as neural progenitor cells . Following transplant, these labeled cells are then detected in an MRI by using imaging sequences where the signal intensity is sensitive to the local magnetic field inhomogeneity caused by the iron oxide particles. This results in dark contrast in the MRI [12, 13]. In the case of a transplant of large numbers of magnetically labeled cells, large areas of dark contrast are formed. In the case of isolated cells, given sufficient magnetic labeling and high image resolution, *in vivo* single cell detection is possible, indicated by a well-defined and well characterized dark spot in the image (See Fig. 1).

Due to the rather complex relationship between iron content, particle distribution, iron crystal integrity, distribution of magnetic label and cells etc., it is difficult to quantify cell numbers in an MRI-based cell tracking experiment. This is especially the case for a single graft with a large number of cells. There are efficient methods of quantifying iron content, most notably using SWIFT based imaging [14], but the direct correlation to cell number is not straightforward, due to the reasons listed above. MRI-based detection of single cells presents a much more direct way of enumerating cells in certain cell therapy type applications, such as hepatocyte transplant [13], or for immune cells that have homed to an organ or a tumor [15]. In this case, the solution is straightforward: if dark spots in the MRI are from single cells, then counting these spots in the MRI should yield cell number. While seemingly straightforward, performing such quantitative analysis on three-dimensional data sets is a difficult task that cannot be accomplished using traditional manual methodologies. Manual analysis and enumeration of cells in MRI is tedious, laborious, and also limited in capturing patterns of cell behavior. In this respect, a manual approach cannot be adopted to analyze *large scale* datasets comprising dozens of research subjects. Various commercial software that are currently available for MRI can only assist a medical expert in conducting manual analysis. The problem is further compounded in the case of eventual MRI detection of single cells

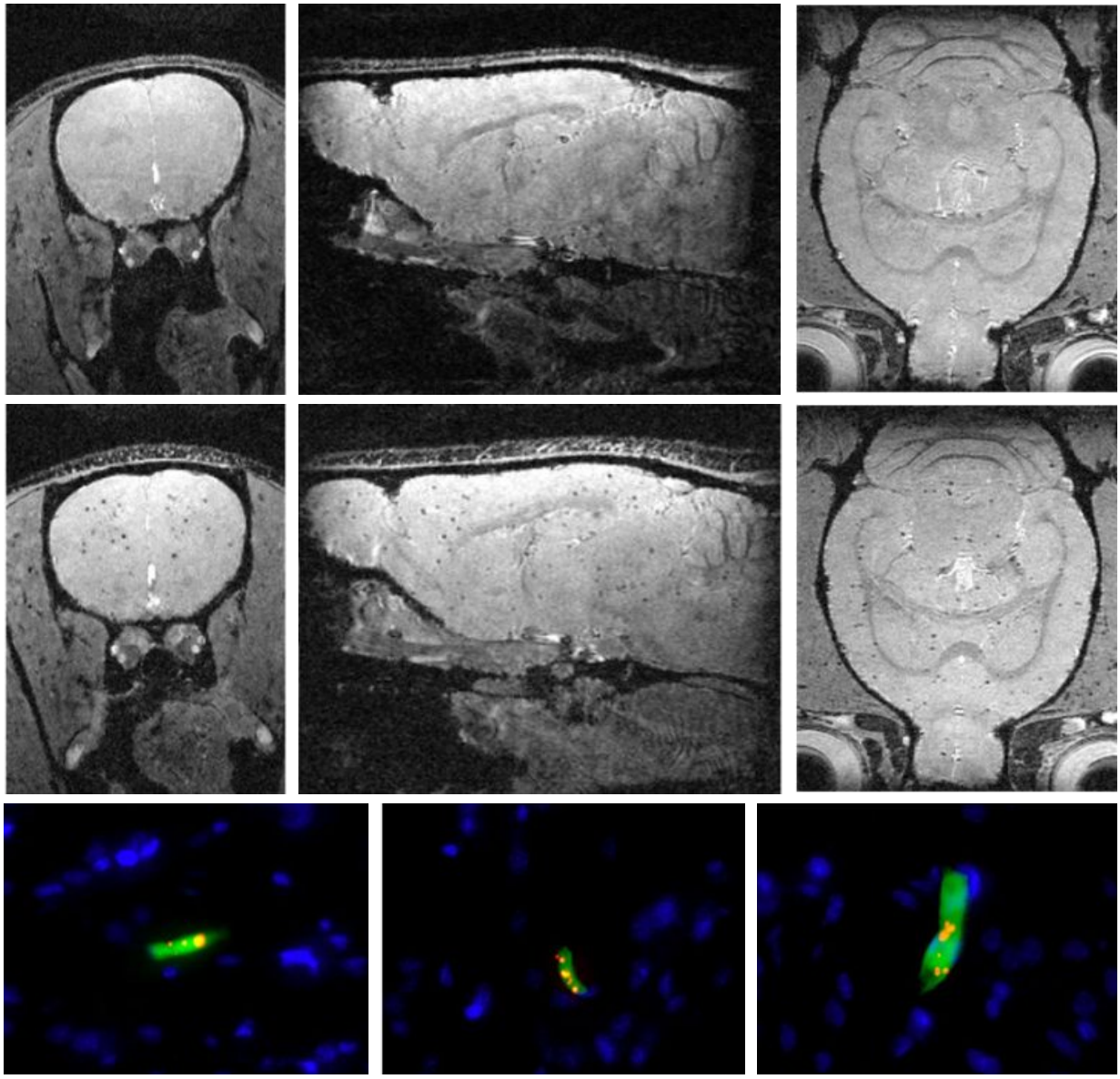

Figure 1.1: Three orthogonal MRI slices extracted from 3D data sets of the brain from animals injected with unlabeled MSCs (top row) and magnetically labeled MSCs (middle row). Note the labeled MSCs appear as distributed dark spots in the brain only. The bottom row shows three different fluorescence histology sections from animals injected with magnetically labeled MSCs confirming that these cells were present in the brain mostly as isolated, single cells. Blue indicates cell nuclei, green is the fluorescent label in the cell, red is the fluorescent label of the magnetic particle.

at clinical resolution, which is lower than that achieved on high field small animal systems. At lower image resolution, the well-defined, well-characterized dark spot loses shape and intensity and can be difficult to manually define in a large number of MRI slices.

These hurdles highlight the pressing need to develop an *automatic* and *intelligent* approach for detecting and enumerating transplanted cells in MRI, meeting all the aforementioned challenges. An automatic and intelligent approach can allow researchers to efficiently conduct large scale analysis of transplanted cells in MRI, facilitating the exploration of new transplant paradigms and cell sources. Such generalized intelligent tools will find use across a broad spectrum of biomedical pursuits. However, the unique challenges of designing such a tool has not been addressed in any prior literature, especially in the context of detecting cells in MRI.

With microscopy imaging, automatic cell detection approaches have achieved reasonable success, especially when making use of florescence to highlight the cells in an image. Notable is the work in [16], which presents a detailed comparative study of different automatic approaches for cell detection in florescence microscopy images. Their study concluded that a machine learning (ML) based automatic cell detection approach performed more superior than other well known methods. However, Note that detecting and locating transplanted cells in a 3D MRI is a different and more challenging task than cell detection in microscopy imaging. Cells appear as very small dark spots, and unlike florescence microscopy images in [16], the MRI data also contains background tissue with many spot-like structural entities. The problem becomes more difficult with small groups of noisy pixels in MRI slices that are also dark.

To design and evaluate an intelligent and automatic approach for cell spot detection in MRI, ground truth definitions, i.e., labels, that annotate spots in MRI images, are required. In [17], authors recognized the need for automation and adopted a threshold based strategy for automatically detecting spots in MRI. However, their approach was not evaluated using a ground truth. Although such threshold reliant approaches are not known to be intelligent for handling variations and diversity in data, their study in fact highlights the need for automation [16, 18]. Automatic ML approaches have been successfully used in a wide range of image analysis applications [19, 20, 16].

However, it is unexplored how such approaches can be appropriated to the problem of MRI spot detection. Further, state-of-the-art ML approaches rely on a large volume of training data for accurate learning. Unfortunately, due to practical limitations, generating large scale annotated data is challenging in both preclinical and the clinical arenas. Annotation can also be prohibitively time-consuming and can only be performed by a medical expert. Hence, crowd sourcing approaches such as the use of Amazon's Mechanical Turk [21], cannot be adopted for annotation in such applications. Therefore, the problem of spot detection using a *limited* amount of annotated training data, is an additional unaddressed challenge.

1.2 Challenges and contributions

1. **Dataset collection:** For thorough evaluation and training of the automated approach, an annotated MRI database needed to be developed. Therefore, a diverse database consisting of 40 MRI scans was assembled and more than $19,700$ manual labels were assigned. To the best of our knowledge, this is the first annotated database collected for automated cell detection in MRI.

2. **Candidate region extraction:** Given an MRI scan, a set of candidate regions needed to be extracted effectively. Each candidate region must represent a region in MRI that can potentially contain a spot. This study discusses how a superpixel based strategy can be designed to extract such regions.

3. **Feature design:** Spots have high intra-class variation due to their diverse appearances in terms of shape and intensity. Therefore, for machine learning approaches to work effectively, a set of robust feature descriptors needed to be extracted from the candidate regions. A new

CNN architecture was designed, specifically for this problem, to automatically extract the most useful spot features. The performance of these features was systematically compared against those extracted by utilizing hand-crafted feature extraction techniques. Results show that automatically learned features performed better with an accuracy of up to 97.3% *in vivo*.

4. **Learning with limited data:** Machine learning approaches typically require a large training dataset for accurate learning. However, in applications in the medical domain, it can be challenging to obtain a large volume of training data. Therefore, this thesis explored how automatic spot detection can be performed using a limited amount of training data. A novel transfer learning strategy for CNNs was developed, where the best source CNN is automatically selected from an ensemble of many source CNNs.

5. **Exploiting labeling behavior:** Labeling data in medical applications is usually more expensive and requires a medical expert. Therefore, can the labeling process in medical applications be better exploited by the classifier? More specifically, in addition to the labels on spots, can the *labeling behavior* of a medical expert be incorporated in a supervised learning framework? In this context, a new CNN framework is proposed that addresses the technical challenges associated with this research and exploits labeling behavior in CNN learning.

Chapter 2

Developing MRI Database

2.1 Introduction

Developing a labeled cellular MRI database requires several steps and involves experts with different specialities. Therefore, the goal of this chapter is to provide a general reader, an overview of our data collection process. Specific details of the collected data will also be explained. The overall process can be divided into four main steps: (1) *Cell preparation* where the goal is to grow a cell culture with magnetic particles injected in them. (2) *Animal preparation* where the prepared cells are injected into the animal under the study. This step involves the process of anaesthesia and animal incubation. (3) *MRI scanning* where the animal undergoes an MRI and the scan of the required organ is obtained. The injected cells appear as dark *spots* in MRI. In case of *in vitro* MRI, the animal may be replaced with a *tube* containing prepared cells. (4) *Label collection* where a medical expert thoroughly analyzes each slice in the MRI scan using a customized software and provides manual ground truth on the spots. The architecture of this procedure can be seen in Fig. 2.1. In the overall context of collecting a labeled database for spot detection in MRI, the contributions here can be listed as follows:

- By following all the aforementioned four steps, this study collects the first labeled MRI database that can be utilized for research in automatic spot detection. A diverse set of 40 MRI scans(both *in vivo* and *in vitro*) constitute this database.

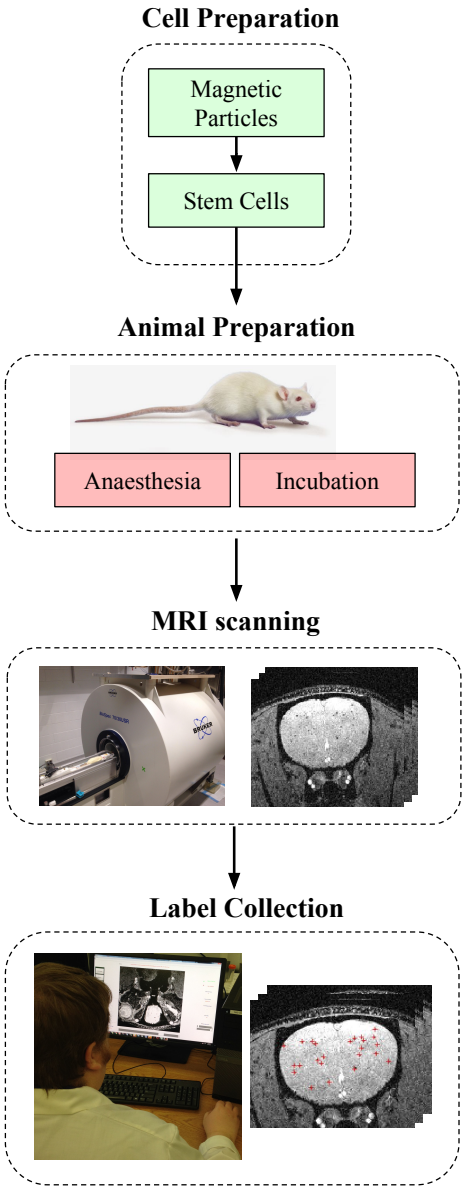

Figure 2.1: Overall architecture of the data collection process

- For label collection, a customized software was developed for experts to conveniently analyze MRI slices and provide labels. The software allows an expert to perform zoom-in and zoom-out operations; change contrast of the image; see basic statistics; and also record the *labeling behavior*.

- A total of more than $19,700$ manual labels were collected on the given MRI database.

More details of on each step of the collection approach is presented next.

2.2 Approach

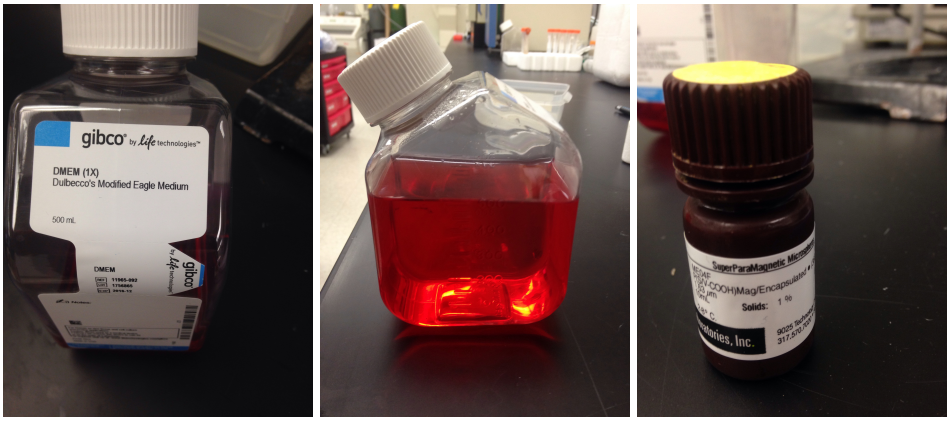

Figure 2.2: The two images on the left show the media utilized in cell culture. The media usually contains a diverse set of essential ingredients such as glucose and glutamine. The image on the right show the MPIO package utilized in our cell preparation process.

Figure 2.3: (Left) Containers with cell culture. (Middle) Temperature and air control equipment that was utilized. (Right) Temperature and Air control settings for the culture.

2.2.1 Cell preparation

The goal here is to culture stem cells such that their final form contains superparamagnetic iron oxide particles (MPIOs) inside them. In this study we utilized Mesenchymal stem cells (MSCs).

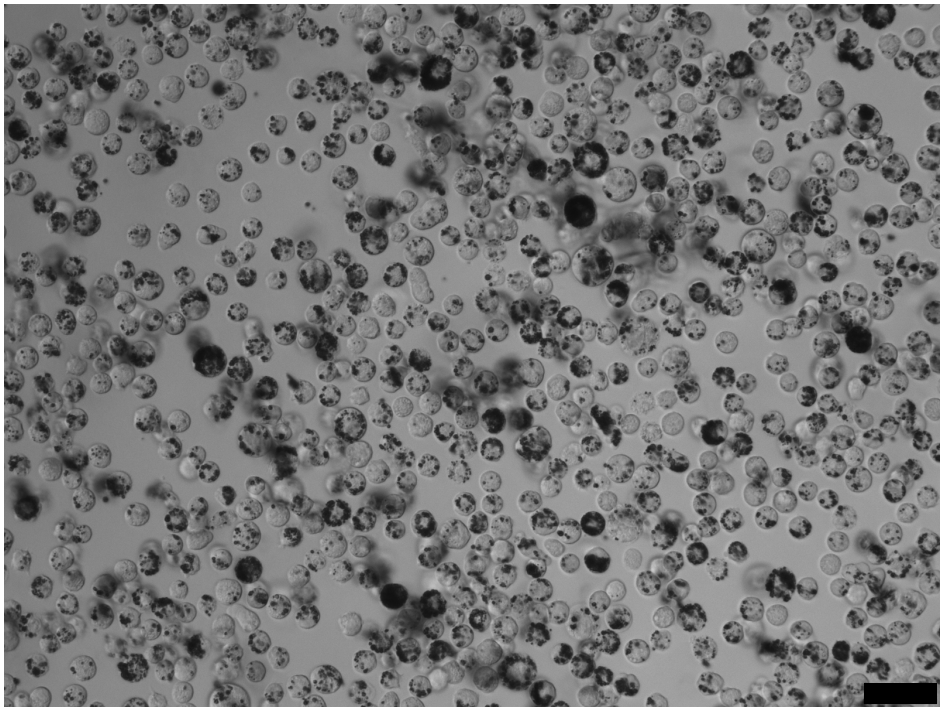

Figure 2.4: Cultured MSCs with MPIOs as seen under a microscope.

These cells were cultured using a media that was mixed with nano-sized MPIOs. In Fig. 2.2, the images of the utilized media and the MPIOs are shown. Cells feed on this media and thus, absorb MPIOs. The media generally contains a number of ingredients including glucose and glutamine. This takes place in special container that is maintained inside a temperature and air controlled equipment shown in Fig. 2.3. Once the cell absorb these particles, they can be viewed under a microscope. In Fig. 2.4, one such image of cells with MPIOs is shown. The intense dark regions represnt the particles inside the cell. Note that the large dark regions outside cells are free floating iron particles which are later cleaned using a centrifuge based procedure. Such a procedure pushes the cells low in a tube whereas the particles float on top which are then removed. This procedure may be repeated several times. More specific details related to the collected *in vitro* and *in vivo* MRI scans will be discussed later.

2.2.2 Animal preparation

2.2.2.1 Anaesthesia

The goal of this step is to prepare a subject animal for cell injection and for conducting MRI. Generally, a colony of required living animals(Rats in this case) is maintained by experts under special, university approved guidelines. For collecting each MRI scan, the animal is first given anesthesia to keep it unconscious during the rest of the procedure. Anesthesia is performed by allowing the animal to inhale Isoflurane gas. Once, the animal is unconscious, the inhalation is still monitored by an expert for a short time (about 5 min. in this case). Fig. 2.5 shows an albino rat going through the unconscious inhalation of Isoflurane which is monitored by an expert.

2.2.2.2 Cell injection

As a next step, a region on animal's body is clearly marked for cell injection. The marking also has a sterilizing purpose and is performed here using a 10% iodine solution. In Fig. 2.5, one such mark for the rat is shown. The figure shows the mark over the heart region of the of the rat indicating that the subject animal will undergo an intracardiac injection. In Fig. 2.6, an expert is shown injecting labeled cells via an intracardiac injection.

2.2.2.3 Incubation

As a next step, the animal goes through an incubation phase. The animal is first mounted on a frame by an expert as shown in Fig. 2.7 and then the incubation equipment is attached to it. The incubation equipment controls the breathing cycle of the unconscious rat. This makes sure that the rat inhales adequate oxygen and that CO_2 is properly exhaled from its lungs. Fig. 2.8 visually show this procedure for a rat.

Figure 2.5: (Top) Rat undergoes anaesthesia by inhaling Isoflurane. (Bottom) Iodine solution is utilized to mark the heart region of the rat.

2.2.3 MRI scanning

The prepared animal with MPIO labeled cells is then shifted to the MRI machine frame which is later inserted into the machine. This procedure is shown in Fig. 2.9. Usually, before performing

Figure 2.6: MSCs with MPIOs are injected into the rat (intracardiac injection).

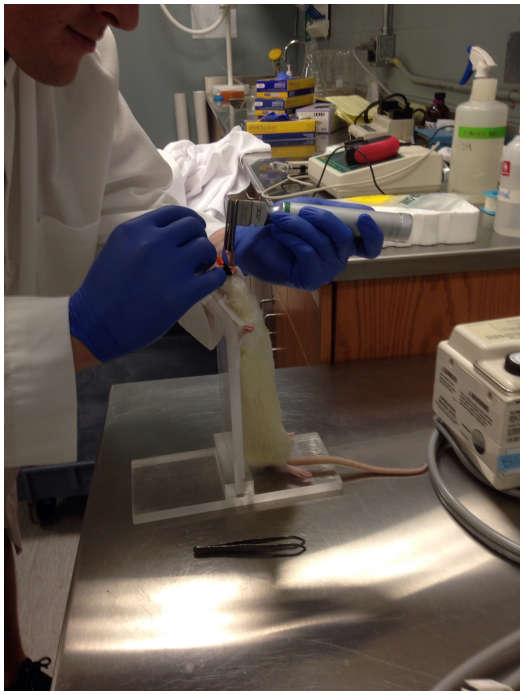

Figure 2.7: A medical expert carefully mounts the rat to a suitable frame and prepares it for the incubation equipment.

this step, experts wait for at least an hour to make sure that the injected cells circulate through the rat's body and reach the desired organ. After this, an MRI expert performs the imaging under a specific field strength using a specific echo time (TE).

Figure 2.8: (Top) Incubation equipment is attached to the rat. (Bottom) A general view of the incubation procedure. The equipment displays the status of the rat's breathing process.

Note that in addition to performing *in vivo* MRI (those involving living animals), *in vitro* scans were also obtained. Tab. 2.1 shows the basic details of the collected database. A set of 33 *in vitro*

scans and 7 *in vivo* scans were collected. The specific details of our collected *in vitro* and *in vivo* MRI along with the corresponding scanning details are presented next.

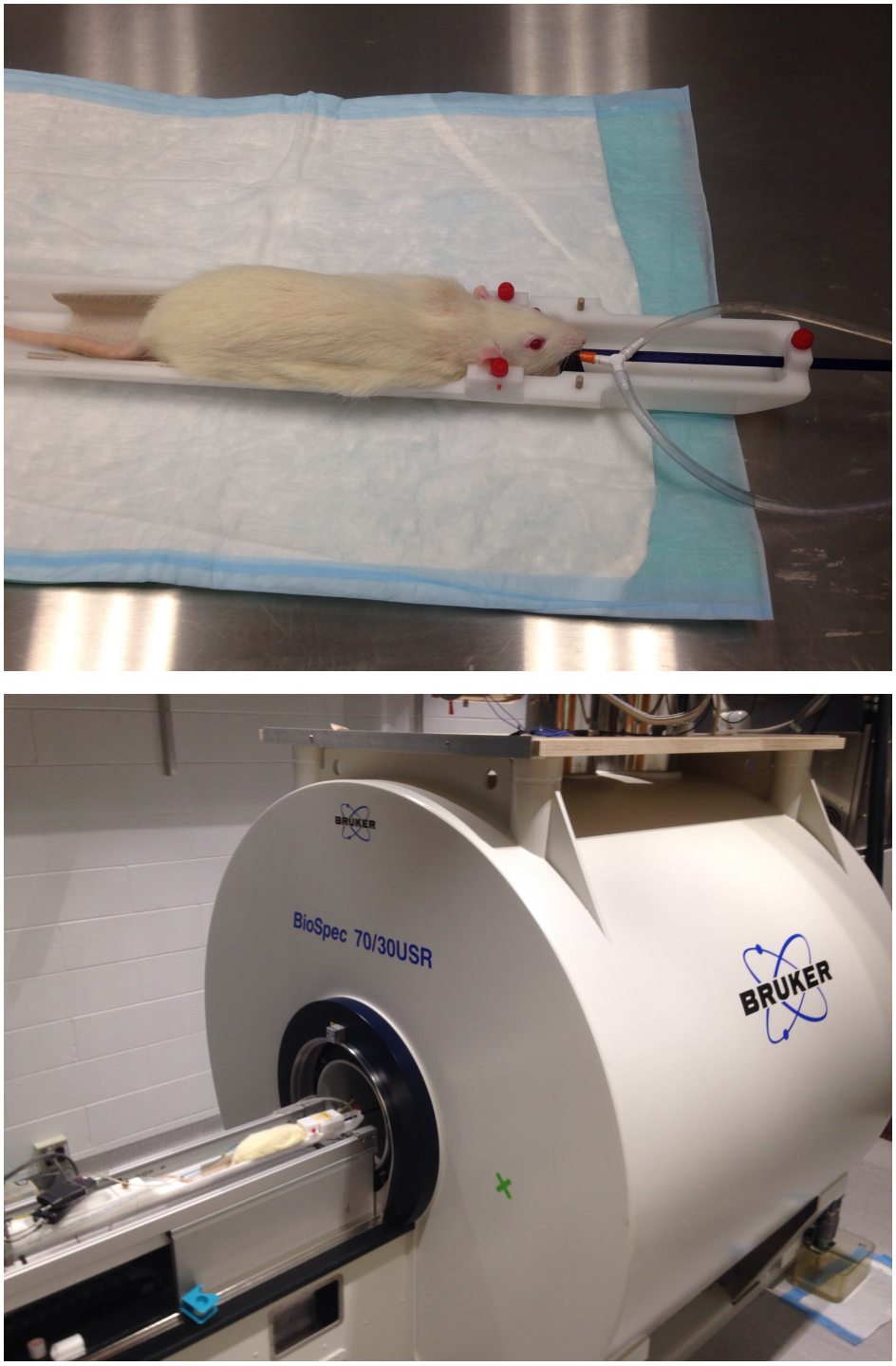

Figure 2.9: (Top) Mounting the rat to the MRI machine's mechanical frame. (Bottom) Rat to undergo an MRI

Table 2.1: Collection details and characteristics of our MRI database

Set	Type	Subject	Labeler	Machine	Labeled Scans	Total Labels	Resolution	Size
G_A	*in vivo*	Brain	R_1	11.7T	$G_{1A}, G_{2A}, G_{3A}, G_{4A}, G_{5A}$	15442	$100\mu m$	$256 \times 256 \times 256$
G_B	*in vivo*	Brain	R_2	7T	G_{1B}, G_{2B}	2992	$100\mu m$	$256 \times 200 \times 256$
G_C	*in vitro*	Tube	R_2	7T	$G_{1C}, G_{2C}, G_{3C}, G_{4C}$	814	$100\mu m$	$128 \times 80 \times 80$
G_D	*in vitro*	Tube	R_2	7T	$G_{1D}, G_{2D}, G_{3D}, G_{4D}$	514	$200\mu m$	$64 \times 40 \times 40$
G_E	*in vitro*	Tube	$\bar{t}$	7T	$G_{1E}, G_{2E}, G_{3E} ..., G_{25E}$	(2400×25)	$100\mu m$	$100 \times 64 \times 64$

2.2.3.1 *In vitro* MRI scans

Imaging phantoms were constructed consisting of a known number of 4.5 micron diameter, magnetic microparticles with 10 pg iron per particle, suspended in agarose samples. Each microparticle approximates a single magnetically labeled cell with appropriate iron content for MRI-based single cell detection [13]. T2*-weighted gradient echo MRI was then performed on these samples at a field strength of 7T.

As can be seen in Tab. 2.1, these scans have variation in resolution, matrix sizes, and amount of spots (labels). G_E has 25 data sets, collected from 5 samples under 5 different MRI conditions. These conditions were variations in TE from 10 - 30 ms (signal to noise $>$ 30:1), and images with low signal to noise ratio ($\sim$ 8:1) at TE = 10 and 20. The effect of increasing TE is to enhance the size of the spots. The higher the TE, the larger the spot [13]. The downside of higher TE is that the physics which governs enlargement of the spot, the difference in magnetic susceptibility between the location in and around the magnetic particles and the surrounding tissue, also causes the background tissue to darken. The rationale to collect images with both high and low signal to noise ratio is to test the robustness of our spot detection procedure in two potential *in vivo* scenarios. Manual ground truths were collected from experts on 8 *in vitro* MRI scans of G_C and G_D. Note that, to study the effect of change in image resolution, G_D was obtained using a low resolution MRI. For G_E, the theoretically computed ground truth was known. This set was used for a direct comparison between the automatically detected spots and the theoretically expected number.

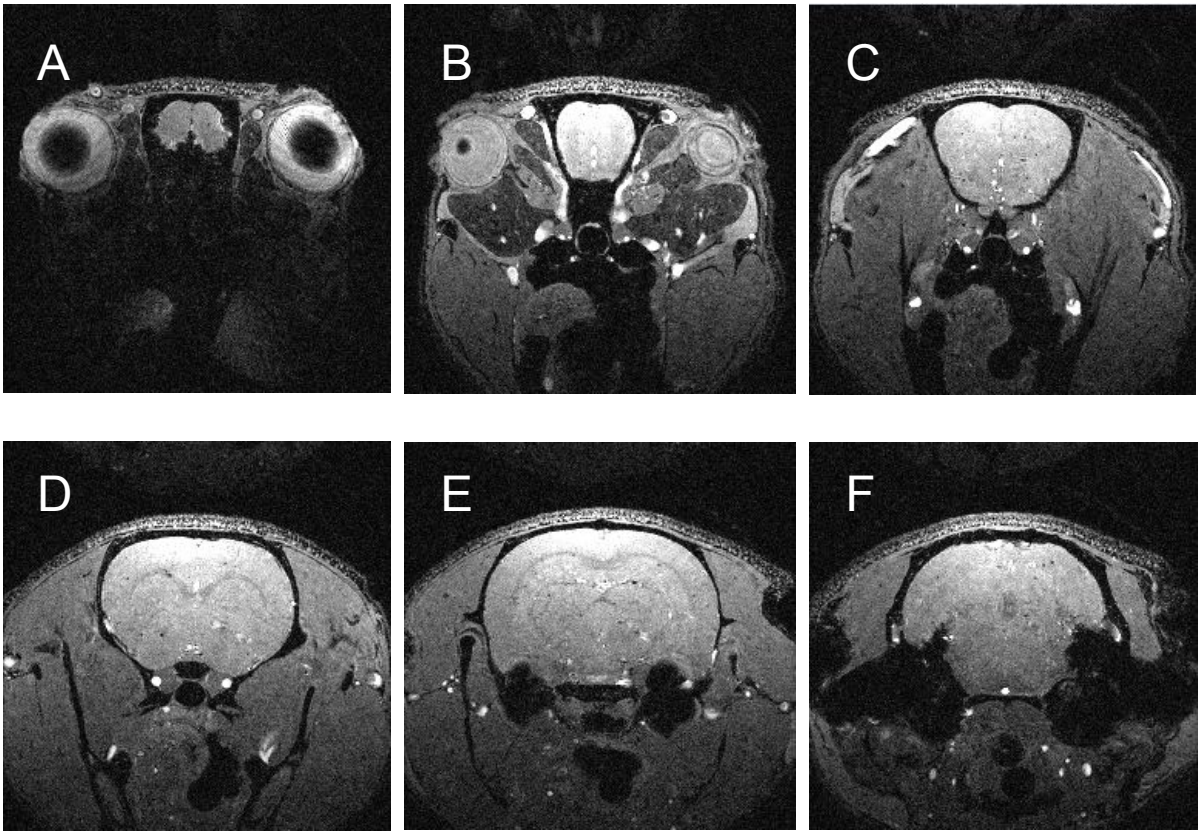

Figure 2.10: A-F show variation in the brain morphology across MRI slices.

2.2.3.2 *In vivo* MRI scans

Two different sets of *in vivo* MRI were collected from two different machines having different field strengths. Using one machine with a field strength of 11.7T, 5 MRI scans of rats were collected, which are denoted as G_A in Tab. 2.1. Three of them were injected intracardiac, $1-1.5$ hours prior to the scan, with rat mesenchymal stem cells (MSCs) that had been labeled with micron sized iron oxide particles (MPIOs) to a level of $\sim$14 pg iron per cell. This transplantation scheme delivers cells to the brain - an intravenous injection would deliver cells only to the liver and lungs. Two additional rats were not injected at all. Using another machine with 7T, 2 additional brain MRI scans of rats were collected. These were also transplanted with MPIO labeled MSCs. G_B is used to

denote these 2 scans in Tab. 2.1. The rationale behind collecting these two different *in vivo* sets was to be able to validate the generalization and robustness of our learned algorithm against potential variations arising from different imaging systems. Note that a different amount of MSCs were injected in different rats to achieve further variations in the data. All MRI were 3D T2*-weighted gradient echo.

2.2.4 Label collection

To collect labels on data, a Labeling tool was designed with the assistance of a medical expert. This tool was designed and tested several times by an expert to meet the requirements of this research work. The final tool that was adopted by the expert is available as an executable and can be conveniently utilized without installing Matlab. The tool allows a medical expert to analyze images and zoom in to specific portions of the image, at the pixel level, if necessary. An option for contrast adjustment is also provided. Experts can view some basic statistics on the interface of the software tool, e.g., the total number of spots labeled, labeled spots on the current slice, slice number, etc. If required, the experts can delete a previously labeled point and immediately skip to a different slice in the MRI.

In addition to collecting traditional data, the tool also captures aspects related to the labeling behavior of an expert. For example, the time taken to label each point, overall time spent on each slice, number of keyboard hits, time of the day, number of deleted points, etc. are also recorded.

Note that these labels represent the entities that a human expert considers as spots/cells in the MRI. Due to human labeling error, it is also possible to have some MRI noise being incorrectly marked as spots and spots being confused with MRI noise. On the other hand, for the set G_E, the number of spots is theoretically computed. However, considering that the process of preparing and injecting spots are manually conducted, the actual spot numbers in these scans may not be exactly

the same as the theoretically estimated number.

2.2.4.1 Data loading and slice selection

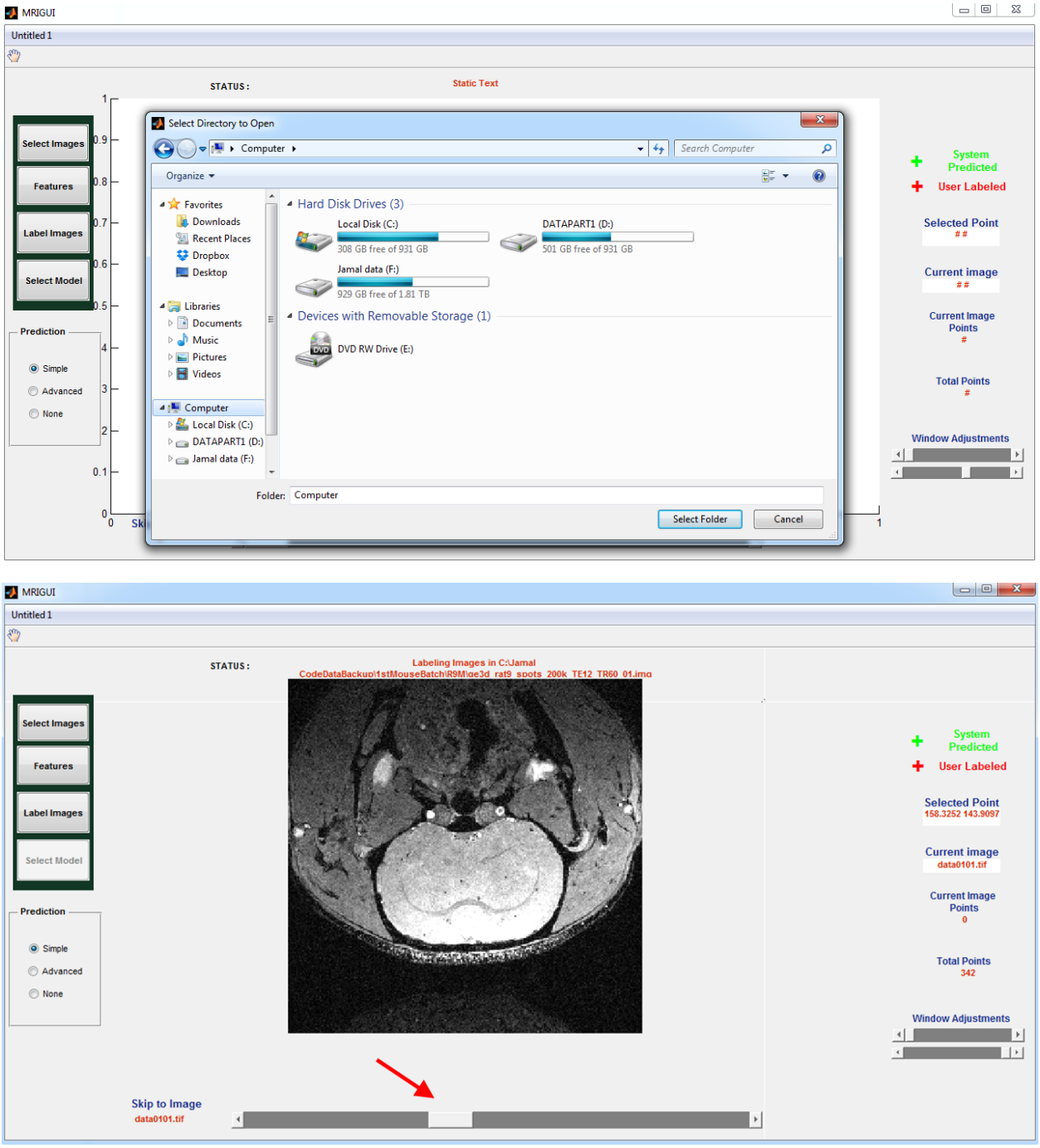

Figure 2.11: (Top) The software interface provides an option to browse to the directory containing the MRI data. (Bottom) Once the data is loaded, an expert can begin labeling from any slice using the slider indicated with a red arrow.

2.2.4.2 Zooming-in to pixel level

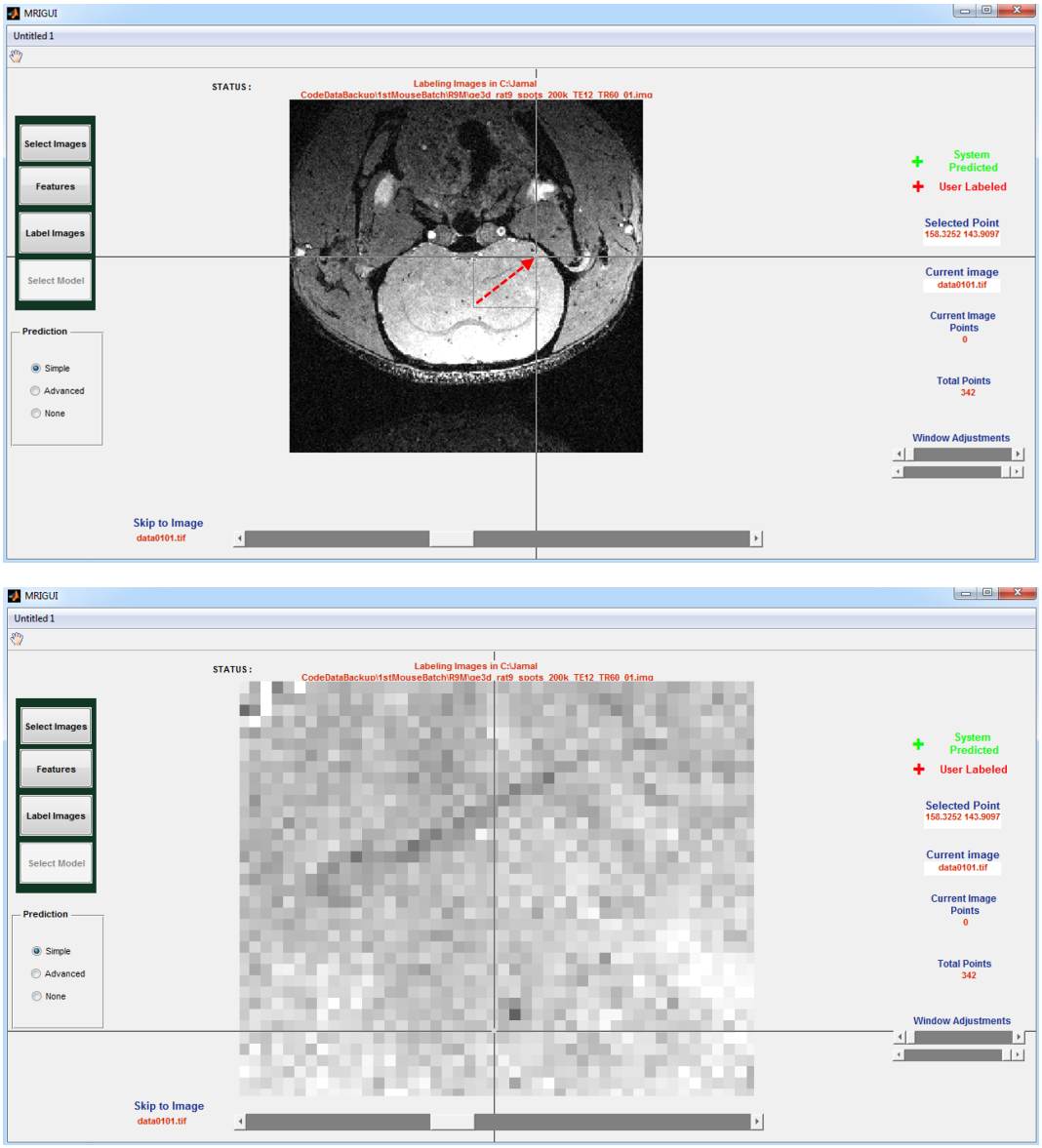

Figure 2.12: (Top) For zooming-in, the operator simply click and drag in the direction shown with the red arrow. This creates a boxed region that will appear in the zoomed-in view. This process can be repeated multiple times if further zoom-in is required within that boxed region. (Bottom) The corresponding zoomed-in view.

2.2.4.3 Operating a Zoom-out

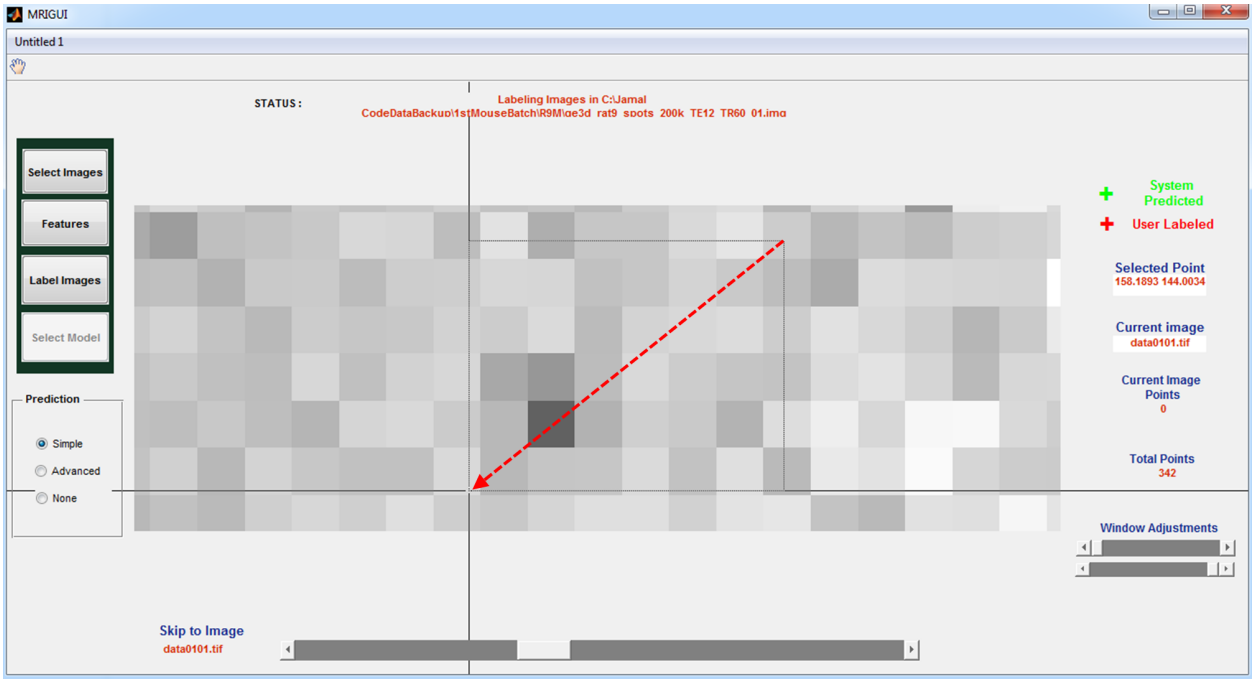

Figure 2.13: Illustrating the zoom-out operation. The expert clicks and drags along the diagonal direction as indicated by the red arrow. This operation brings up the original labeling view.

2.2.4.4 Labeling statistics and contrast adjustment

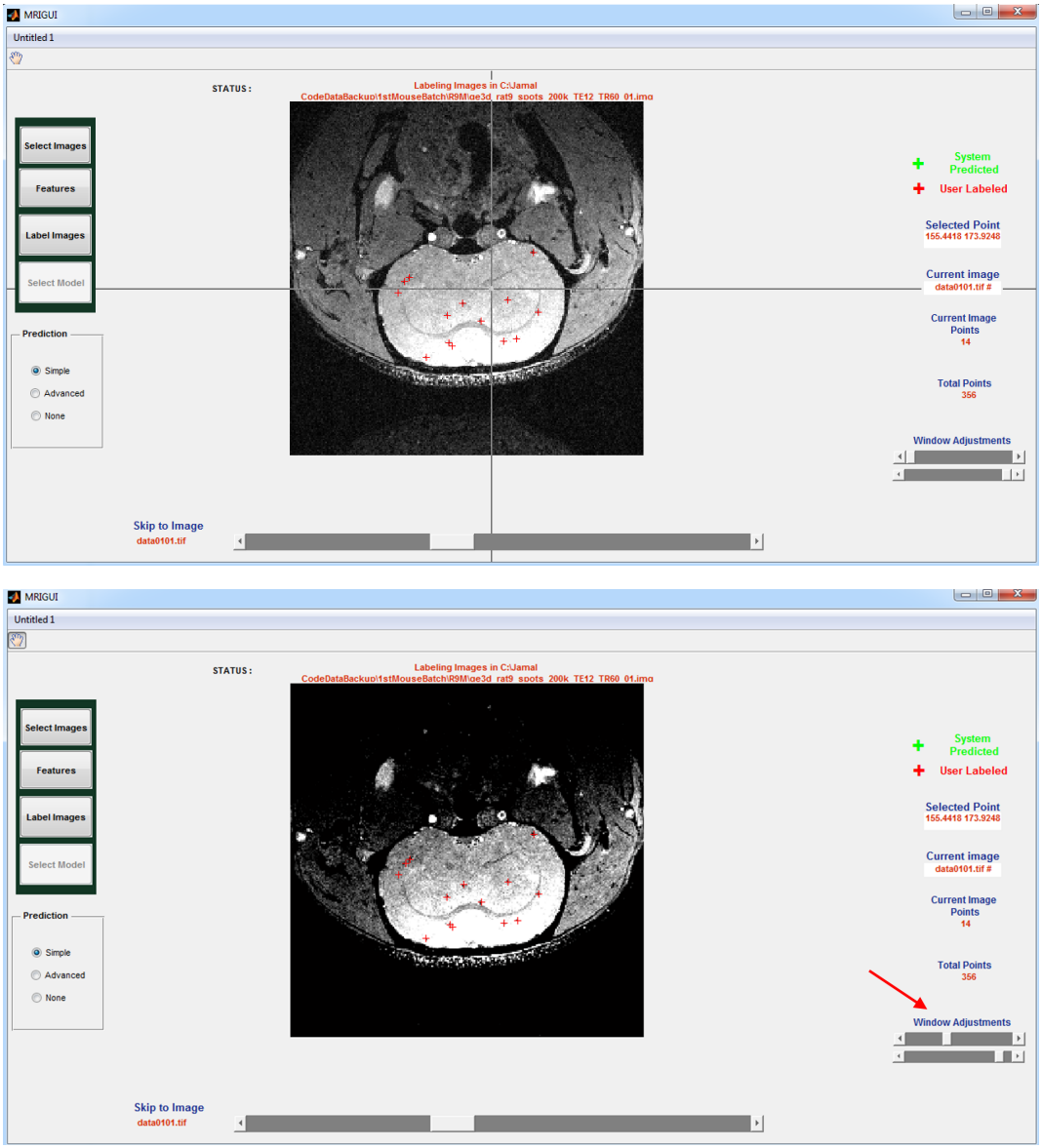

Figure 2.14: (Top)The expert labels are overlaid on the MRI slice.The operator uses a left-click to indicate a label. A label can also be deleted by clicking on it again. Basic labeling statistics, such as location of the last labeled point, total number of labeled spots, labeled spots on the current slide, and the slice number, are displayed on the right side of the tool. (Bottom) Shows the effect of contrast adjustment. Note that all these operations can also be performed with the zoomed-in view.

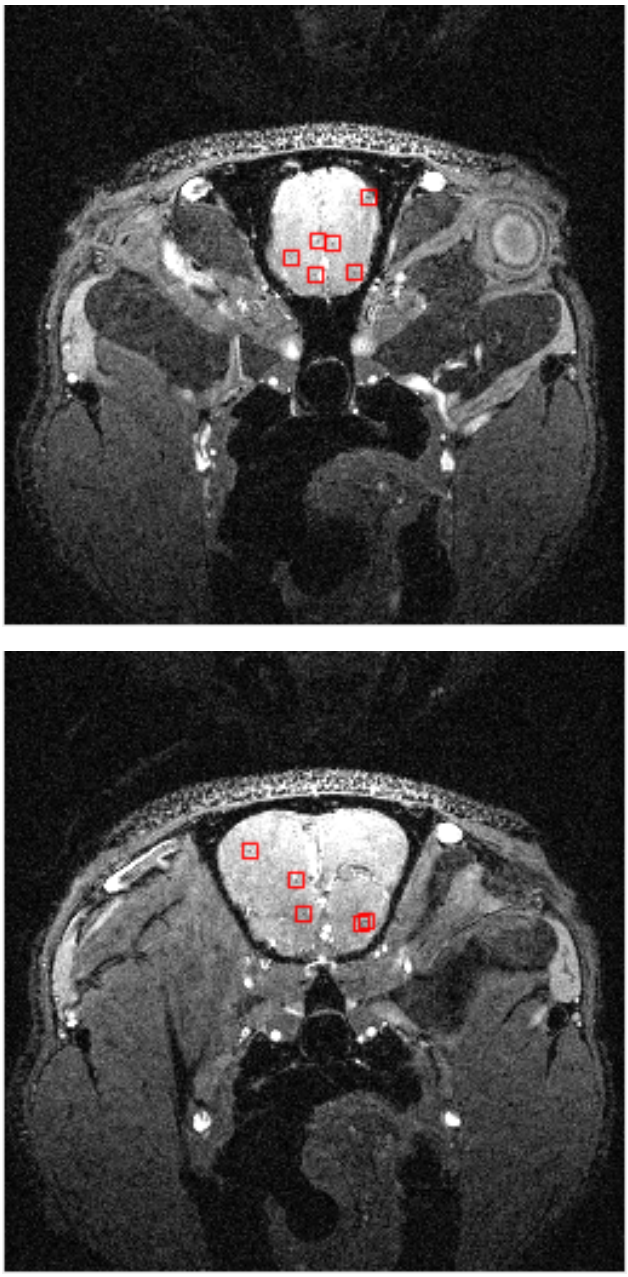

Figure 2.15: The squares represent the labels from an expert. Distribution of these labels on two MRI slices is shown here.

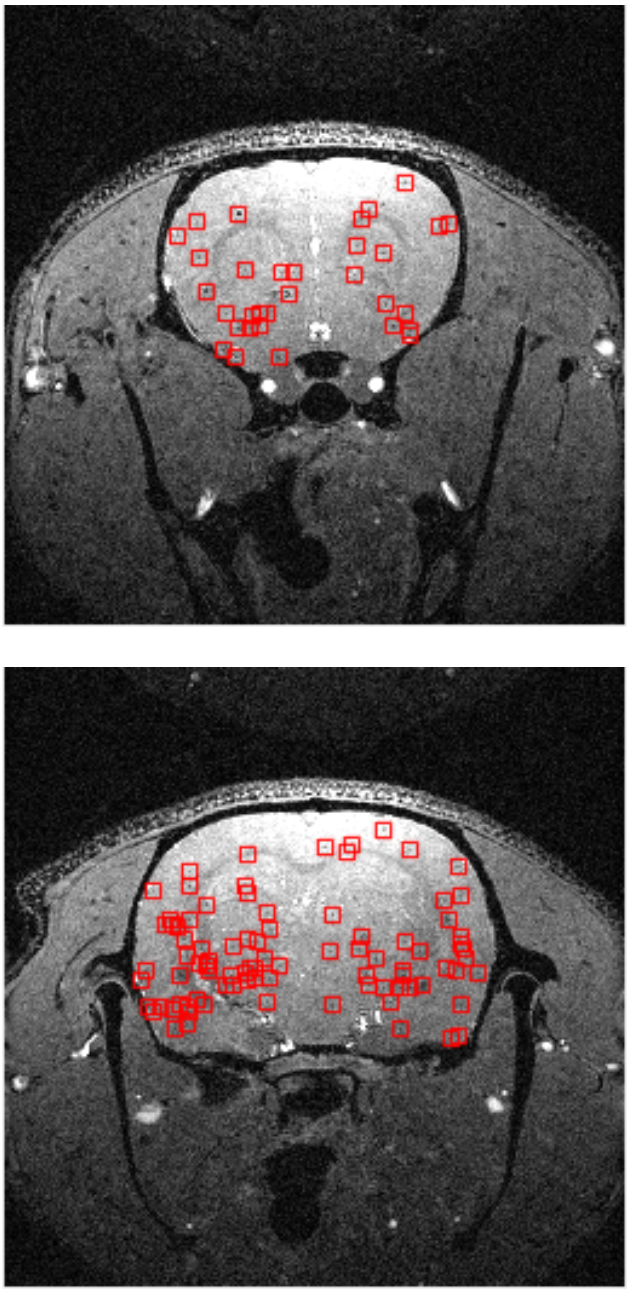

Figure 2.16: The squares represent the labels from an expert. Distribution of these labels on two MRI slices is shown here.

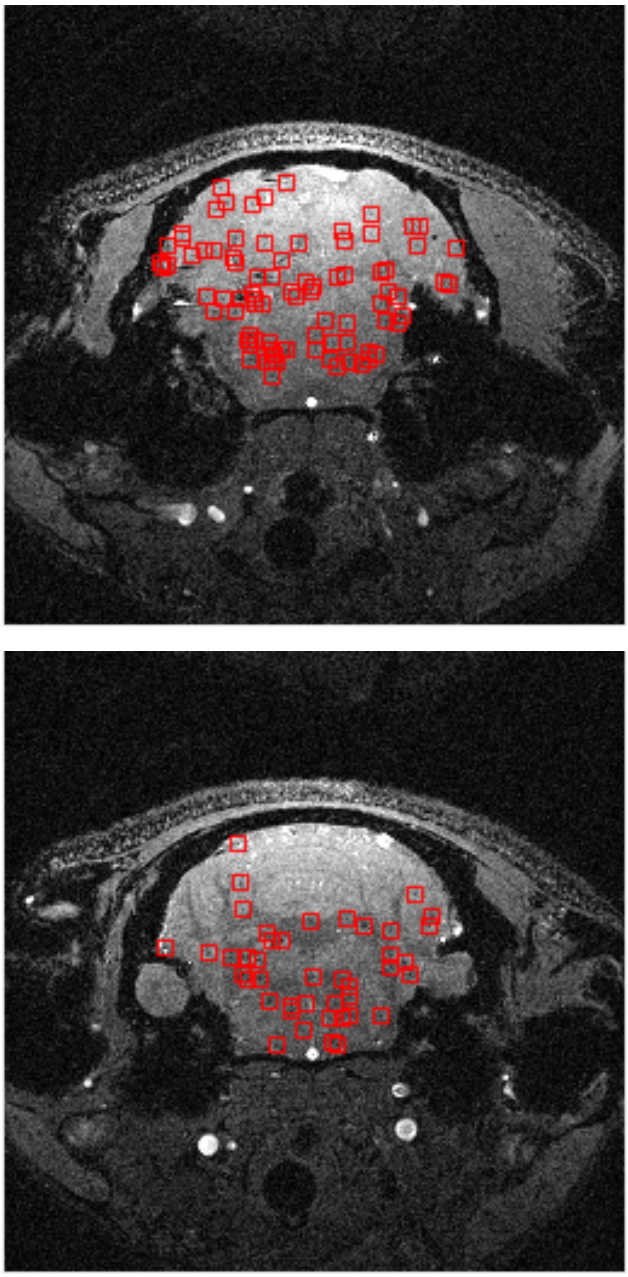

Figure 2.17: The squares represent the labels from expert. Distribution of these labels on two MRI slices is shown here.

Chapter 3

Regions-of-Interest and Feature Representations

3.1 Introduction

In machine learning, a classification approach maps a real world problem into a classification task where two or more entities (classes) are to be intelligently distinguished from each other (see [22] for basic details). For example, classifying a potential candidate region in MRI as spot or non-spot can be viewed as a classification task. However, in the context of this work, the first challenge is how to effectively extract these potential candidate regions, called *Regions-of-Interest(RoI)*, from an MRI scan. Should the RoI be based on one pixel, two pixels, how many pixels? what will be a systematic approach to extract such RoI? Once extracted, the RoI can be input to a classifier that can be constructed using different classification paradigms. The classifier will then learn to distinguish a spot from a non-spot in these RoIs.

In the context of this work, classification paradigms can be categorized into two fundamental types. In the first paradigm type (P-1), discriminating information is extracted from the images (RoI in this case) using a pre-defined approach that is designed by an expert based on intuition and experience. This information may be in the form of a numeric array of values known as *features*. For each RoI such features along with their ground truth classification labels are then forwarded to another algorithm called *classifier* or *classification technique* which learns to distinguish between

the classes.

In the second paradigm type (P-2), the feature representations are not manually designed by an expert but rather automatically learned from the data. Generally, both feature representations and the classifiers are learned automatically in a single unified framework. Many neural network based approaches fall into this category which can take image datasets directly as input, along with the labels, and learn a classification model.

Note that both the aforementioned approaches require RoI as input. Therefore, in the next section we first propose an effective superpixel based strategy to extract RoI and then investigate the design of approaches that belong to both the classification paradigms.

3.2 Approach

3.2.1 Generating RoI

The first challenge in this research is to define the RoI. Processing each pixel an RoI can result in a huge computational burden. We addressed this issue by extracting *superpixels* from each MRI scan [18]. A superpixel technique groups locally close pixels with similar intensities into a single unit. Each unit is called a superpixel. Superpixel-based methods are becoming increasingly popular. For example, authors of [23] discuss how the superpixels extracted using different techniques can be combined to achieve better image segmentation. Similarly, various studies utilize superpixels for classifying local image segments. In [24], authors use a multi-scale superpixel classification approach for tumor segmentation. Furthermore, superpixels have been utilized in various other applications as shown in [25, 26, 27]. Since spots are usually darker than their surrounding, they are characterized as superpixels with lower average intensity than the surrounding superpixels. This superpixel based model of a spot as illustrated in Fig. 3.1.

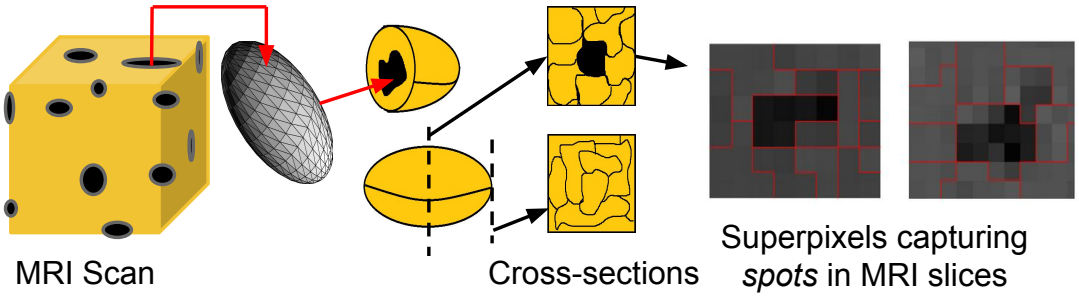

Figure 3.1: A diagrammatic representation of a spot in MRI slices. The figure also shows two real spots in MRI slices and how they were captured by superpixels.

Based on this idea, a novel set of features utilizing the average superpixel intensities, was proposed in our study in [18] (*see supplementary material in Chap 6 for details*). However, this approach has the following limitations: (1) The accuracy of the approach was dependent on the preciseness of the superpixel algorithms. (2) The approach assumes a superpixel based model for a spot in terms of its depth across consecutive MRI slices. This does not hold true for all spots in different MRI settings.

The strategy adopted in this thesis is resilient to imprecisions in the superpixel extraction algorithms. Based on each superpixel unit, a representative patch is extracted from the MRI scan as explained in Fig. 3.2. Each patch is then taken as a candidate region and undergoes a feature extraction process. The approach is *model-free* and imitates the strategy adopted by a human labeler. All candidate patches are first detected in 2D MRI slices and then neighboring patches detected in consecutive slices are connected without imposing any restriction on their depth in 3D.

The spatial location of each patch in MRI is also recorded. Consequently, these extracted patches are forwarded to the machine learning algorithms as input data.

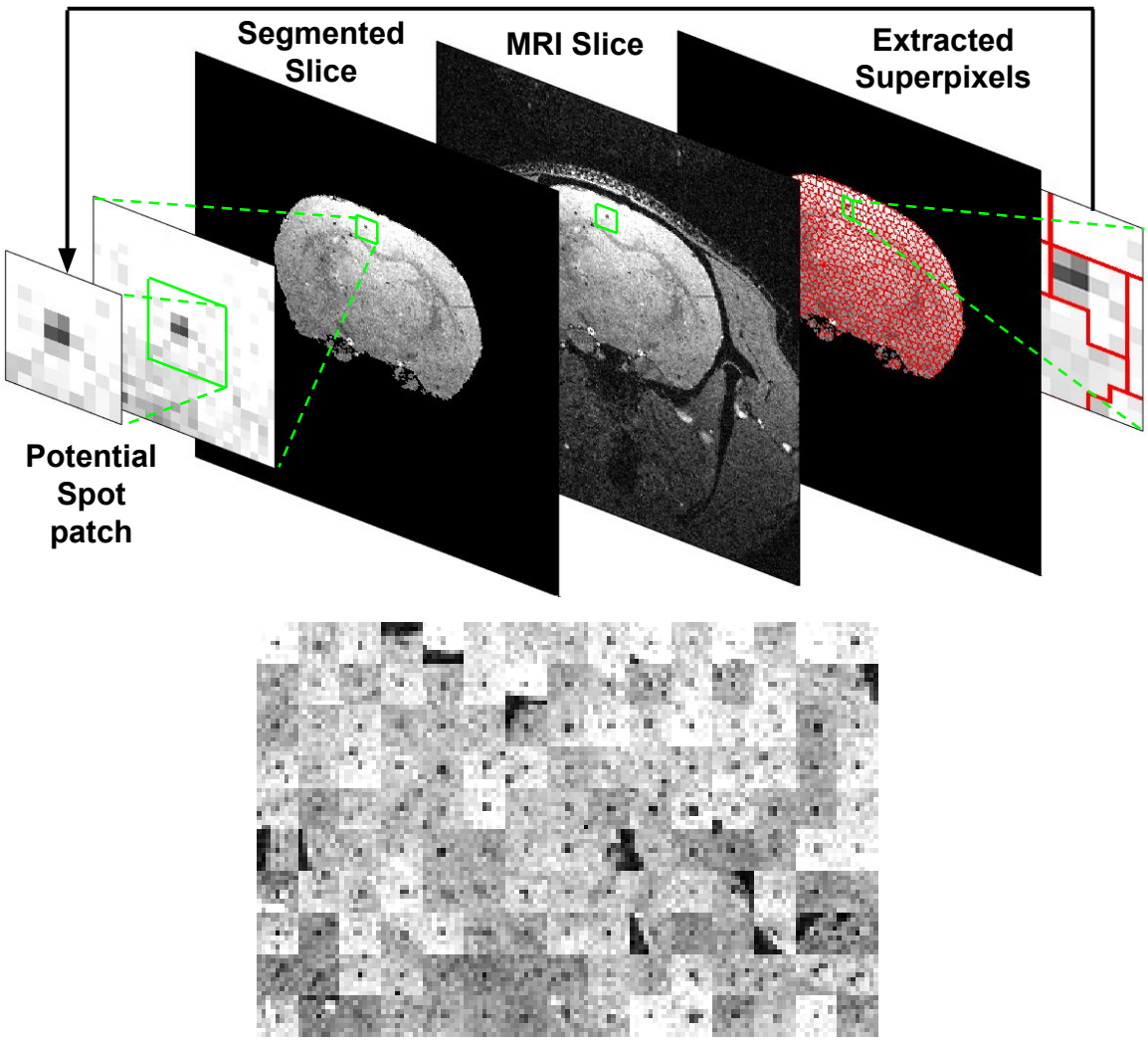

Figure 3.2: (Top) Illustrating the generation of candidate regions: A superpixel algorithm is first applied to each slice in MRI and then the brain region is automatically segmented using basic image processing techniques. The superpixels that correspond to only the segmented brain region are considered and the rest are ignored. For each such superpixel, the darkest pixel is selected as the center and a fixed size patch is extracted around it. (Bottom) A mosaic of several 9×9 patches extracted from an MRI slice. It can be seen that all patches have a dark region in the center representing a spot in a 2D slice.

3.2.2 Feature extraction

3.2.2.1 Feature extraction with fixed designs (P-1)

This is the traditional and most widely adopted paradigm in computer vision and pattern recognition based studies [18, 16]. Many studies on automated cell detection in microscope-based imaging

are based on this paradigm [16]. Further, our initial study proposed in [18] can also be categorized into this particular paradigm. In this study, an elaborate set of feature extraction methods are utilized that extract shape, intensity, texture and context information about the entities in the candidate patches. Fig. 3.3, Fig. 3.4, and Fig. 3.5 present a brief explanation on how hand-designed features can be extracted specifically for the task of capturing *spot appearance* in MRI.

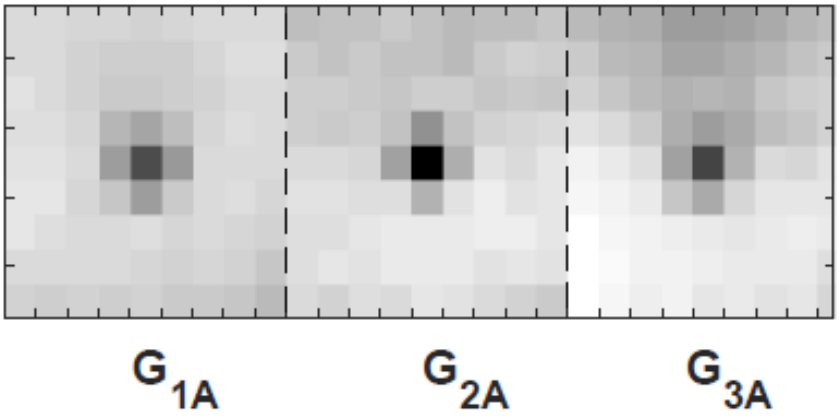

Figure 3.3: Principle Component Analysis (PCA) was utilized to extract eigen spot shapes using all of the 9×9 spot patches in the training set. The top PCA components for the spot patches obtained on three labeled rats in G_A are shown here. An iteratively increasing threshold is then applied on the values of these top PCA components to extract different binary patches that are utilized as filters to capture the shape and intensity information on spot patches.

All the extracted features are finally concatenated to form a feature vector for each candidate patch. Usually, in long feature vectors, some features are irrelevant or redundant. Therefore, from the obtained feature vector, the most useful features are selected and the irrelevant features are eliminated using a feature selection module that employs a correlation based feature selection algorithm [28]. These selected feature vectors along with their corresponding labels for the patches are then forwarded to tune a classifier. In this study, a diverse group of well-known classifiers such as probabilistic (Naive bayes), functional (Multi-layer perceptron(MLP)), and decision tree (Random Forest), are utilized and compared (see [22] for details).

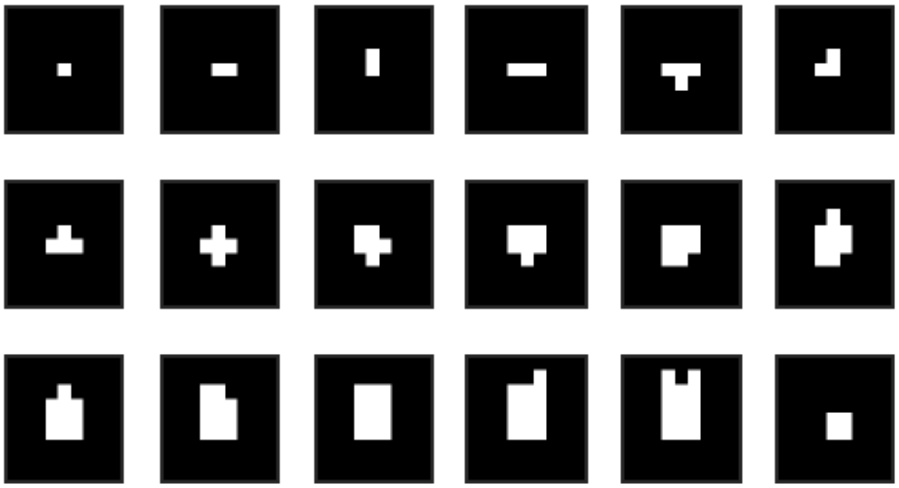

Figure 3.4: Binary shape filters are obtained using the top PCA components. By iteratively increasing the threshold values from dark to light intensities, PCA components can result in different binary shapes. Domain experts agree that these binary filter patches represent many frequent shapes of the actual spots. All these patches are rotated and translated to obtain a large set of different shape filters. These filters are convolved with each candidate patch and the computed response is taken as a feature. A large set of these responses comprehensively capture the shape and intensity information of spot patches.

3.2.2.2 Feature extraction with learned designs (P-2)

Based on expert intuition and experience, features extracted in P-1 can be subjective. Therefore, the key goal of P-2 approaches is to *automatically* learn the most optimal spot feature representation from the data. Neural networks are a well-known example of the P-2 approaches.

Deep convolutional neural networks (CNN) [29, 19] have been highly successful in many image based ML studies. Before, we discuss the design of the proposed CNN architecture for this task, a brief introduction of the well-known CNN architectures is presented as follows:

- **LeNet:** This is one of the first CNN architecture proposed by Yann LeCun in the 1990's. This architecture was applied for automatically reading zip codes, hand written digits, etc. More details on this architecture can be seen in [30].

- **AlexNet:**This architecture popularized the use of CNN architectures in modern computer

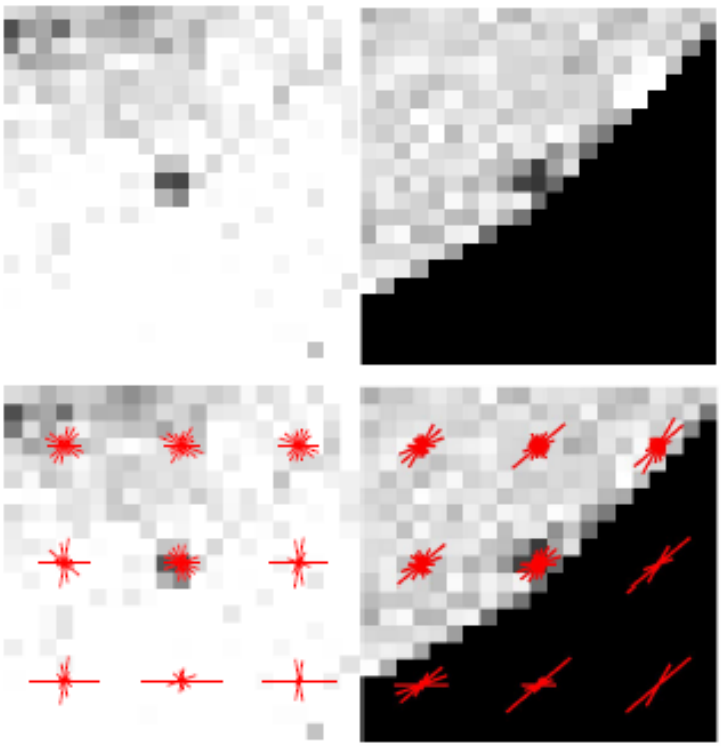

Figure 3.5: Visual representation of context features for two context patches: While learning the definition of a spot, it may also be useful for a classifier to learn more about its surrounding context. Therefore, to capture the appearance of the patch's context, a larger patch of 21×21 was extracted around the center of a candidate patch x_i. Two well-known appearance descriptors in computer vision were then used to extract features: (a) Histogram of Gradients (HoG) [1] and (b) Gist [2]. A visual representation of HoG for two context patches is shown here. Red lines here indicate the different directions of intensity gradients whereas the lengths of the lines determine their magnitude.

vision. This was proposed in 2012 by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton. This architecture was more deeper than LeNet and significantly performed superior to other approaches on the ImageNet ILSVRC-2012. Their approach popularized the use of rectified linear units (ReLU) as non-linearities in the CNN architectures. Also, their use of dropout technique to selectively ignore neurons in the training phase was considered effective to avoid over-fitting. More details can be seen in [29].

- **Overfeat:** Overfeat architecture was the winner of the localization task of the ILSVRC-

2013. This architecture can be seen as a derivative of the AlexNet CNN architecture. Overfeat also obtained competitive results for the detection and classifications tasks in ILSVRC-2013. More details can be seen in [31].

- **ZFNet:** This CNN architecture was proposed by Matthew Zeiler and Rob Fergus and hence became famous as ZFNet. This architecture is also a derivative of the basic AlexNet architecture. This architecture became famous after its high accuracy on ILSVRC-2013. The details on ZFNet can be seen in [32].

- **VGG Net:** VGG was the runner-up of the ImageNet ILSVRC-2014. VGG presented a more deeper CNN networks which resulted in a better performance. For example, VGG-16 (16 layered) and VGG-19 (19 layered) were significantly deeper than AlexNet and its derivatives. Another interesting property of their architecture was the use of very small filters for convolution (3×3) and pooling (2×2). Further details on VGG CNN architectures can be seen in [33].

- **GoogLeNet:** This CNN architecture by Szegedy et al. from Google was the winner of the ImageNet ILSVRC-2014. Due to the use of their proposed inception module, this CNN architecture was able to significantly reduce the number of parameters in their network. For more details on GoogLeNet see [34].

- **ResNet:** Kaiming He et al. in [35] designed Residual network which was the winner of ILSVRC 2015. It utilizes special *skip connections* which allows the lower layers to be connected with the higher layers. ResNet also allows for learning much deeper CNNs to improve performance. More recent work of Kaiming He can be seen in [36].

Note that unlike P-1 approaches, features were hierarchically learned in all the aforementioned

CNNs using multiple layers in an *automatic fashion*. In these CNN architectures, feature extraction and classification was performed in a unified framework. However, these architectures were designed for standard computer vision tasks where a large entity is of interest in the image.

Consider, $M = f()$ as an overall classification model learned by our P-2 approach. In the context of this work, let this M denote a sequence of convolutional filters and transformation functions that will serially applied to an input image to output a classification decision. Considering this, f can be decomposed into multiple functional layers:

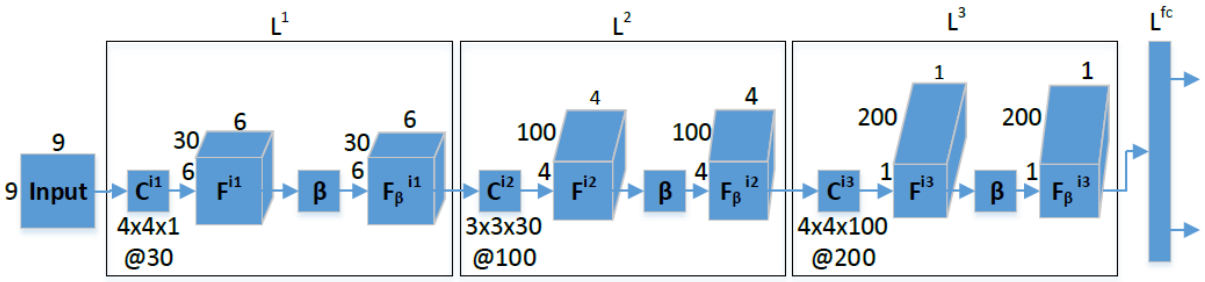

Figure 3.6: CNN architecture used in this work. The network takes a 9x9 image patch as input for a classification task i. Each composite layer L^k , $k \in \{1,2,3\}$, is composed of a convolutional layer C^{ik} which produces the feature maps F^{ik} and a non-linear gating function β producing the transformed feature maps F^{ik}_{β}. After passing through the composite layers, the net passes through the fully connected layer L^{fc} which produces the output. The *softmax* function is then applied to the output. Note that in the context of this work, this architecture represent the model M. The weights of all the filters across its processing layers are learned using the training data.

$$f() = (f_u \circ f_{(u-1)} \circ f_{(u-2)} \circ ... \circ f_1). \tag{3.1}$$

Here, each function, f_j, $j \in [1,u]$, can represent a (a) convolutional layer, (b) non-linear gating layer, (c) pooling layer, (d) full-connected layer (see [29, 37, 19] for more details). For a given task, weights for these convolutional filters are learned automatically using the training data. Different architectures of a CNN are created by utilizing different number of layers and also by sequencing these layers differently. CNN architectures also vary depending on the choice of the non-linear

gating function. Filter sizes for convolutional layers are also determined depending on the application at hand. Well-known CNN architectures such as AlexNet [29] or GoogLeNet [34] cannot be utilized for spot detection in MRI. Therefore, a new CNN architecture, specifically designed for spot detection in MRI, is proposed here. The proposed CNN architecture has 3 composite layers and 1 fully connected layer (see Fig 3.6). Each composite layer consists of a convolutional layer and a gating function. Note that in a conventional CNN architecture, a pooling layer is also used which reduces the dimensionality of the input data. However, a pooling layer is not utilized in this architecture due to the small size of the input patches (9×9). Using a pooling layer, in this context, may result in the loss of valuable information which may be essential to be utilized by the next layers. Further, a gating function is usually added for introducing non-linearity into a CNN. Without non-linear gating, a CNN can be seen as a sequence of linear operations which can hinder its ability to learn the inherent non-linearities in the training data. In conventional neural networks, a sigmoid function or a hyperbolic tangent function was generally utilized for this purpose. However, in recent studies, utilizing ReLU (Rectified Linear Units) has shown significantly superior results for this role [29]. Therefore, the proposed architecture uses ReLU as a non-linear gating function.

Further customizing to the task at hand, the sizes of all the convolutional filters were kept small. However, their numbers were kept high. The goal was to provide a higher capacity to the CNN architecture for capturing a diverse set of local features of a patch. Filter sizes and dimensions of resulting feature maps can be seen in Fig. 3.6 (see Fig. 3.7 for learned filters). For any task i, the proposed model (CNN architecture) can be written as

$$M = (\gamma \circ L_{fc} \circ \beta \circ C^{i3} \circ \beta \circ C^{i2} \circ \beta \circ C^{i1}). \tag{3.2}$$

Table 3.1: Experimental comparison of *in vivo* spot detection performance using P-1 and P-2.

	Algorithms	J_1	J_2	J_3	J_4	J_5	J_6	means
P-1	Random Forest	94.0	86.9	95.3	94.1	86.0	94.7	91.8 ± 4.2
	Naive Bayes	82.9	81.8	84.3	84.1	80.1	83.7	82.8 ± 1.6
P-2	CNN	**96.4**	**92.3**	**96.1**	**96.4**	**91.2**	95.0	94.6 ± 2.3
	MLP	91.1	85.2	90.9	91.4	84.2	90.3	88.9 ± 3.3
	MLP (P-1/2)	93.9	89.4	95.8	95.4	90.0	**95.7**	93.4 ± 2.9
means		91.7 ± 5.2	87.1 ± 4.0	92.5 ± 5.0	92.3 ± 4.9	86.3 ± 4.5	91.9 ± 5.0	

where γ represents a standard softmax function that can be applied to the output of the fully connected layer L_{fc}. β denotes the non-linear gating function and C^{ik} represents the convolutional layer in the composite layer k.

3.3 Experiments, results and discussion

Experiments were performed to answer the following main questions: (1) Which of the two ML technique results in the best detection accuracy for *in vivo* spots in MRI? (2) How does the best approach perform on *in vitro* evaluation studies? (4) Can a ML approach learned on *in vivo data* be tested for spot detection on *in vitro* data? (5) How is the performance affected if the MRI is conducted at low resolution? (6) Is the proposed approach robust to the differences in MRI machines in terms of field strength, make and model etc.? Importantly, it is also of interest to investigate how the theoretically computed number of spots for *in vitro* MRI scans compares with the automatically detected spot numbers.

3.3.1 *In vivo* evaluation studies

In this study, the spot classification performance of a diverse set of approaches was evaluated using the two sets of *in vivo* MRI scans i.e G_A and G_B. First, experiments and results are discussed for G_A that has 5 different MRI scans obtained from one MRI machine and labeled by one expert. Three

of these *in vivo* scans contain spots that were manually labeled by experts whereas the remaining two were naive. Six combinations of testing and training *pairs* are created such that two scans are always present in the testing set of each pair, where one of the scans is a naive and the other contains spots. The remaining 3 out of the 5 scans are used for training the ML algorithms. Note that each MRI scan resulted in about a $100,000$ candidate patches and about 5000 of these represented the spots. Area Under the Curve (AUC) was utilized as a standard measure for classification accuracy. Experimental results for all the algorithms are listed in Tab. 3.1.

It was observed that the best results were achieved by a CNN, with a mean accuracy of 94.6%. The superior performance of CNN can be mainly attributed to its ability to automatically explore the most optimal features using training data rather than relying on hand-crafted features utilized in traditional machine learning. Second, CNN learn features in a deep hierarchy across multiple layers. Recent research shows that such a hierarchy provides a superior framework to CNN for learning more complex concepts, unlike traditional machine learning approaches which learns in a shallow manner [29, 34, 37].

The second best results were observed with the simple MLP approach when it takes the carefully designed, handcrafted features as an input, rather than the raw data X. This MLP can be viewed as a mixed paradigm approach (P-1/2). However, the deep learning CNN that inherently extracts hierarchical features without using any hand crafted features resulted in the overall best performance.

Probabilistic Naive bayes, using P-1, shows the worst detection performance with an average accuracy of 82.8%. This can be because naive bayes assumes complete independence between the features which in many practical problems may not be true. Further, it can be seen in Tab. 3.1 that J_2 and J_5 testing sets proved to be the most challenging with low mean accuracies of 87.1% and 86.3%, respectively, from all algorithms. Dataset J_4 resulted in the overall best performance with

mean accuracy of 92.5%. When investigating this, it was found that both J_2 and J_5 contained MRI scan G_{A1} in their test set accompanied with a different naive scan. It was seen that the labeled patches in G_{A1} were significantly more challenging in terms of morphology and intensity than those extracted from other scans.

The best two approaches, i.e., MLP (P-1/2) and deep CNN (P-2), were then further compared using another set of *in vivo* scans i.e G_B. This data was collected from a different machine having a different field strength and was also labeled by a different expert. In this study, all the previous 5 scans of G_A were used for training both approaches (creating a larger training set), and then the learned spot detection models were tested on the *in vivo* scans in $G_B = \{G_{B1}, G_{B2}\}$. Note that despite the differences in machine, its field strength, and also the labeling expert, CNN performed best with an accuracy of 97.3% whereas the mixed paradigm MLP (P-1/2) achieved 95.3%. We show the ROC curves for this test in Fig. 3.8.

3.3.2 *In vitro* evaluation studies

It can be observed that CNN yields the best result on the *in-vivo* datasets despite the simplicity of its approach. In this study, its performance is evaluated on the *in vitro* data in set G_C and G_D. Its performance is first tested on G_C that has 4 *in vitro* MRI scans each with a $100\mu m$ resolution creating a 3D matrix of $(128 \times 80 \times 80)$. Using these 4 scans, 3 different testing and training pairs were developed. Each testing and training pair has 2 MRI scans. The naive MRI scan was always kept in the test set, thereby generating 3 combinations with the remaining other sets. It was observed that CNN performed with a mean accuracy of 99.6% on *in vitro* scans. The individual ROC plots for these tests are shown in Fig 4.

A different study was then conducted to see the degradation in performance when each of the 4 *in vitro* scans are obtained with a much lower resolution of $200\mu m$ creating a matrix of

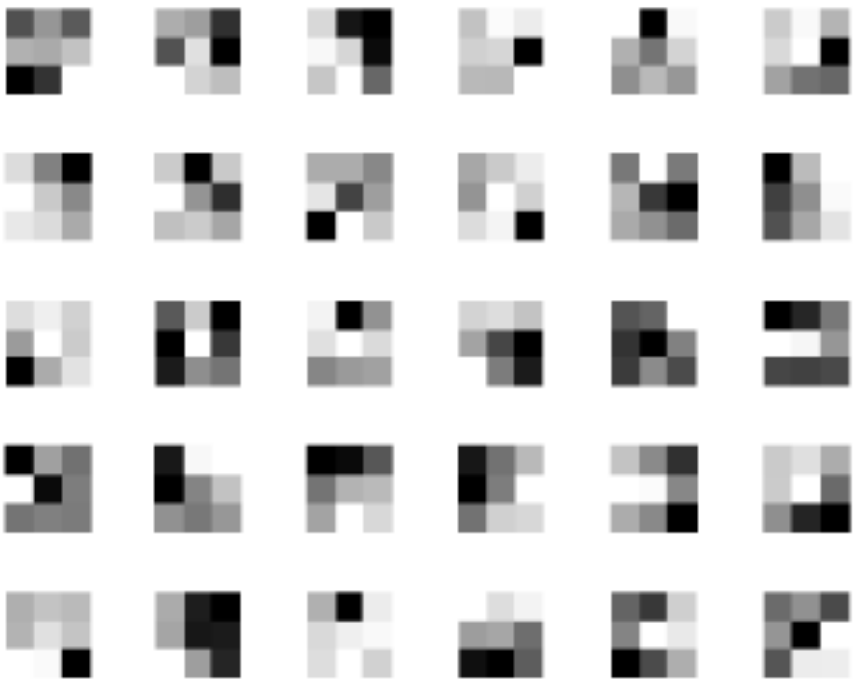

Figure 3.7: Some convolution filters learned by the 2nd composite layers of our deep learning approach. Usually, each filter acts as a neuron and due to co-adaptation between a large number of such neurons, highly sophisticated features are extracted that can potentially model spot shape, intensity, texture etc.

$(64 \times 40 \times 40)$. Such a study is desirable since in some practical applications it may be more convenient to rapidly obtain an MRI at a lower resolution, particularly in human examinations. Using the same procedure as before, three different testing and training pairs were created. It was noted that the mean performance decreased to $86.6\% \pm 5.6$. However, it was also seen that when the number of learning layers for CNN was increased to 5 (4 composite and 1 fully connected) the performance improves to $90.6\% \pm 7.1$. The individual improvements on all the three sets are shown in Fig. 3.8.

Table 3.2: Automatically detected number of spots in 5 samples under 5 conditions. The theoretically expected number of spots in each sample is 2400.

Condition	Tube 1	Tube 2	Tube 3	Tube 4	Tube 5
TE 10	2147	2272	2474	2152	2270
TE 20	2608	2750	3039	2644	2660
TE 30	2844	2993	3272	2809	2909
TE 10 (Low SNR)	1982	2023	2247	1949	2014
TE 20 (Low SNR)	2419	2563	2794	2401	2445

3.3.3 Comparison with theoretically computed spot numbers

A comparison between the automatically detected number of spots with the theoretically computed number of spots was conducted using 25 *in vitro* MRI scans of set G_E. This is an important experiment as it allows a *direct comparison* with the actual number of injected spots. All the available data from G_A to G_D was used for training a CNN and then the trained CNN model was used for testing on these 25 scans in set G_E. Each scan is expected to contain about 2400 spots. However, it is important to understand that due to the use of *manual procedures* in preparing the solution in tubes, the actual number of spots may vary about 2400. The results of automatic spot detection are tabulated in Tab. 3.2 under different MRI conditions.

3.3.4 Model generalization studies

In this section, the generalization ability of the proposed approach is determined by testing it in different possible practical scenarios. In practice, *in vivo* scans might be collected with different MRI machines at different laboratories using different field strengths. G_A and G_B represent two such *in vivo* datasets. As discussed before in the *in vivo* evaluation studies, and as shown in Fig. 3.8, the CNN based approach demonstrates robustness to such variations and achieves 97.3% accuracy despite such differences. Further, it is necessary to know how the performance would be affected

if *in vivo* data is used for training but the *in vitro* data is used for testing. Therefore, an experiment was conducted where a CNN was trained using G_A (*in vivo*) and then tested it using G_C (*in vitro*). CNN still performed with an accuracy of 96.1%. A visualization for the detected spots in *in vitro* and *in vivo* MRI scans is shown in Fig. 3.9 and Fig. 3.9 respectively.

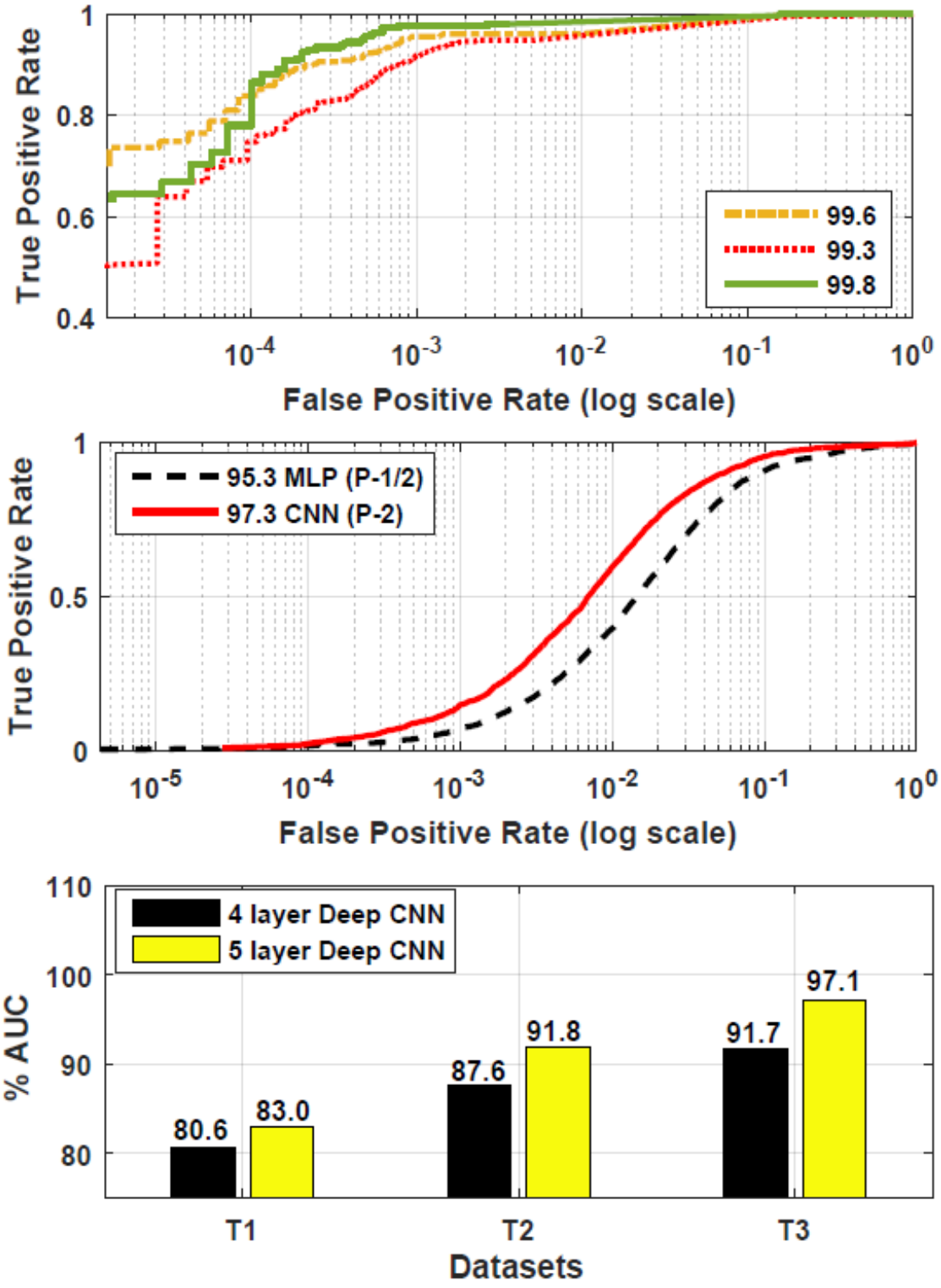

Figure 3.8: Comparison and results: (Top) *in vitro* results 100 micron, (Middle) generalization test using *in vivo* scans, (Bottom) *in vitro* results 200 micron.

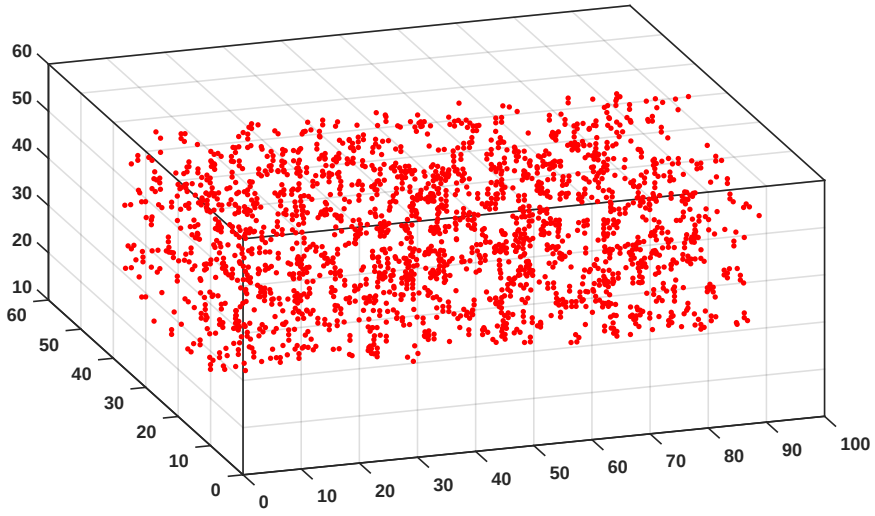

Figure 3.9: 3D visualization of the detected spots in an *in vitro* scan.

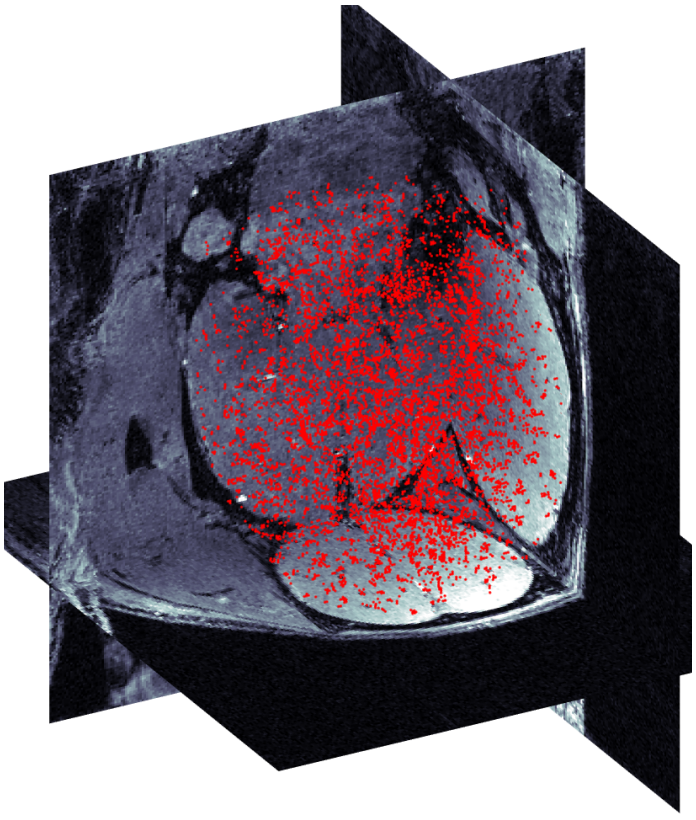

Figure 3.10: 3D visualization of the detected spots in an *in vivo* scan.

3.4 Conclusion

In conclusion, this study investigated different feature design approaches for spot detection in MRI. An approach to extract RoI from MRI was presented. A CNN architecture specific to the problem of spot detection was also proposed. Results show that features that are automatically learned using a deep learning CNN outperform hand-crafted features. Further, the proposed approach was evaluated using a diverse set of MRI scans that were obtained with variations in field strength, echo times, and resolution changes. A study was also conducted to compare its performance against the known number of spots in *in vitro* MRI scans.

Chapter 4

Learning with Small Training Data

4.1 Introduction

4.1.1 Background and motivation

One key reason behind the unprecedented success of CNNs is the availability of large application-specific, annotated datasets. However, in many practical applications, especially those related to medical imaging and radiology (e.g. spot detection), obtaining a large annotated (e.g., labeled) dataset can be challenging. In many cases, annotation can only be performed by qualified field experts and so crowd sourcing methods, such as Amazon's Mechanical Turk [21], cannot be used for annotating data. These limitations can often preclude the use of CNNs in such applications.

In order to address the problem of limited training data, the concept of *transfer learning* can be used. In transfer learning, knowledge learned for performing one task is used for learning a different task. The idea of transfer learning is not new. For example, the NIPS'95 workshop on *Learning to Learn* highlighted the importance of pursuing research in transfer learning. A number of research studies have been published in the past investigating different aspects of transfer learning as summarized in Tab. 4.2.

In case of CNNs, transfer learning typically entails the transfer of information from a selected source concept (source CNN, *learned for a source task*) to learn the target concept (target CNN, *learned for a target task*). Recent studies detail how transfer learning can be performed via CNNs

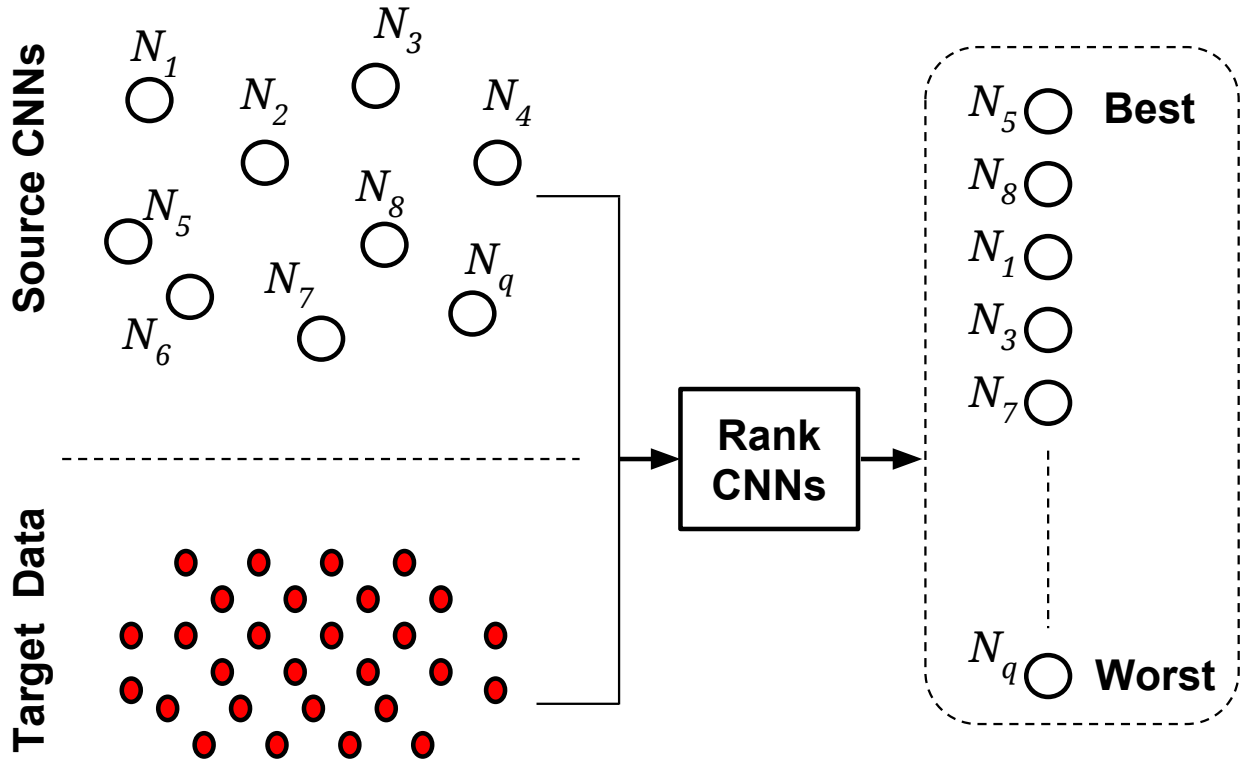

Figure 4.1: Given a large number of pre-trained source CNNs, the proposed approach ranks them in the order in which they are likely to impact the performance of a given target task. The source task data is *not* used in this determination.

by transplanting the learned feature layers from one CNN to initialize another [37] (See Fig. 4.2 and Fig. 4.3). Due to its significant impact on improving the performance of the target task, transfer learning is becoming a critical tool in many applications [38][39]. Usually this process is referred to as *fine-tuning* to indicate that the transplanted feature layers of a source CNN are merely refined using the target data. It is necessary to note that for such a transfer, the source *data* is not needed; only the source *concept* as embodied by the source CNN is required. This allows researchers to freely share and reuse previously learned CNN models.[1] Attempts to convert CNN models from one programming platform to another[2] has also facilitated the reusability of CNNs. Given these

[1] https://github.com/BVLC/caffe/wiki/Model-Zoo
[2] https://github.com/facebook/fb-caffe-exts

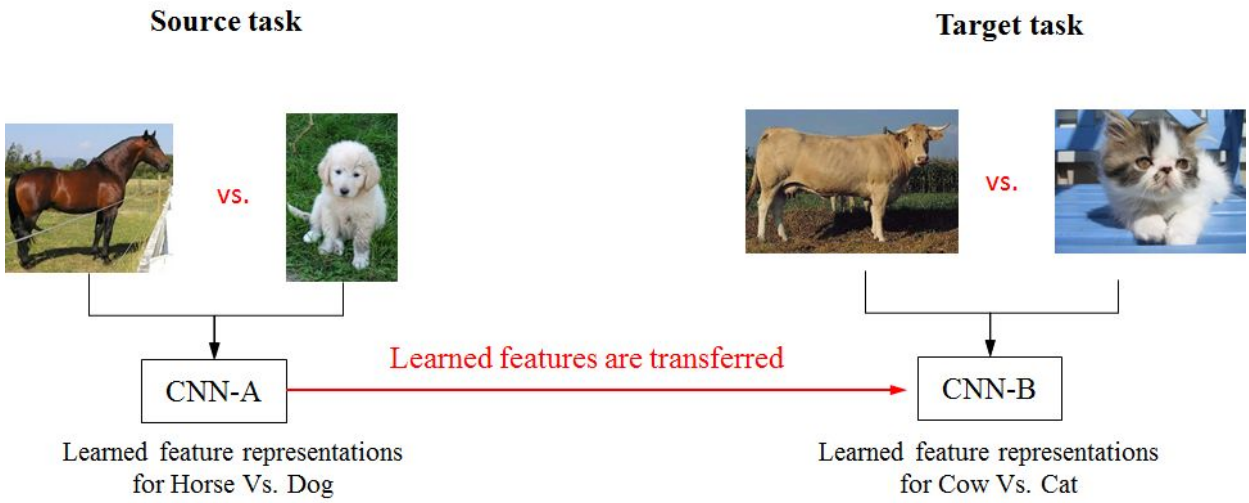

Figure 4.2: The availability of source task data is not necessary in CNN based transfer learning. Transfer learning may only require the source CNN model and the target data for tuning.

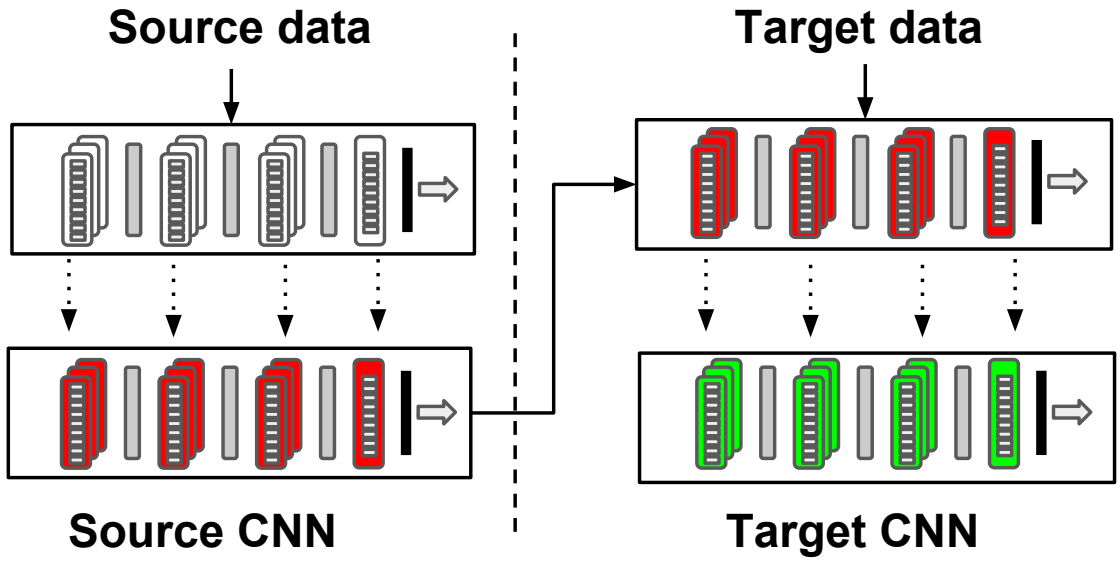

Figure 4.3: This figure demonstrate the basic process of knowledge transfer. Learned feature layers of a source CNN are transplanted to initialize a target CNN which is then tuned using the target data.

developments, it has become necessary to investigate how CNN models learned on various source tasks can be effectively used when learning a target task that has very limited training data.

Given a selected source task or a source CNN, recent studies show a number of useful ways to transfer and exploit its information for maximizing the performance gain on the target task [37][40][41][42][43].

Previous research has clearly demonstrated that the *choice* of the source CNN has an impact on the performance of the target task [37]. Some sources[3] may also result in a phenomenon called *negative transfer* where the performance on the target task is degraded as a result of transfer learning. However, a principled reason for such a degradation has not been clearly determined. Further, in CNN-based transfer learning, the source is *manually* chosen (e.g., [39, 38]). Several different approaches have been suggested to manually select a source for transfer learning. In [38], Agrawal et al. demonstrate that source data obtained from a moving vehicle [44] can be effective for transfer learning, thereby highlighting the importance of motion-based data. In [37], Yosinki et al. argue that source tasks that appear to be semantically relevant to the target task would result in better performance. A large number of studies, however, show that semantic relevance between source and target tasks is not always necessary; performance improvement has been observed even when the source and target tasks are superficially not related [38][45].

Manual selection has three major drawbacks: it is *subjective*, where multiple experts may choose a different source for the same target task; *unreliable*, where there is no guarantee that the chosen source will result in better performance than others; and *laborious*, where an expert has to manually analyze a very large number of potential sources tasks. Currently, there is no principled way to automatically select the best source CNN for a given target task.

4.1.2 Technical goal

The key technical goal of this study, therefore, is to investigate the possibility of automating source CNN selection. By choosing the best source CNN for a given target task, we anticipate that high performance can be achieved despite tuning with very limited target data. Since, this is the first study attempting to automate source CNN selection, we first present the following three ideal

[3]Note that in this study, *source* will be used as a general term referring to both source task and source CNN.

requirements of such a ranking measure:

- **Scalable:** It only utilizes source CNNs. It does not require us to additionally store and maintain the source data of each source task.

- **Efficient:** Unlike a standard learning based problem where an objective function is defined and optimized using a training dataset, the ideal ranking approach should perform a *zero-shot* ranking of CNNs, i.e. the ranking approach should not utilize a learning phase that is based on source CNN characteristics.

- **Reliable:** Ideally, the ranking measure should not be based on heuristics, especially those simply based on the notion of perceived similarity or difference between the tasks. The ranking measure should be theoretically sustained and not heavily dependent on the specific target task.

4.1.3 Novelty and contributions

- This study is the first to demonstrate that automatically ranking pre-trained source CNNs is possible.

- This study presents an information theoretic framework to rank source CNNs in an efficient, reliable, *zero-shot* manner thereby satisfying all the requirements stated above.

- This study presents a thorough experimental evaluation of the proposed theory using Places-MIT database, CalTech-256 database, MNIST database and a *real-world MRI database (which is the focus of this thesis).*

Table 4.1: A summary of related research in transfer learning via CNNs

Research	Focus	Source selection
Oqub et al. [46]	Transfer in CNNs by transplanting feature layers	Manual
Yosinki et al. [37]	Impact of transplanting different CNN layers	Manual
Long et al. [40]	CNN based transfer in deeper layers	Manual
Agrawal et al. [38]	Application of transfer learning via CNN	Manual
Tulsiani et al. [39]	Application of transfer learning via CNN	Manual
E. Littwin et al. [43]	Effect of a multiverse loss for improving transfer in CNNs	Manual
Proposed	Zero-shot ranking of source CNNs	*Automated*

4.1.4 Related work

4.1.4.1 Transfer learning via CNNs

Oqub et al. in [46] explained how transfer between CNNs can be implemented by transplanting network layers from one CNN to initialize another. This procedure provides significant improvement on the target task and has been utilized in different applications [38][39][45]. Yosinki et al. in [37] present an empirical understanding of the impact of transferring features learned in different CNN layers. They show that CNN features learned in the first layer are generic and similar across multiple tasks. These features become more and more task specific in the deeper layers. The authors also discuss the differential impact of source CNNs on the target task. Long et al. in [40] describe how deeper layers can be more effectively transferred to the target CNN. A recent study in [43] provides intuitions on the effect of a multiverse loss function in improving the performance of transfer learning in CNNs.

The goal of our work is significantly different from the above. In particular, we seek to develop a principled way for *automatically* ranking source CNNS based on their potential to favorably influence the performance of the target task. Given the increasing availability of source CNNs in the public domain and the diversity of practical applications that have to contend with scarcity of training data, the proposed approach is expected to have a significant impact on the viability of transfer learning.

4.1.4.2 Transfer learning in traditional research

As shown in Tab. 4.1, a number studies have been published in traditional transfer learning research that does not utilize CNNs. These studies adopt different approaches for transferring information across tasks. In the context of transfer learning, the meaning of several learning terminologies vary across different studies in the literature. In the following subsections, we present a brief discussion on these terminologies and differences.

4.1.4.3 Supervised transfer learning

Different studies refer to the term *supervised transfer learning* to represent slightly different contexts of learning. In many studies supervised transfer learning means the case where there is abundant labeled data for the source task but limited labeled data for the target task. Research by Daume [47] and Chattophadyay [48] use this terminology to mean this particular context. On the other hand, in studies such as those by Gong [49] and Blitzer [50] use a different term for this setup. They relate such a setup with a *semi-supervised transfer learning*. For Cook [51] and Feuz [52], the term supervised transfer learning only relates to the source task data. If the source task has any labeled data, they consider it as the case of supervised transfer learning. Further, they call the learning as *informed* or *uninformed* based on the availability or absence of labeled, target task data.

4.1.4.4 Semi-supervised transfer learning

The term *semi-supervised transfer learning* has also been used in the literature to represent different contexts. For example, studies by Daume [47] and Chattophadyay [48] use this term to refer to a case where there is abundant availability of labeled source task data but the target task data is not available. On the other hand, as mentioned before, Blitzer [50] and Gong [49] use the term

semi-supervised transfer learning to refer to a case where the labeled source task data is abundant and the labeled target task data is limited.

4.1.4.5 Unsupervised transfer learning

In the general context, when learning approach only utilizes unlabeled training data, the approach is referred to as unsupervised. In the context of transfer learning, different studies use the term *unsupervised transfer learning* to refer to slightly different contexts. For Fuez [52] and Cook [51] unsupervised transfer learning represents a case where the labeled source task data is not available. For Blitzer [50] and Gong [49], it means the case where there is abundant labeled data for the source task but no labeled data for the target task. Note that this scenario was referred as semi-supervised transfer learning by Chattophadyay [48] and Daume [47]. Further, for Pan [53], this term refers to the case where there is no labeled data for both, the source task and the target task.

4.1.4.6 Inductive and transductive transfer learning

Pan [53], use the term *inductive transfer* to refer to a scenario where some labeled data for the target task available. On the other hand, transductive transfer refer to a case where labeled target task data is not available, however, labeled source task data is present. Note that for this setup, Gong [49] and Blitzer [50] used the term of unsupervised learning.

Based on *what* is transferred, these approaches can be mainly categorized as (1) instance-based transfer learning, where the *labeled data* in the source task is re-weighted to be utilized for the target task [54, 55, 56, 57], (2) feature-based transfer learning, where the *features* of the source task are transformed to closely match those of the target task, or a common latent feature space is discovered [58, 59, 60], (3) parameter-based transfer learning, where the the goal is to discover *shared parameters* across tasks [61, 62] and (4) relational knowledge-based transfer learning, which is a

Table 4.2: A brief overview of transfer learning research

Paper	**Focus of research**
Dai et al. [54]	Transfer learning via boosting algorithm
Jiang et al. [55]	Source instance weighting for domain adaptation
Liao et al. [56]	Utilizing auxiliary data for target labeling
Wu et al. [57]	Integrating source task data in SVM learning framework
Pan et al. [58]	Transfer learning via dimensionality reduction
Pan et al. [59]	Domain adaptation using efficient feature transformation
Blitzer et al. [50]	extracting features to reduce difference between domains
Dai et al. [64]	Labeling target task data using unlabeled source task data
Duame et al. [47]	Domain adaptation using feature augmentation
Xing et al. [65]	Correcting the predicted labels of shift-unaware classifier
Rosenstein et al. [66]	Negative transfer between tasks
Pan et al. [67]	Spectral feature alignment for transfer learning
Raina et al. [60]	Learning high-level features for transfer learning
Gong et al. [49]	Reducing domain difference in a low dimensional feature space
Tommasi et al. [61]	Transferring SVM hyperplane information
Yao et al. [62]	Transferring internal learner parameter information
Mihalkova et al. [63]	Markov logic networks for transferring relational knowledge
Long et al. [68]	Joint domain adaptation
Ammar et al. [3]	Automated source selection in reinforcement learning using RBMs

comparatively less explored area in this context, and where the goal is to transfer the *relationship* among data from a source task to a target task [63].

While the history of transfer learning research spans over two decades [69, 70, 53], the question of how to predict the transferability of a source task, in a supervised framework, is relatively less studied. Some studies assumed that the source and target tasks had to be similar in order for the transfer learning to be effective [66][54]. Such an assumption may not be true in practice. For example, if the target task itself is duplicated and presented as a source task, the similarity between target and source tasks would be maximal; however, such an arrangement will be undesirable due to redundancy and over-fitting. In addition, such approaches may necessitate the storing of source data. In [71], a method to choose auxiliary training data to facilitate transfer learning is discussed. The method utilizes a validation set based on the target task in order to select the auxiliary training

samples. However, the method is iterative, computationally expensive, and does not utilize the auxiliary data in a zero-shot manner. Recently, in [3], the authors utilized a *restricted boltzman machine* based approach to automatically select the source task for transfer in the specific context of *reinforcement learning*. However, the approach has two distinct shortcomings. Firstly, it is based on the implicit assumption that the source and target data have to be visually similar in order for the transfer learning to be effective. Secondly, the approach does not explicitly link the ranking criteria with performance gain on the target task. However, the approach is observed to perform well on the target tasks considered by the authors. Therefore, we compare the proposed approach with the approach in [3].

4.2 Approach

In this section, before we present the proposed theoretical framework, two intuitive and preliminary studies are discussed first. The detailed experimental results from these two intuitive approaches will be presented later in the supplementary material. In the following subsections, only the motivation for these approaches and their limitations are discussed. This will be followed by a detail discussion on the proposed theoretical framework that meets all the design requirements mentioned previously.

4.2.1 Intuitive approach: A solution space based approach

The approach here is based on the hypothesis that given a CNN architecture, there exist a solution space for it. Different points in this space represent CNN based solutions for different applications. This indicates that there may be a spatial region representing ideal solution(s). Thus, the data is merely utilized by a CNN during its training phase to traverse this space so as it moves away

from a randomly initialized spatial location. As the CNNs utilize training data for a specific task, it traverses far from the random, non-ideal space and hence learns more useful features. In this regard, the following concepts are more formally presented:

4.2.1.1 CNN solution space

Consider a high dimensional solution space with each point denoting the weights in the layers of a CNN that are transferred. For a fixed CNN architecture, each point in this space denotes a CNN based solution for some task. For example, one point in this space may represent an *ideal* solution for the face-recognition problem whereas another may represent a *non-ideal* solution for disease estimation. It has been generally accepted in the literature that the weights in the initial layers of a CNN act as *general* feature extraction operators and, thus, the CNNs for many different tasks may have similar first layers; in contrast, the weights that are in the deeper layers of the network become increasingly *task specific* [37, 40]. Hence, the weights in the deepest convolutional layers represent the most task specific weights and are utilized here to denote a CNN.

4.2.1.2 Solution difference

This measures the Euclidean distance between two points in the solution space. Solution difference between two CNNs N_i and N_j is computed as $\rho_{ij} = ||W_i - W_j||_2^2$. Here W_i and W_j are two points in solution space denoting N_i and N_j respectively.

4.2.1.3 Solution path

During training, a CNN is first initialized to a point (random or otherwise) in the solution space. Then the learning algorithm adjusts the weights incrementally after each epoch, and the updated weights traverse a path in the solution space, referred to here as the *solution path*. For a task i, let

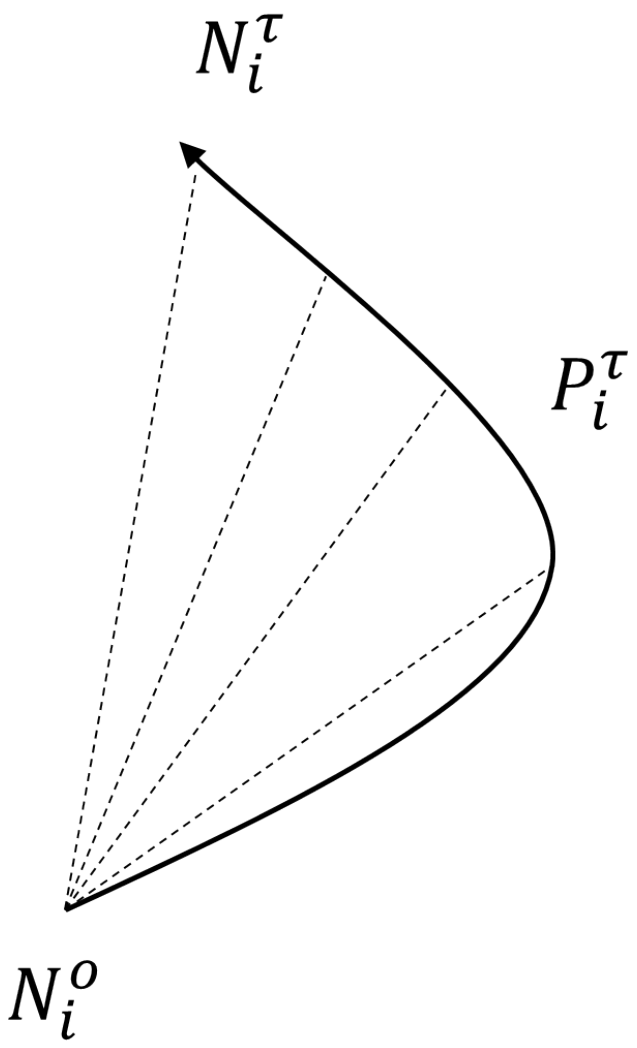

Figure 4.4: The output of the last layer m is task dependent and is therefore its dimensionality is different depending on task. Hence, the k-dimensional output of layer $m-1$ is utilized. Also, it is the output of this layer which will be utilized later in the experiment section for visualization.

$P_i^\tau = [N_i^o, N_i^1, ..., N_i^\tau]$ denote its solution path, where N_τ^i denotes the solution point at epoch τ and N_o^i represents the initialization point.

4.2.1.4 Path-to-point profile

A sequence of differences between a CNN N_j and each point in P_i can be computed using the solution difference measure above. This results in a sequence of differences $\eta_{ij}^\tau = [\rho_{ij}^o, \rho_{ij}^1, ..., \rho_{ij}^\tau]$,

where ρ_{ij}^{o} is the solution difference between N_i^o and N_j.

4.2.1.5 Source CNN ranking

Given a source CNN N_i^τ and a randomly initialized CNN N_o, the ranking score can be computed as,

$$E_i = \rho_{io}^{\tau}. \tag{4.1}$$

Now, the training process results in τ intermediate CNNs: i.e., the CNNs in P_i^τ. Any of these τ CNNs could potentially be a suitable candidate for transfer learning; it is not necessarily the case that N_i^τ will result in the best performance gain after transfer on the target task. Therefore, the criterion E_i is updated as,

$$E_i = max\{\eta_{ij}^{\tau}\}. \tag{4.2}$$

Note that the development of E_i relies on the fact that small training datasets, in general, are incapable of imputing a comprehensive representational power to a CNN. As the *supplementary information* shows this approach can work well when the target training data is limited (see Chap. 6 for experimental results). However, this approach also has two distinct drawbacks:

- The presented approach is intuitive and is not derived from theory.
- CNN ranking does not take into consideration the variability in the target task.

Considering these limitations, a theoretical approach is presented next which meets all the aforementioned ideal requirements. In Supplementary material (Sec. 6.2) experimental results of this approach are presented.

Table 4.3: Summary of the basic notations used in this section

Notation	Description
$D_i = (X_i, Y_i)$	Training dataset for *source* task i.
$D_e = (X_e, Y_e)$	Test dataset for target task.
$D_t = (X_t, Y_t)$	Training dataset for target task.
$D_v = (X_v, Y_v)$	Validation dataset for target task.
X_i	Training samples (images) for source task i.
Y_i	Corresponding ground truth labels on X_i.
X_t, X_v, X_e	Training, Validation, and Test samples for target task, respectively.
Y_t, Y_v, Y_e	Corresponding ground truth labels on X_t, X_v, and X_e respectively.
N_i	Source CNN learned using data D_i.
N_t	Target CNN learned using data D_t.
m	Total number of processing layers in a CNN.
l	Denotes the *layer number* in a CNN, where $l \leq m$.
$H(A)$	Entropy of variable A.
$H(A\|B)$	Conditional entropy of A given variable B.
$I(A;B)$	Mutual information between A and B.

4.2.2 Theoretical approach

4.2.2.1 Notations

Consider a set of q source tasks with corresponding training datasets $\{D_1, D_2, ..., D_q\}$. For each task $i \leq q$, dataset $D_i = (X_i, Y_i)$ where X_i represents the training samples and Y_i denotes the corresponding labels on them. Also, for each task i, a CNN N_i is learned by utilizing D_i for training. This results in a set of q source CNNs $\{N_1, N_2, ..., N_q\}$.

Similarly, consider a target dataset that is divided into D_e and D_a. The test set is represented as D_e while $D_a = \{D_t, D_v\}$ represents the data that can be utilized for training (D_t) and validation (D_v). Note that the sizes of the training and validation data will be kept very small in our experiments in order to assess efficacy of the proposed approach in real-world applications with small training data.

Similar to the source task datasets, each of the datasets corresponding to the target task also comprises of images and corresponding labels. For example, the target training set can be denoted

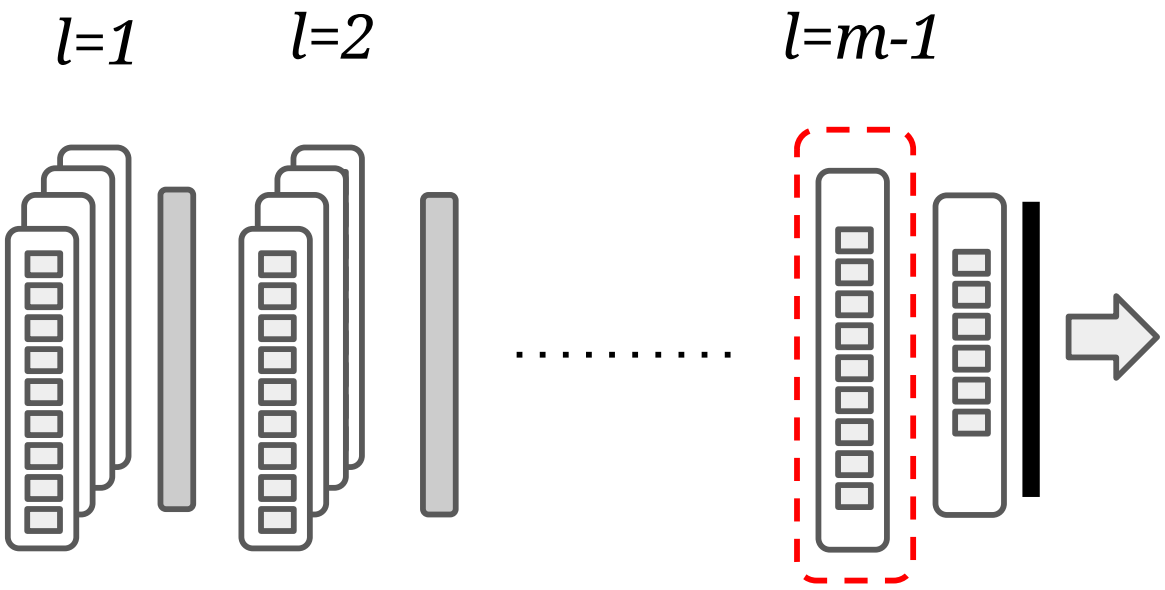

Figure 4.5: The output of the last layer m is task dependent and is therefore its dimensionality is different depending on task. Hence, the k-dimensional output of layer $m-1$ is utilized. Also, it is the output of this layer which will be utilized later in the experiment section for visualization.

as $D_t = (X_t, Y_t)$, where X_t are the images and Y_t are the corresponding labels. Further, let N_t denote the CNN that is learned using the small training set D_t. A brief summary of the notations is tabulated in Tab. 4.3.

4.2.2.2 Deriving the measure

The goal here is to derive a ranking measure on source CNNs that is explicitly based on reducing the error on the target task. The uncertainty in predicting the testing labels Y_e is given by the entropy $H(Y_e)$, where $H()$ represents the entropy function. A higher entropy value would mean a larger uncertainty in prediction and, therefore, the goal is to reduce $H(Y_e)$. With the availability of more information, which can be potentially useful in label prediction, this uncertainty can decrease. Given that we have a trained CNN N_t that was derived using the small training data D_t, additional information $N_t^m(X_e)$ can, in principle, be extracted from the test images X_e. The notation $N_t^m(X_e)$

indicates that images in X_e are input into the CNN N_t and the output of the m^{th} layer is obtained. Here, m is the last layer (total depth) of the CNN. Since, $N_t^m(X_e)$ represents the final output score by the last layer of N_t on X_e, we write $N_t^m(X_e)$ as $N_t(X_e)$ for simplicity.

Theoretically, as conditioning reduces entropy, therefore,

$$H(Y_e) \geq H(Y_e|N_t(X_e)). \tag{4.3}$$

Similarly, additional information can also be extracted from X_e by utilizing the feature representations learned by the CNN for a source task i. This information can be denoted as $N_i^l(X_e)$. Since, the dimensionality of the output of last layer, i.e., at $l = m$, can be different for different source tasks[4], the output of the layer $l = m-1$ is extracted and utilized. Again, as conditioning reduces the entropy, we have,

$$H(Y_e) \geq H(Y_e|N_t(X_e)) \geq H(Y_e|N_t(X_e), N_i^{m-1}(X_e)). \tag{4.4}$$

Further, as the test images X_e and the labels Y_e will not be available during the training stage, the validation data $D_v = (X_v, Y_v)$ is utilized instead. Thus,

$$H(Y_v) \geq H(Y_v|N_t(X_v)) \geq H(Y_v|N_t(X_v), N_i^{m-1}(X_v)). \tag{4.5}$$

This equation shows that with additional information extracted using a source CNN, the uncertainty in prediction can further decrease. Now, the total decrease in uncertainty can be written as

[4]Here, the dimensionality pertains to the number of classes in a task

the difference between the following terms:

$$\phi = H(Y_v) - H(Y_v|N_t(X_v), N_i^{m-1}(X_v)). \tag{4.6}$$

In information theory, this difference ϕ is called gain or information gain. This gain can also be rewritten in the form of mutual information as,

$$\phi = H(Y_v) - [H(Y_v|N_t(X_v)) - I(N_i^{m-1}(X_v); Y_v|N_t(X_v))]. \tag{4.7}$$

Here, the mutual information is denoted by the function $I()$. For any three variables A, B and C; $I(A,B|C) = I(B,A|C)$ and so:

$$\phi = H(Y_v) - H(Y_v|N_t(X_v)) + I(Y_v; N_i^{m-1}(X_v)|N_t(X_v)). \tag{4.8}$$

In the context of two variables, $H(Y_v) - H(Y_v|N_t(X_v)) = I(Y_v; N_t(X_v))$ hence:

$$\phi = I(Y_v; N_t(X_v)) + I(Y_v; N_i^{m-1}(X_v)|N_t(X_v)). \tag{4.9}$$

The final equation here has two terms. The first term $I(Y_v; N_t(X_v))$ denotes the gain due to the mutual information between the target labels Y_v and the predicted output scores $N_t(X_v)$ by the target CNN. The higher this mutual information, the lesser the uncertainty in predicting Y_v. Fig. 4.6 shows an information diagram[5] for the aforementioned terms. Region-1 in this figure represents the first term of Eqn. (7).

Note that the second term, $I(Y_v; N_i^{m-1}(X_v)|N_t(X_v))$, represents the gain due to a specific source,

[5]An information diagram is similar to a venn diagram but is used to show relationship between Shannon's basic measures of information.

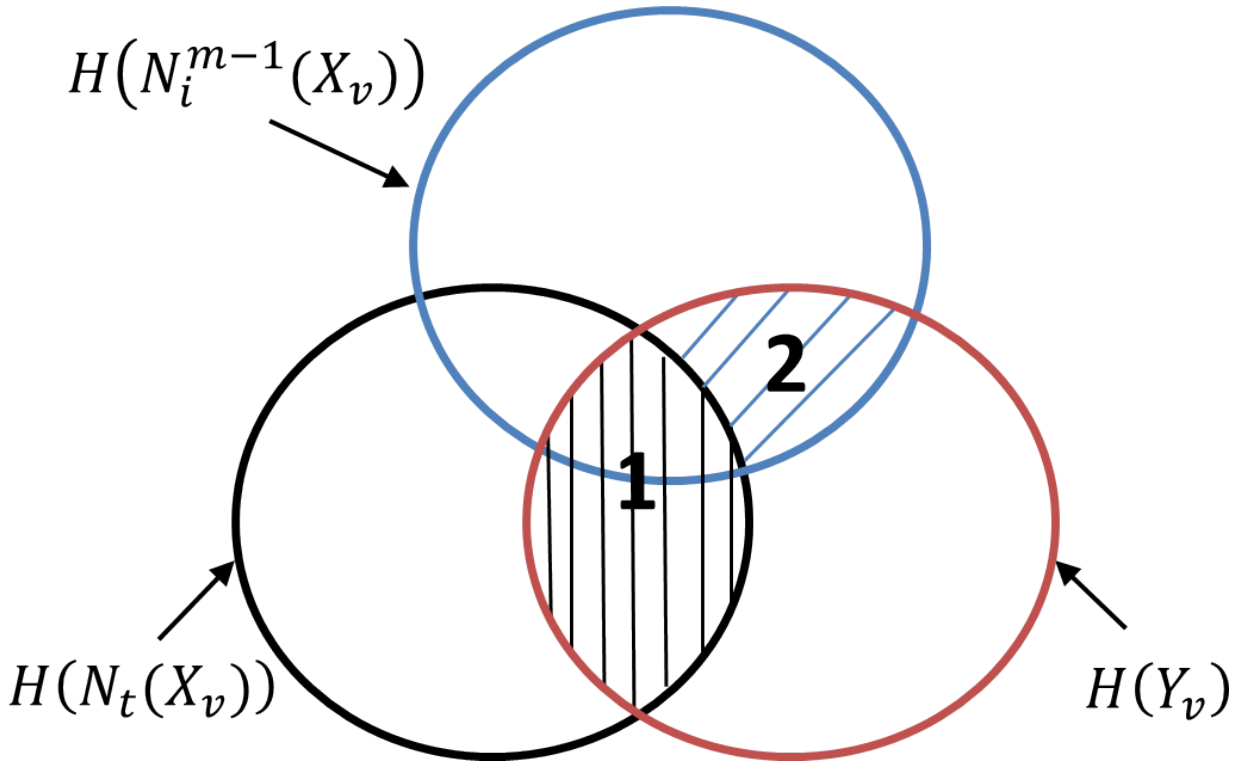

Figure 4.6: Information diagram: The first term in Eqn. (7) is represented by region-1 whereas the second term is represented by region-2. The larger the region-2, the more useful is the source CNN.

that is *not already accounted* for by $N_t(X_v)$. In Fig. 4.6, region-2 represents this term. This term provides additional, *relevant* information that was not available when only utilizing the target's training data D_t. The higher the value of this term, the more useful a source will be. Since, the first term is independent of the source, the second term here can be utilized to measure the worth of a source CNN. Therefore, for a source CNN N_i, its ***transferability***[6] γ^i is given as,

$$\gamma^i = I(Y_v; N_i^{m-1}(X_v) | N_t(X_v)). \tag{4.10}$$

Note that this term can easily be computed using publicly available implementations for mutual information. For the reproducibility of the results, the implementation and datasets used here will be made publicly available.

[6]In this study, the term *transferability* and *ranking score* will be used alternatively.

4.2.2.3 Discussion

The proposed term is aware that if the information extracted via source CNN is exactly that of the target CNN, i.e., $N_i^{m-1}(X_v) = N_t(X_v)$, the transferability will be zero. By computing the conditional mutual information, the term is explicitly evaluating a source CNN, N_i, based on the additional, predictive information between Y_v and $N_i^{m-1}(X_v)$ that cannot be obtained from $N_t(X_v)$. If the source CNN is completely irrelevant, the additional, predictive information from $N_i^{m-1}(X_v)$ will have zero mutual information with Y_v, resulting in zero transferability.

4.2.2.4 Upper bound on transferability

The upper-bound on transferability can also be estimated. This estimate will denote the maximum transferability that can be achieved by a source CNN. The total uncertainty in predicting labels is estimated by $H(Y_v)$. Some predictive information is provided by N_t that is trained on the target's training data. This information is denoted by region-1 in Fig. 4.6. This overlap of information can be written as $H(Y_v) \cap H(N_t(X_v))$ or simply as the mutual information $I((Y_v); N_t(X_v))$, as discussed before. The remaining information, $H(Y_v) - I((Y_v); N_t(X_v))$, can be provided by a source CNN. Theoretically, this estimates the maximum amount of information that is required. Since, $H(Y_v) - I((Y_v); N_t(X_v)) = H(Y_v|N_t(X_v))$, the upper bound on transferability, γ^{max}, can simply be written as $H(Y_v|N_t(X_v))$.

4.2.3 Datasets

4.2.3.1 Target Data - MRI database

A real world MRI dataset [18][72] is utilized as the target data. The task is to detect the injected cells in *in vivo* MRI scans that appear as dark *spots*. In this thesis, this data is denoted by set G_A. In many medical applications such as this, not only is the collection of data challenging but the labeling of the data is also expensive and highly time consuming. For the long-term success of cell based therapies, it is essential that in such applications, injected cells are detected accurately with *minimum labeling input* which is currently a practical challenge [72][73].

This dataset comprises of 5 MRI scans of different *in vivo* rat brains. Spots in 3 of these scans were labeled by a medical expert. These 3 scans were utilized in this study. From each scan about $100,000$ patches were extracted as potential spots by authors in [72]. Only about $5,000$ of these were spot-patches (positive class) and the remaining were non-spot patches (negative class). Train and test scans were mutually exclusive. From each training scan, only 5% of the patches (about $5,000$) were randomly selected and utilized. Further, only 85% of the selected 5% were used for training N_t and the remaining 15% was used as the validation set D_v.

4.2.3.2 Target Data - MNIST database

In a separate experiment, we test the generalization of the proposed approach using the standard MNIST database. Here, the multi-class task involves differentiating between written digits ranging from “0" and “9". The total number of training samples in MNIST database is about $60,000$, out of which *only* 5% are randomly chosen and utilized in the same manner.

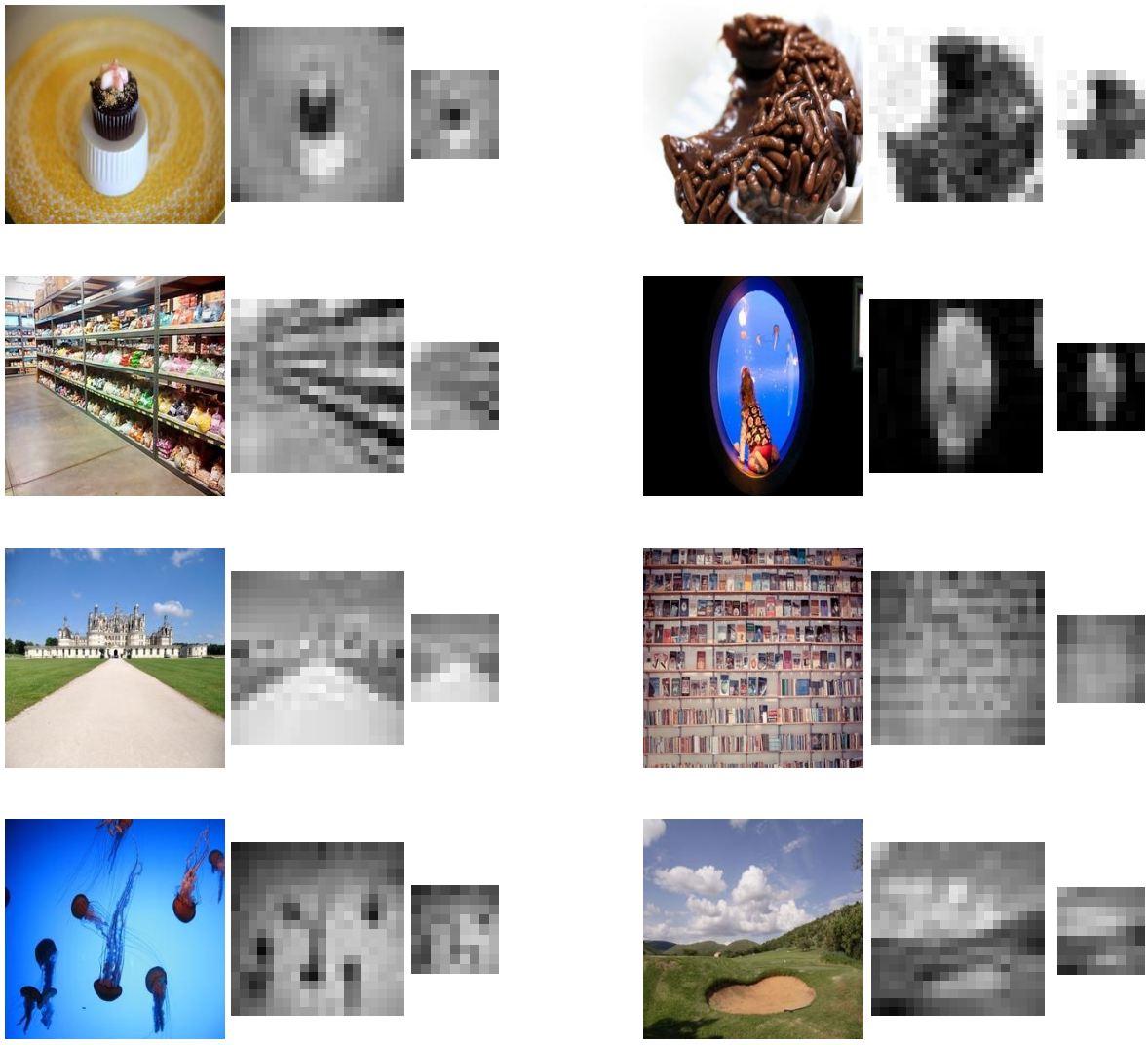

Figure 4.7: Each image in the source dataset was converted to gray scale and then down-sampled to 20×20 and 9×9. Some of these images along with their transformed versions are shown here.

4.2.3.3 Source Data - Places-MIT database

In this study, the publicly available Places-MIT dataset was utilized[74]. This dataset has a diverse set of 205 different classes with images containing cluttered urban scenes, empty hall-ways, cakes (in bakery), fish (in aquarium), etc. A set of 500 different tasks were randomly generated with classes ranging from 2 to 205. The images in this database are much different in dimensions from the 9×9 patches in the MRI database and the 20×20 images of the MNIST database. Therefore, each image here was converted to gray scale and then down-sampled to the size compatible with

the images of the two target tasks. The transformed images exhibit diversity in their content, as shown in Fig. 4.7.

4.3 Experiments, Results and Discussion

In this section, we design experiments to answer the following questions: (1) How well does the proposed measure rank the source CNNs for a target task that has scarcity of training data? (2) How does the performance of the proposed approach compare with a previous approach in the literature that is heuristic-based? (3) Can the impact of the top and the worst ranked source on the target task be visualized and compared? (4) How does the number of training samples impact the performance gain due to transfer learning in CNNs? In all experiments, AUC (Area Under ROC) was utilized as the measure of accuracy.

4.3.1 MRI based target task

Ranking Source CNNs: Using the 500 source tasks generated from Places-MIT database, 500 CNNs were learned. The CNN architecture used in [72][73] was adopted for this target task. Using the proposed approach, all these source CNNs were ranked prior to conducting the transfer. The ranking scores for each CNN, i.e., measured transferability, is shown on the horizontal axis of Fig. 4.8 while the performance with transfer learning is presented on the vertical axis. For vertical axis in Fig. 4.8, 500 more CNNs were learned by tuning each source CNN on the target training data. **Note the high degree of correlation between the ranking score and the degree of improvement in performance after transfer learning.** The scores shown here are the normalized scores obtained after dividing the rank score of each source CNN by the maximum score achieved by any of the 500 source CNNs. The two sub-figures in Fig. 4.8 represent the results on two

different test MRI scans. In each case 500 CNNs were evaluated on the complete set of test patches (about $100,000$ in each scan).

When training using the source data, each source CNN underwent a pre-determined number of 15 epochs. When tuning on the target data, the training of each source CNN proceeded until convergence.

Performance Comparison: In this experiment, the goal is to compare the performance of the proposed approach with another approach in the literature that merely relies on similarity between the source and target tasks. In [3] the authors propose utilizing RBMs for automated source selection which is based on the similarity between tasks. Therefore, using their proposed protocol, an RBM model was first trained for each source task. Then, using each source RBM model, the reconstruction error on the target data is computed. The normalized reconstruction errors for each source RBM model is shown on the horizontal axis in Fig. 4.8. The vertical axis represents the performance transfer learning using the corresponding source CNN. **It can be clearly observed that there is a lack of correlation between reconstruction error and performance improvement after transfer learning.** As mentioned before, the approaches based on *heuristics* of similarity or difference can fail in practice and may not be applicable for all source/target tasks.

Analyzing Ranked Source CNNs: The goal here is to visually investigate the difference between source CNNs that were ranked the best and the worst. To both these source CNNs, two different test sets were given as inputs and the 200-dimensional output of the fully connected layer the samples was obtained for all test samples. Note that these outputs are from source CNNs that have not yet been tuned using any target data. The 200 dimensional outputs were then projected to a 3D space using principal component analysis. The spot samples and the non-spot samples were

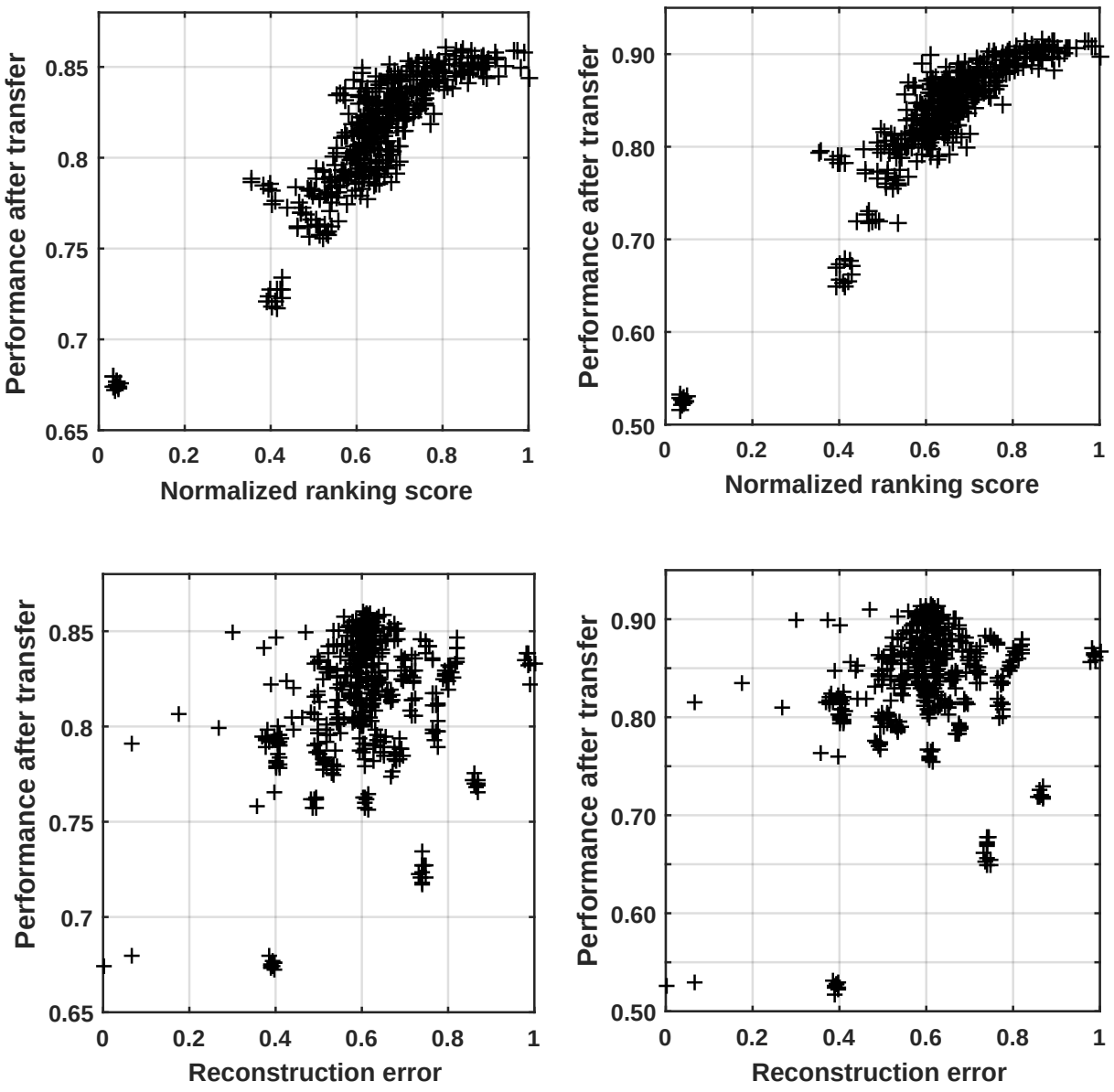

Figure 4.8: (Top row) The two sub-figures show the result of the proposed approach on two different test sets. (Bottom Row) These two sub-figures show the result of the Restricted Boltzmann Machine based approach in [3]. The horizontal axis shows the reconstruction error computed on the target's training data using the source RBM model. Note the high degree of correlation exhibited by the proposed measure (top row) with improvement in performance.

colored differently and visualized in this space (see Fig 4.9). In each figure, the viewpoint that best illustrates the decision boundary is presented.

It can be seen that the best ranked source, even prior to observing any MRI data, has the po-

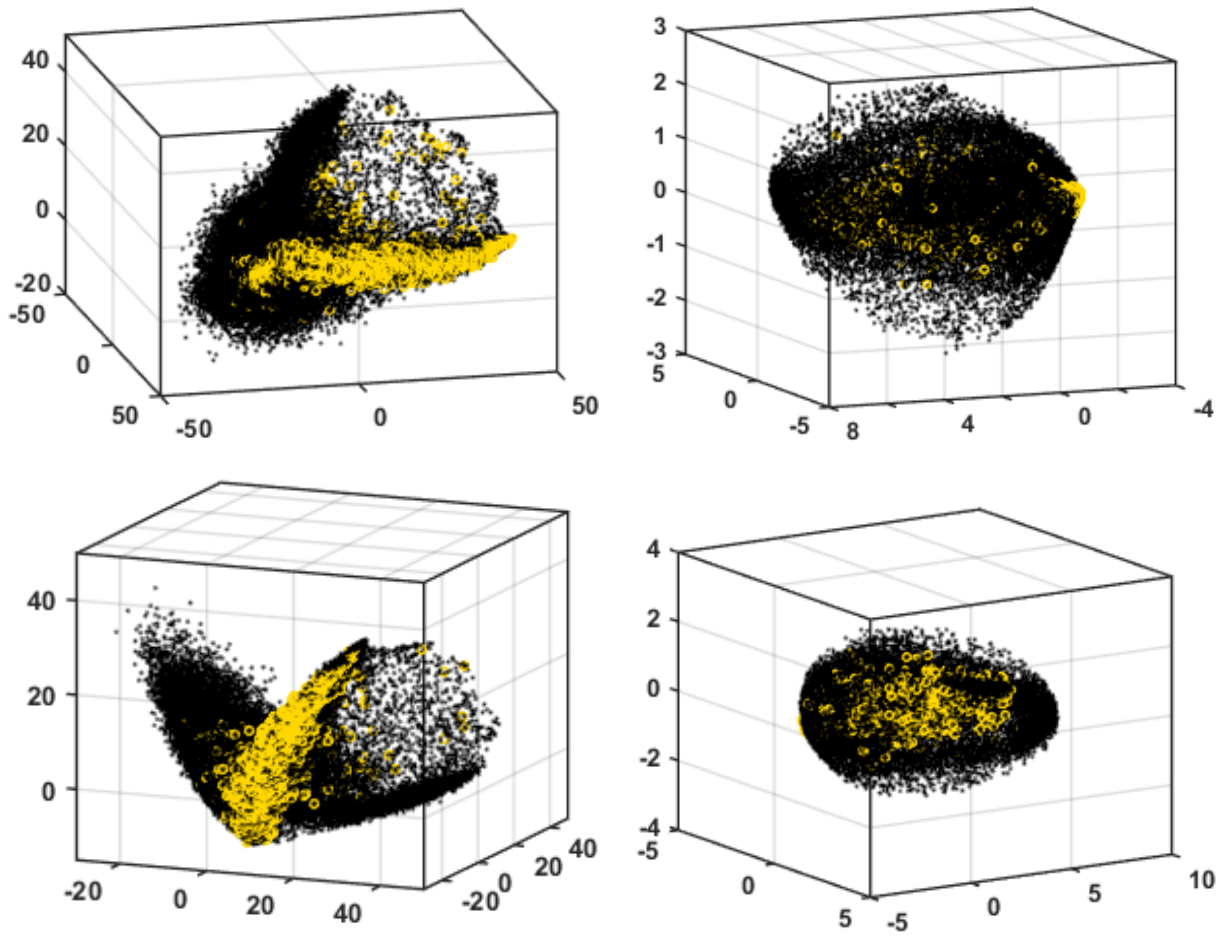

Figure 4.9: Prior to conducting any transfer, the ability to discriminate between spot and non-spot patches is visualized in 3D space for the best ranked and the worst ranked CNN. The figures in the left column correspond to the best ranked CNN on two test sets, while the figures in the right column correspond to the worst ranked CNN on two test sets. See text for further explanation.

tential to separate spot samples (*yellow*) from non-spot samples (*black*). The worst ranked source does not differentiate between the two classes. In fact, the spread of samples across the three dimensions is very low and all samples appear to be concentrated in a smaller region. Therefore, taking the best CNN as the initial point in learning the target concept clearly provides an edge over random initialization or using other source CNNs with much lesser ranking scores.

Impact of training sample size: Although, the main focus of this study is to test the efficacy

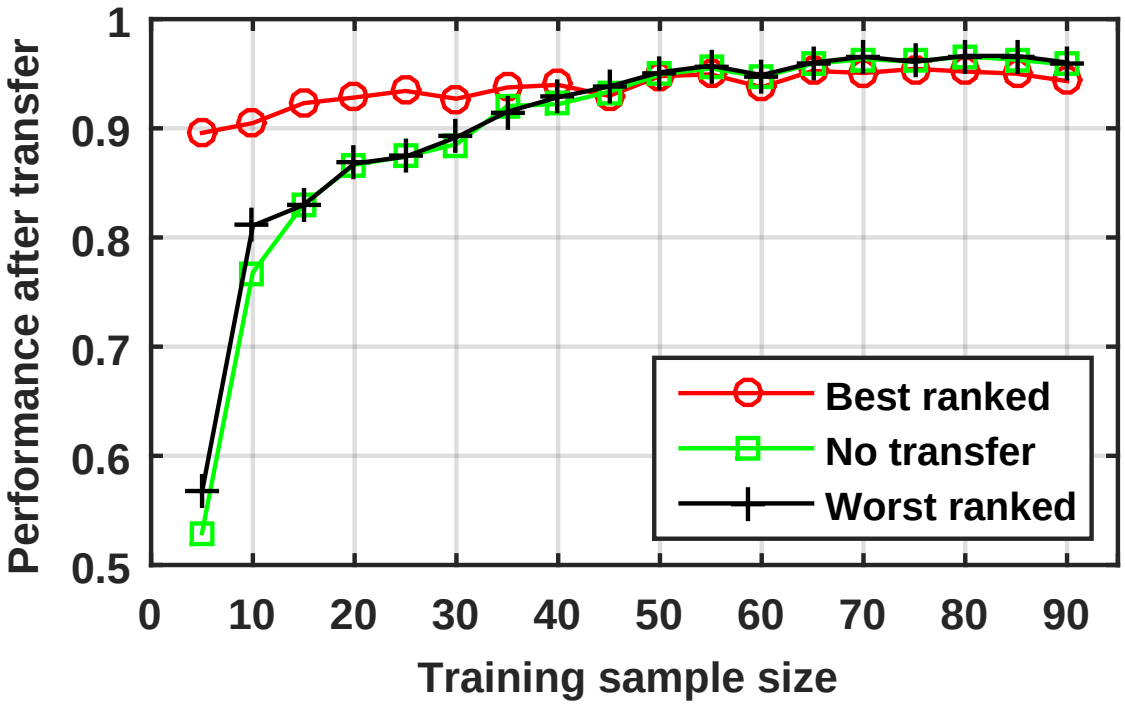

Figure 4.10: The horizontal axis show the *percentage* of the target training data utilized in tuning. The y-axis shows the performance after transfer. Dataset size was incremented in values of 5%, and for each dataset, the proposed approach was used to rank the source CNNs. Here, the transfer was only conducted using the best and the worst ranked CNN. Note the performance improvement for smaller training sizes which conveys the importance of the proposed method.

of the proposed approach when the target training data is very small, we are also interested in investigating the effect of increasing the target training size. In Fig. 4.10, we see that the proposed approach is especially very useful when the target training sizes are small. *Using only* 5% *of the available training data, the performance is observed to improve by more than* 35% *after transfer learning.* This means that the labeling effort from a medical expert can be significantly reduced without compromising the AUC performance. However, when there is already a large amount of training data available, transfer of knowledge from a source CNN may not bring a significant change in the results.

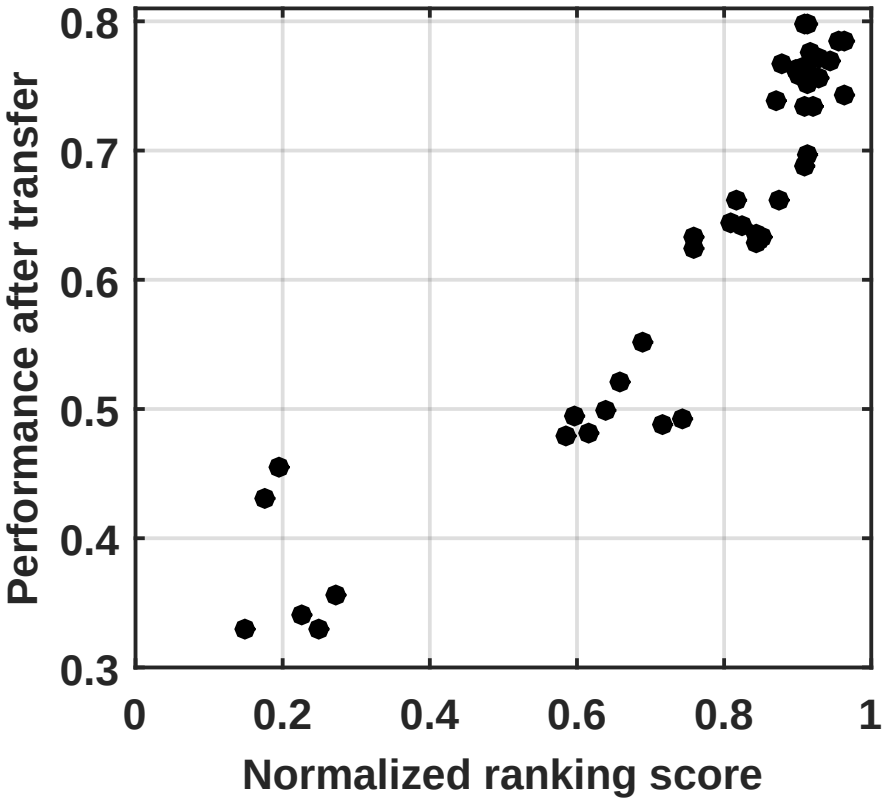

Figure 4.11: Ranking performance for the MNIST target task: 44 CNNs, based on randomly chosen source tasks, are ranked. For tuning, only 5% of the available training set was randomly chosen for the given task. Testing was performed on all the images in the MNIST test set. Note that the performance without transfer learning, using the selected 5% of the training data, was about 0.21.

4.3.2 MNIST based target task

We further evaluate the proposed ranking measure using the multi-class MNIST database. The experimental protocol used here is the same as the one used in the previous target task. A standard LeNet-like CNN architecture with ReLU activation layers was utilized. However, only 44 different source tasks were randomly picked and the corresponding CNNs were learned. Note that similar to the previous target task, only 5% of the available data was utilized, as explained in 4.2.3.2. The experimental results are shown in Fig. 4.11.

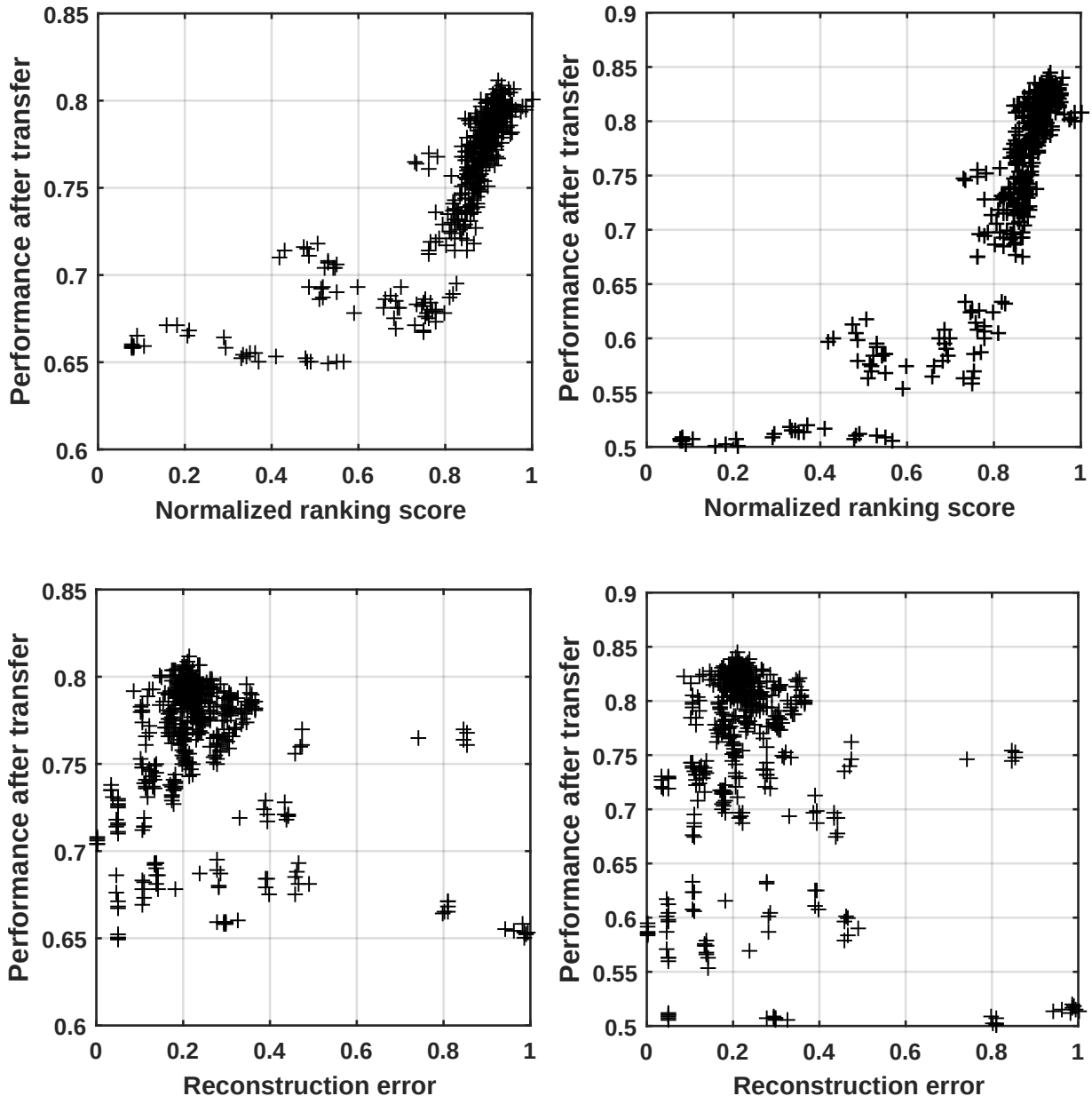

Figure 4.12: (Top row) The two sub-figures show the result of the proposed approach on two different test sets. (Bottom Row) These two sub-figures show the result of the Restricted Bolzmann Machine based approach in [3]. The horizontal axis shows the reconstruction error computed on the target's training data using the source RBM model.

4.3.3 Experiments using CalTech-256

Generally, in the literature on transfer learning, the target task is assumed to contain limited training data while the source task is assumed to have a large amount of training data. In this experiment,

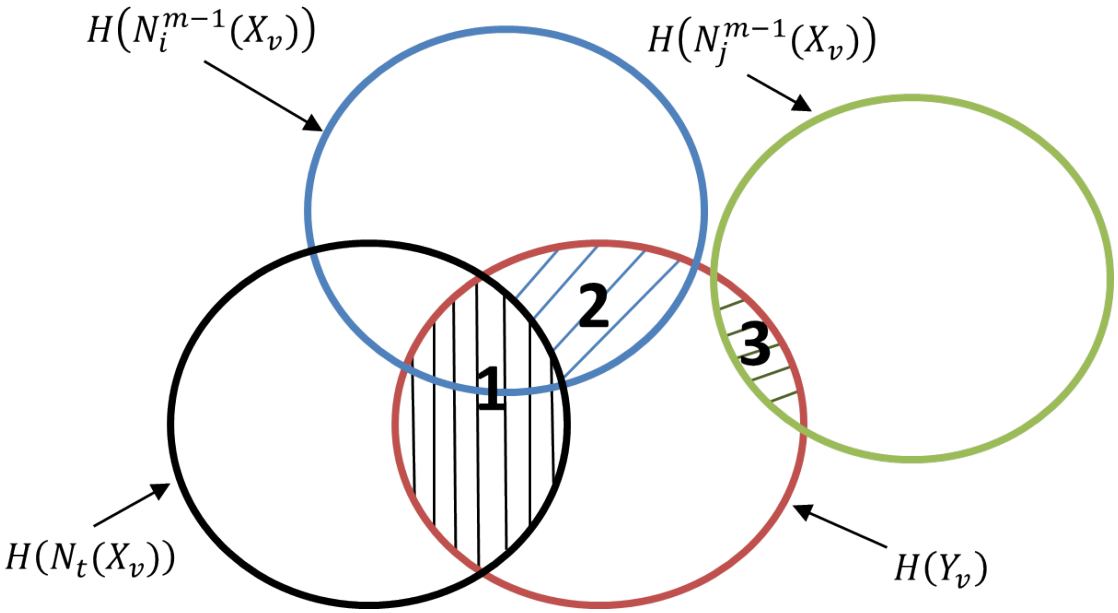

Figure 4.13: Information diagram: Emphasizing the need to exploit multiple sources.

a challenging, *non-conventional* case is considered to further test the robustness of the ranking measure. Here, the source CNNs are also trained using limited training data. Further, the training data for each class has large intra-class variations. To facilitate this, 500 additional source tasks were randomly generated using the publicly available CalTech-256 dataset. This dataset contains about 256 classes and the average number of images in each class is about 120. Classes in the additional source tasks ranged from 2 to 256. The problem of spot detection in MRI, as discussed in 4.2.3.1, was used for the target task. Despite a non-traditional scenario, we see in Fig. 4.12 that the approach is still able to differentiate between the sources when tested on two different MRI test sets. However, the variance at higher ranking scores is larger, indicating the challenge posed by ranking such CNN models.

4.4 Conclusion

This study is the first to show that the source CNNs can be ranked in increasing order of benefit for a given target task. An information theoretic framework that performs reliable, zero-shot ranking of CNNs was presented. The approach was thoroughly evaluated using Places-MIT database, CalTech-256 database, MNIST datbase, and a *real world MRI database (which is the focus of this thesis)*. We demonstrated that due transferring knowledge from the best source CNN, high performance can be achieved on the target task despite using small training data. Automating the crucial step of source selection is a fundamental improvement in the standard practice of transfer learning in CNNs. This study, also open doors to better investigate several other related research problems such as automatically finding the optimal numbers of layers to transfer. More details on potential future work are as follows:

4.4.1 Multiple sources

Using the proposed framework, in Fig. 4.13, we show an information diagram where another source CNN N_j brings information that is not accounted for by both N_t and N_i. Since, Region-3 is much smaller in comparison to Region-2, such a source should have a low rank score and is anticipated to be less beneficial compared to N_i. However, if N_j is utilized appropriately in combination with other sources such as N_i, the overall entropy will further reduce as $H(Y_v|N_t(X_v), N_i^{m-1}(X_v)) \geq H(Y_v|N_t(X_v), N_i^{m-1}(X_v), N_j^{m-1}(X_v))$ Therefore, one interesting future direction would be to extend the current framework to incorporate multiple source CNNs.

For example, the formulation presented here can also be seen as simplifying the problem of source CNN selection to a feature selection exercise. In this context, it will also be interesting to investigate how different feature selection approaches can be appropriated and experimentally

compared in this context.

4.4.2 Layers to transfer

The goal of this study was not to find the optimal *number* of layers to transfer, rather all the convolutional layers were transferred here. Finding an optimal number of layers to transfer, in a principled manner, is still an open problem. In future, we plan to investigate how the performance due to different number of transfer layers is *correlated* with the ranking score of a given source CNN.

Chapter 5

Exploiting Labeling latency

5.1 Introduction

In this chapter, we investigate the role of incorporating an expert's *labeling behavior* into a particular classifier, the convolutional neural network (CNN). The inspiration for this approach comes from research in psychophysiology where it has been observed that the human mind processes different images differently, based on the salient characteristics of individual images/stimuli [75][76]. This is perhaps the case when a medical expert carefully analyzes and labels spots in MRI scans. For example, an easy-to-classify spot may take less time to label, while a difficult-to-locate spot may require more time to label. Thus, the time taken by an expert to label each spot, i.e., the *labeling latency*, can be viewed as a variable that models the labeling behavior of an expert. However, the labeling latency value associated with a spot (positive sample) provides additional information that is *only available during training* and is absent during testing. Further, a medical expert only labels the positive samples in an MRI (the remaining samples in the MRI are automatically assumed to be negative samples). Thus, labeling latency is only available for one class, i.e., the positive class.

The paradigm of learning using privileged information (LUPI) is closely related to the problem at hand. Privileged information (also known as *side* or *hidden* information) is also available only during training but absent during testing [77, 78, 79, 80, 81, 82, 83, 84]. However, existing LUPI approaches cannot be appropriated in the context of supervised classifier learning where the side

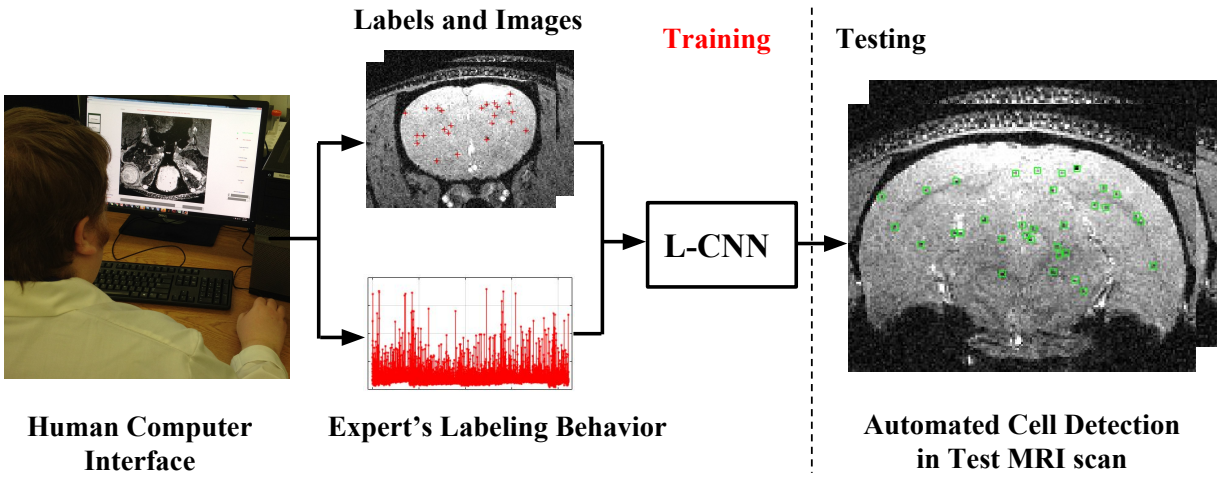

Figure 5.1: This chapter describes a CNN architecture that incorporates the *labeling behavior* of an expert during the training phase. The labeling behavior is anticipated to provide side information that captures the intra-class variability of positive exemplars in a two-class problem. (Note: Green markers have been used to indicate spots in a MRI scan).

information is only available for samples of one class and missing for the other class(es).

In this regard, the contributions of this chapter are three-fold:

◇ Utilizing labeling latency as an additional variable for learning in the context of a medical imaging application.

◇ Introducing the problem of exploiting side information that is only available for one class.

◇ Designing a new CNN framework, L-CNN,[1] that exploits labeling latency as side information.

5.1.1 Prior literature

In this section, a brief overview of the related work is presented.

[1]The term L-CNN is used to indicate that the CNN exploits **L**abeling behavior.

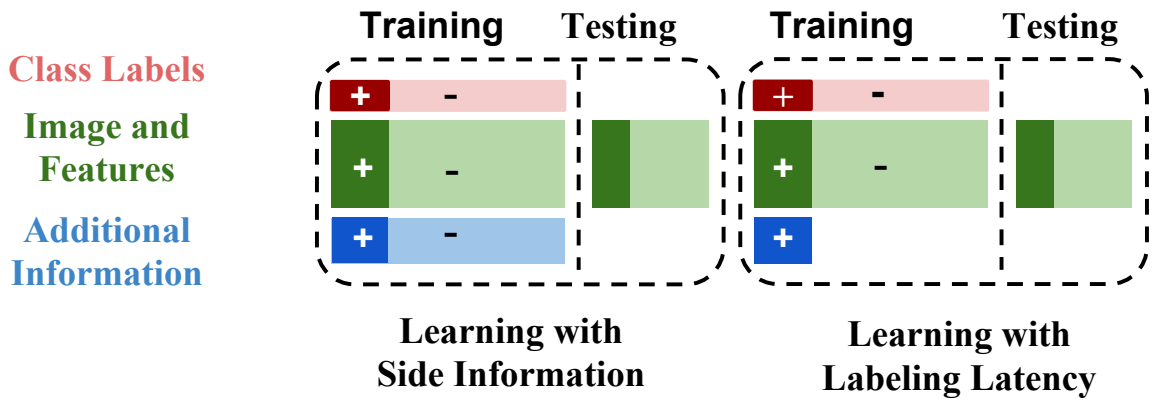

Figure 5.2: Unlike traditional features, labeling latency is only available during the training phase. Further, unlike traditional side information, it is associated with a single class only. In this figure, a two-class problem ("+" and "-") is considered.

5.1.1.1 Classifier learning with labeling latency

The literature on LUPI-based approaches is closely related to our work. The basic goal of the LUPI paradigm is to exploit side or privileged information that is available only during training and not during testing. Side information has been successfully utilized in the context of *unsupervised* learning frameworks [85, 86]. A number of approaches also show the benefits of using side information in a *supervised* learning framework [77, 78, 79, 80, 81, 82, 83]. However, in the supervised learning framework, existing LUPI approaches cannot be utilized if the side information is only available for a single class and completely absent for the other classes.

A recent study [84] utilized *reaction time* of a labeler as an additional side information in a SVM based framework. However, in [84], the side information is available for *both positive and negative classes*. In the experiments, an image was displayed for a very short period of time to the labeler, who had to indicate whether the image contained a face image or not. The reaction time to label was taken as the side information for each image that could potentially model the *classification difficulty* of each image.

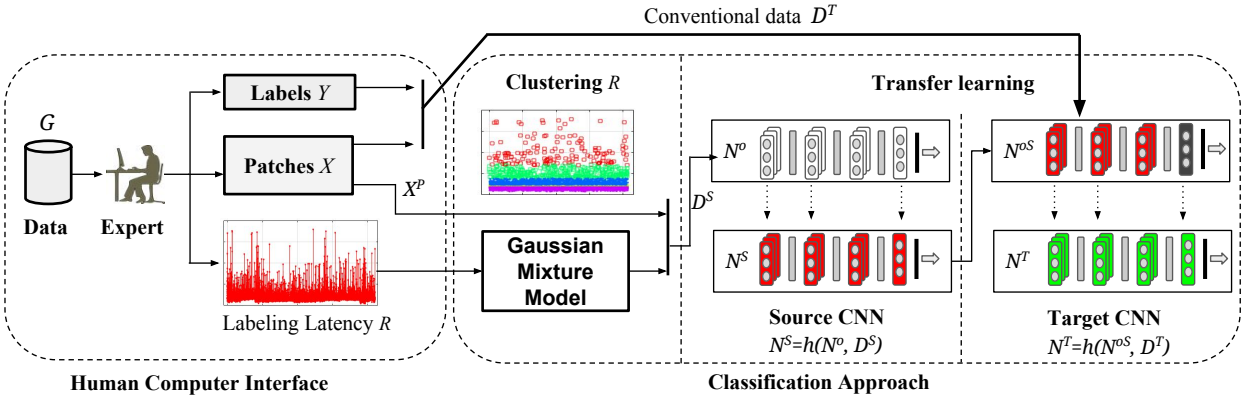

Figure 5.3: Basic architecture of the proposed L-CNN framework.

5.1.1.2 CNN learning with side information

The idea of exploiting side information with CNNs is relatively less explored, especially in the context of image based learning. The few approaches that have been studied [87, 88] suffer from the same limitation as standard LUPI approaches and, therefore, cannot be easily appropriated to the problem at hand. This study is one of the first to demonstrate how a CNN learning framework can exploit labeling latency.

5.2 Approach

The basic architecture of the proposed L-CNN framework is shown in Fig. 6.1.

The human computer interface (HCI) takes an MRI scan G as an input and extracts patches X from its slices, where each extracted patch can potentially contain a spot. During the training phase, it also allows the expert to label spots (positive patches) in each slice in an interactive manner using an image viewer (e.g., by zooming in, changing image contrast, etc.), resulting in a set of labels Y associated with these patches. Further, the labeling latency value associated with each *positive* spot label is also recorded. Labeling latency is utilized later as an additional source of information to *categorize* spot patches. After this, a transfer learning paradigm is adopted for

which the *source task* involves learning a CNN that can distinguish between these categories of spot patches. The *target task* is to differentiate between spot and non-spot patches using training data that has a limited number of positive examples. Since CNNs are initialization dependent and unfavorably impacted when the training data is limited, rather than using a randomly initialized CNN, the proposed approach transfers layers from the source CNN to the target CNN. The results obtained with this approach are significantly superior to previous state-of-the-art for the problem of spot detection in MRI scans.

5.2.1 Image viewer: Human-computer interface

This module has two main purposes: First, it is used to obtain ground truth on spots and to record labeling latency when an expert manually locates spots in an MRI scan. Second, after the spots have been labeled by the expert, the system extracts patches (described below) from the MRI slice. An extracted patch containing a *clicked* pixel is labeled as a positive sample (containing a spot) and the corresponding labeling latency is associated with it. The remaining patches are labeled as non-spot patches (negative samples).

5.2.1.1 Labeling spots

Given an MRI scan G, this module presents a software with zooming and contrast adjustment capabilities for the expert to carefully analyze each 2D slice in the MRI scan and *click* on the spatial location to indicate a spot. Collectively, these $3D$ locations are denoted by a set $\Phi = \{\cup\phi_u\}_{u=1}^{k}$ where ϕ_u denotes the location of a clicked point and k represents the total number of clicked points. In addition, the *time lapse* between clicks is also recorded as *labeling latency*. It was observed that experts label the *easier* spots first without engaging in any detailed analysis as shown in Fig. 5.4. Difficult-to-label spots, on the other hand, were typically labeled at the

end. Every potential spot entity is carefully analyzed by the medical expert by zooming-in and, occasionally, by changing the contrast of the locally selected region (see Fig. 5.5). Labeling latency is indirectly related to the cognitive overload involved in labeling a point as a spot. It is denoted as $R = \{\cup r_u\}_{u=1}^{k}$ where r_u is the labeling latency associated with the clicked point ϕ_u. Certain latency values were rather high, due to breaks taken by the expert or due to distractions. Hence, values greater than 45 seconds were simply replaced with the *mean time* taken between mouse clicks. These pre-processed values of labeling latencies corresponding to one MRI scan are shown in Fig. 5.6. Factors such as spatial distance between two consecutive clicks have little or no effect on labeling latency (as small movements on the mouse, translate to large spatial distances on the screen). For simplicity, any possible effects of such factors are ignored.

Since the labeling task is laborious, experts are asked to indicate positive samples (spot entities) only. However, this means that latency information is available only for positive samples and not for the negative samples. For experts, this creates an easier and more practical labeling environment. However, from a pattern classification standpoint, this introduces a new challenge: “features" that are available during training but not during testing and are associated with one class only.

5.2.1.2 Extracting RoI for classification

The annotated (i.e., clicked) locations, Φ, must be associated with some regions in the MRI scan; these regions, representing a collection of pixel intensities, will then be input to the classifier. The question to address here is, how should a region be defined? This has been discussed before in Chapter 2. For a brief overview, a superpixel [89] based strategy similar to [18] is utilized to extract a large number of patches from MRI G and these regions are denoted as $X = \{x_1, x_2, ...x_n\}$ where $n > k$. As shown in Fig. 3.2, for each superpixel in a 2D MRI slice, a patch of size $z \times z$ is

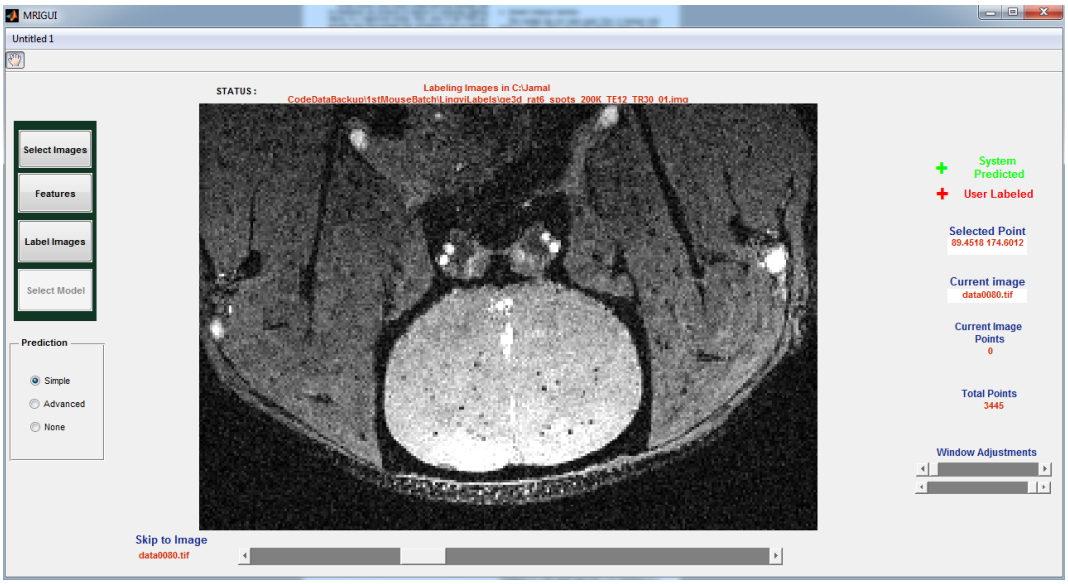

Figure 5.4: Basic view is used to label easy-to-detect spots.

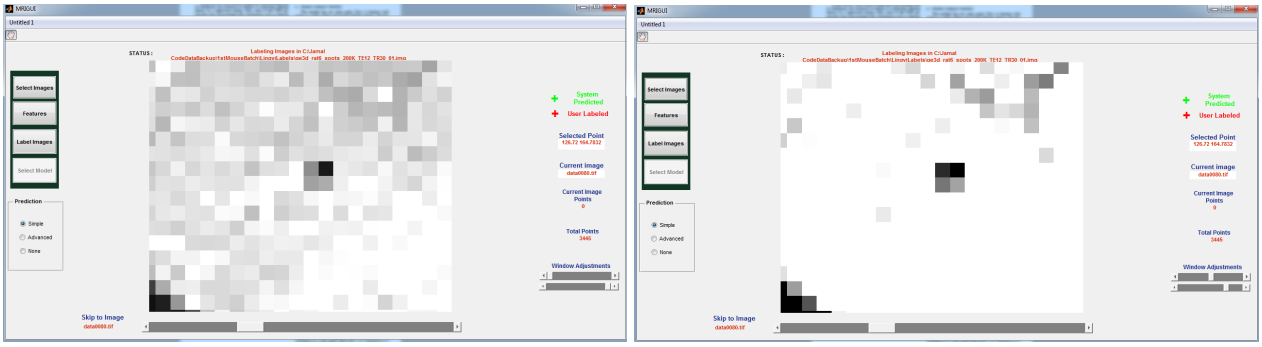

Figure 5.5: (Left) Zoomed-in view to locate a spot. (Right) Zoomed-in view with contrast adjustment for detailed contextual observation prior to labeling spots.

extracted by keeping the darkest pixel of the superpixel at the center (since spots typically appear darker than the surrounding tissue). Note that a superpixel approach is preferred over a dense sampling method for defining regions as the former results in fewer but more relevant patches for further processing and is less likely to associate multiple spots to a single patch. Further, as mentioned in [18], the superpixel algorithm used in [89][20] outperforms other superpixel and $3D$ supervoxel algorithms (such as $3D$ SLIC) for capturing *spot boundaries*. Therefore, the $2D$ spot patches detected in neighboring slices are later joined to form a $3D$ spot.

Formally, $X = \{x_1, x_2, ...x_n\}$ where each patch x_u has a center a_u. Note that a_u^q denotes the slice

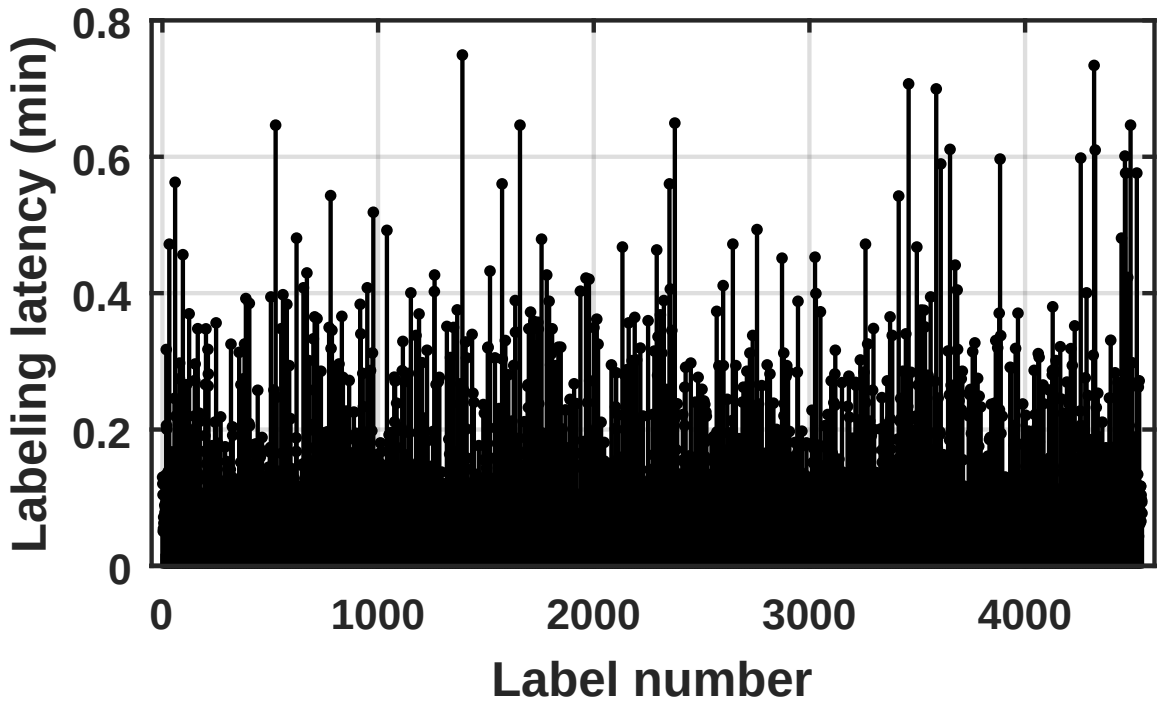

Figure 5.6: Labeling latency for a single MRI scan.

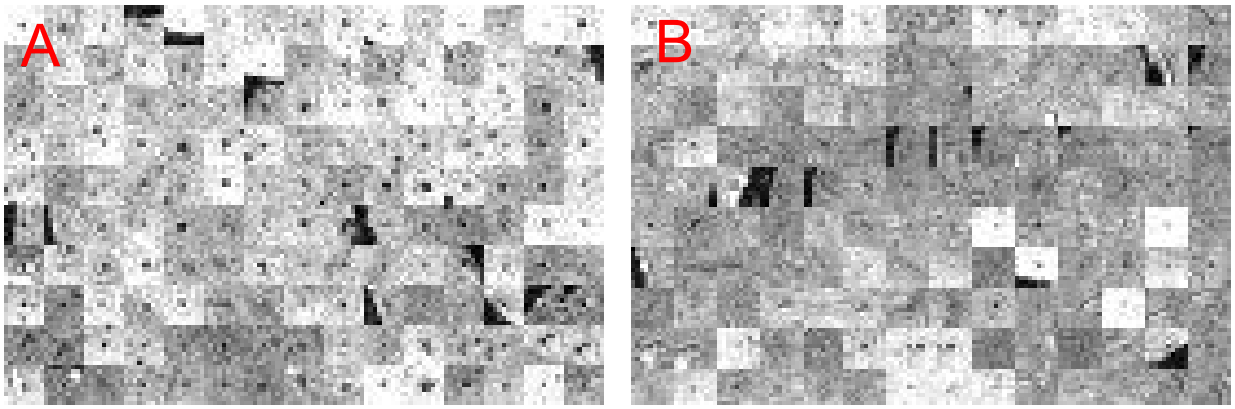

Figure 5.7: (A) Spot patches extracted from one MRI scan (concatenated as 10×15 patches), (B)Spot patches from another MRI scan. These patches represent inter-scan and intra-scan variations in spot patches.

number where the patch is located based on the $2D$ spatial location defined by a_u^1 and a_u^2. The distance between the center location of each patch with respect to all the clicked locations in Φ is computed. If the smallest of these distances, d_{min}, is less than or equal to a pre-defined threshold τ, the patch is considered to be a *spot*; otherwise, it is considered to be a non-spot that forms

the negative class of the dataset. Note that $d_{min} = \min_j \|\phi_j - a_u\|_2$ and $1 \leq j \leq k$. Based on this step, a label y_u is assigned to each patch x_u. Thus, $Y = \{y_1, y_2, ..., y_n\}$ where $y_u \in \{0, 1\}$. Further, $X = \{X^p, X^g\}$ where $X^p = \{x_1, x_2, ...x_k\}$ denotes all the spot patches while $X^g = \{x_{k+1}, x_{k+2}, ..., x_n\}$ represents all the non-spot patches. Note that $\forall x_u \in X^p$, the training samples exist as triplets $\{\cup(x_u, y_u, r_u)\}_{u=1}^{k}$, while for the remaining non-spot patches the labeling latency values (r_u) are *not* available.

5.2.2 Classification approach

5.2.2.1 Clustering

The variation in labeling latency values for an MRI scan can be seen in Fig. 5.6. A Gaussian Mixture Model (GMM) is utilized here to categorize X_p into m clusters based on their corresponding labeling latency values. Here, m denotes the optimum number of clusters selected using the standard Akaike infomation criterion. Thus, each patch $x_u \in X^p$ is associated with a clustering index v_u where $1 \leq v_u \leq m$ and $V = \{v_1, v_2, ..., v_k\}$.

5.2.2.2 Transfer learning

In transfer learning, the knowledge gained to perform one task (source task) is used to benefit learning in another task (target task). In the context of CNNs, transfer learning can be implemented by transplanting the learned network layers (feature representations) from a source CNN for initializing a target CNN. In this study, the target task is the task of separating spot patches from non-spot patches. To train a CNN to perform this task, one approach would be to initialize it randomly and then utilize the target task training data to update it. Another approach would be to initialize it based on the layers of a source CNN that has been trained for a different task. Here,

the source task involves differentiating between categories of spots generated using the clustering process described earlier. The layers of the ensuing source CNN is then used to initialize the target CNN.

The source dataset $D^S = (X^p, V)$ is developed, where the cluster indices in V act like labels for the spot patches in X^p. A CNN, N^S, is then trained to distinguish between the patches of these clusters. Formally, the goal is to learn weights for N^S that minimize the loss $\sum_{u=1}^{k} J(N^S(x_u), v_u)$ where J denotes the standard cross-entropy loss function. Functionally, this CNN learning process is denoted as $N^S = h(N^o, D^S)$ where N^o represents a randomly initialized CNN architecture and N^S denotes the CNN that has learned a feature representation (weights) to distinguish between spot patches belonging to different clusters. The CNN architecture customized for this data is shown in Fig. 3.6. Note that due to the small size of the input patches ($z = 9$), a pooling layer has not been utilized.

In the second step, the goal is to utilize the target dataset $D^T = (X, Y)$ for learning a target CNN, N^T, which can distinguish between spot and non-spot patches. Formally, the objective is to minimize the loss $\sum_{u=1}^{n} J(N^T(x_u), y_u)$. However, in this case, N^T is not randomly initialized; rather the feature layers in N^S are transplanted to initialize it. This is denoted as $N^T = h(N^{oS}, D^T)$. The transfer is conducted in the standard manner detailed in [37, 40]. In this case, all the convolutional layers are transferred for initialization. The fully connected layer is incompatible for transfer due to structural differences induced by the two tasks. Therefore, as typically done in transfer learning, the fully connected layer is randomly initialized. The resulting initialized CNN is denoted as N^{oS}. Experimental results show that this transfer of knowledge brings an improvement that is not achieved when using a randomly initialized CNN $\overline{N^T} = h(N^o, D^T)$ that is updated using only the dataset D^T. Note that it may be possible to achieve different levels of improvement based on the labeling behavior of different experts. This could be one interesting direction to explore in the

future.

5.3 Experiments, results, and discussion

In this section experiments are designed to answer the following questions: (a) How does the L-CNN compare with a traditional CNN? (b) What is the result if random clustering is used instead of GMM based clustering? (c) What is the effect of transferring different number of CNN layers? (d) How do the results obtained in this study compare with the previous state-of-the-art for spot detection in MRI scans? Note that in all experiments, the Area Under the Curve (AUC) value was used as measure of accuracy.

Setup In this study, the *in vivo* MRI database of [18] comprising 5 MRI scans of different Rat brains was used. 3 of these Rats were injected with Mesenchymal stem cells which appear as dark spots in MRI. About $100,000$ patches are extracted from each of the 3 scans. The number of positive samples in each scan is about 5000. The labeling latency for each labeled patch was also documented. Each of the three scans is successively used for training while the remaining two independent MRI scans are used for testing. This creates 6 testing scenarios. The following parameters were used: $z = 9$, $m \in [5,9]$, and $\tau = \sqrt{2}$.

5.3.1 Comparison with conventional CNN approach

In this experiment, the result of L-CNN is compared with a conventional CNN $\overline{N^T}$ that is randomly initialized and then simply trained using D^T. Results in Fig. 5.8 clearly demonstrate that the L-CNN results in better performance than the conventional CNN on all 6 testing scenarios. It is interesting to note that exploiting labeling behavior using L-CNN can provide a performance increase of up to 4% (see test set $T3$). Thus, the significance of labeling behavior in performance

improvement has been clearly established.

5.3.2 Comparison with random clustering

It can be seen that the L-CNN architecture exploits clustering to create sub-categories of the labeled spot patches. In this experiment, we investigate the performance when spot patches are randomly assigned to categories instead of using GMM. These results are shown in Fig. 5.8 for each of the 6 testing scenarios, and also compared with the L-CNN. In all testing scenarios, L-CNN clearly performs better when GMM is used instead of random clustering. Further, it can be seen that the performance due to random clustering is, in general, very similar to that of the conventional CNN.

5.3.3 Comparison using different number of transfer layers

Here, the effect of transferring different layers is investigated. The proposed CNN architecture has three convolutional layers and a fully connected layer. The results of transferring different number of convolutional layers are shown in Fig. 5.9. It is evident that , in general, transferring all three layers results in superior performance.

5.3.4 Comparison with a previous approach

We compare the results of L-CNN with the previous state-of-the-art for spot detection reported in [18]. For this comparison to be compatible, a leave-2-out approach was utilized using the same experimental setup mentioned in [18]. The proposed approach clearly results in superior performance with an accuracy of 94.68%, compared to the 89.1% accuracy achieved in [18].

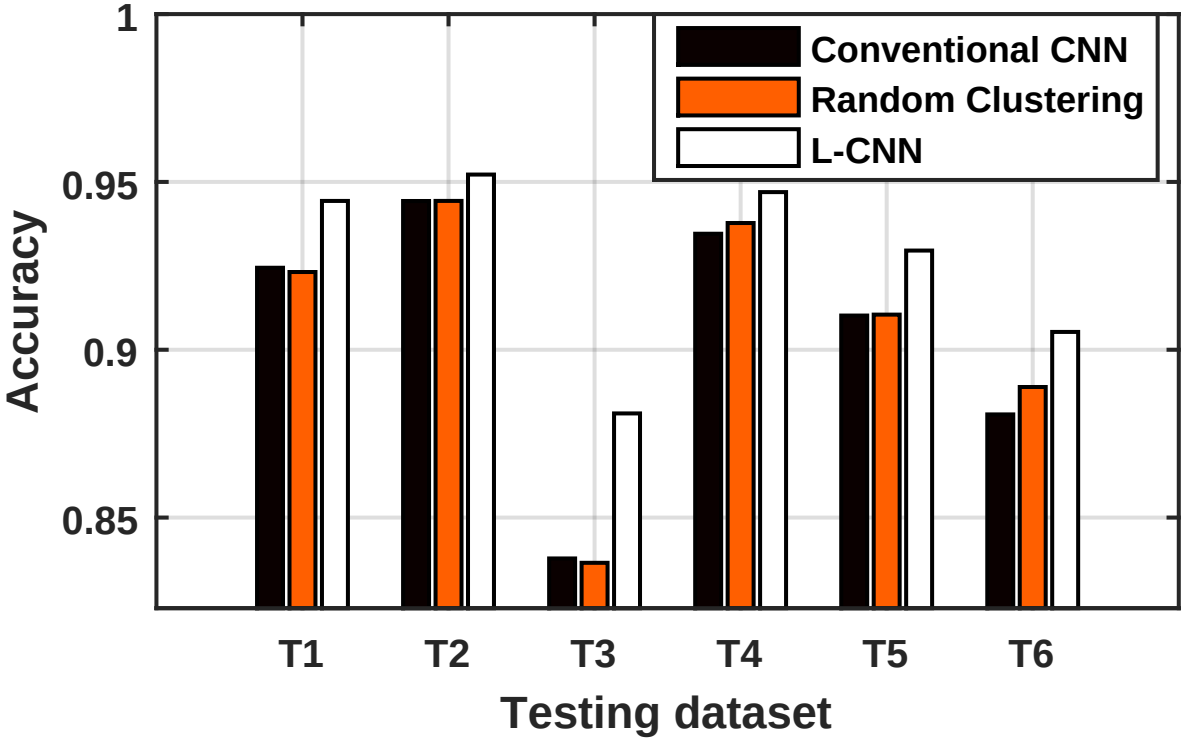

Figure 5.8: Performance of the proposed L-CNN.

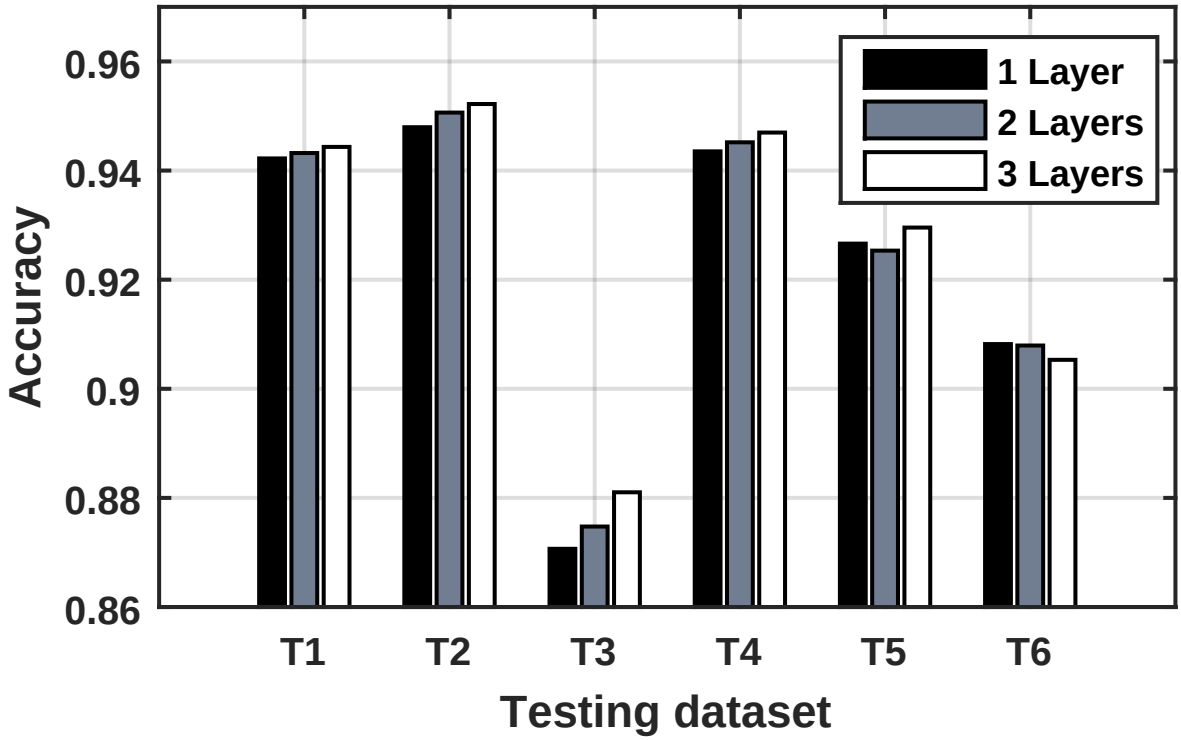

Figure 5.9: Results with different number of transfer layers.

Chapter 6

Supplementary Information

In order to make this thesis self-contained, information on our related supplementary studies is presented in this chapter. The following two studies have been referenced in the main chapters:

1. **A model based approach for spot detection:** This study was briefly discussed in Chap. 3. details of the approach and the experimental results are discussed here.
2. **CNN ranking with intuitive approach:** This approach was briefly discussed in Chap. 4. Experimental results using this approach are presented here.

6.1 A model based approach for spot detection

This section presents supplementary information on our learning-based approach that utilizes a spot model. The limitations of this approach were discussed in Chap. 3. In this approach, we consider spots as 3D entities and represent its general structural model using superpixels. We then extract a novel set of "superferns" features and finally classify it using multiple definitions of spots learned by a partitioning-based ensemble of Bayesian networks. Experimental results show that it performs significantly better than previously related approaches.

In summary, this chapter makes the following contributions: (i) It proposes a novel superpixel-based 3D model to characterize cellular spots that can potentially be used in other medical problems. (ii) It introduces the *superferns* feature that exploits superpixel-based representations and is more discriminative than traditional fern features. (iii) It demonstrates *how* a partitioning-based ensemble learning can be effectively utilized for MRI spot detection.

6.1.1 Approach

As mentioned before, the cell/spot detection problem in MRI scans has unique challenges, where a number of questions should be carefully considered prior to algorithm design. First, since a spot is essentially a 3D entity in an MRI cube, *how* to model its three dimensional characteristics? Second, a spot is also a small group of dark pixels with varying shapes and sizes. *What* is the basic *unit* within an MRI cube (e.g., one, two, or N pixels) for which the two-class classification decision can be made? Third, there is a huge number of candidate locations. Therefore, our feature representation for spots should be not only highly discriminative, but also efficient and based on computationally light operations. Fourth, the appearance of a spot varies relative to its local and regional neighborhood. *How* to make learning robust to these variations should be addressed.

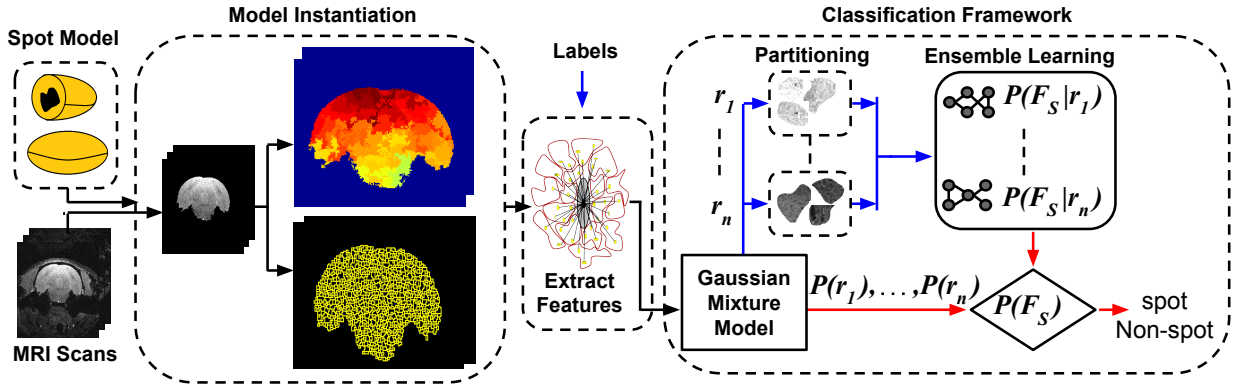

Figure 6.1: The architecture of our approach. Blue, red, and black arrows are the processing flow during the training stage, testing stage and both stages, respectively.

Considering these challenges, we design our technical approach as in Fig. 6.1, with details below.

6.1.1.1 Spot modeling

Visually, a cellular spot **S** appears as a cluster of N dark 3D pixels with high variations in its 3D shape and intensity, wrapped inside a cover of background pixels. In this work, we call the small group of dark pixels as a spot's interior I, and their local neighboring pixels in the background as the exterior E of a 3D spot. This model is consistent with the manual labeling of spots by domain experts, who inspect the cross-sections of these spots in consecutive 2D MRI slices, and look for a small region (interior) that is darker than its neighboring pixels (exterior). Furthermore, human eyes can also adjust the amount of relative darkness based on the characteristics of the *specific brain region* containing that spot. Therefore, in addition to model a spot with its interior/exterior, we also model the specific region it belongs to, termed *region context* R.

6.1.1.2 Model instantiation via superpixel

Given the conceptual spot model $\mathbf{S} = \{I, E, R\}$, we now describe how to define I, E, and R for a spot, by three steps. Since no spot should be outside the brain region, the first step is to perform brain segmenation in every 2D MRI slices with basic image processing techniques. The second

step is to define I and E by applying 2D superpixel extraction [89] to the segmented brain region of each MRI slice. A superpixel is a group of N neighboring pixels with similar intensities, i.e., $V_{z,u} = \{x_i, y_i, z\}_{i=1}^{N}$ where u is the superpixel ID in slice z. In general superpixels can tightly capture the boundaries of a spot's interior; however, some imprecise localization is also expected in practice (see Chap. 3). After extraction, we denote $\mathbb{M} = \{V_{z,u}\}_{z=u=1}^{L,U}$ as the set of all superpixels in the brain region, where L and U are the number of slices and superpixel IDs, respectively. Due to the exclusiveness of the interior and exterior of spots, we have $\mathbb{M} = \mathbb{I} \cup \mathbb{E}$ where $\mathbb{I}$ and $\mathbb{E}$ are the set of all interior or exterior superpixels, respectively. With that, for a spot $\mathbf{S}$ with length l in z-axis, we formally define its interior as $I = \{V_{z,u}, \cdots, V_{z+l,u} \mid V \subset \mathbb{I}\}$ and the exterior as $E = \{V_{z-1,.}, V_{z,\bar{u}}, \ldots, V_{z+l,\bar{u}}, V_{z+l+1,.} \mid ||(m(I) - m(V)|| \leq \tau, V \subset \mathbb{E}\}$, where $m()$ computes the mean of a set, τ is the maximum L^2 distance between the centers of a spot and an exterior superpixel, and $V_{z-1,.}$ and $V_{z+l+1,.}$ are superpixels in two adjacent neighboring slices. Assuming the second step extracts N_1 superpixels per slice, the third step also relies on superpixels to define R where the number of extracted superpixels $N_2 \ll N_1$. This is reasonable since R can include very large superpixels that are representative of the regional appearance. Thus, we define the region context of a spot as $R = \{\tilde{V}_{z,v} \mid m(I) \subset \tilde{V}_{z,v}\}$, which is the large superpixel enclosing the spot center $m(I)$.

The superpixel-based 3D spot model has a few advantages. First, it addresses the issue of *unit*, by going beyond pixels and using the superpixel-based model for feature extraction and classification. Second, this model substantially reduces the number of total candidate spots to be tested, since the candidates can be nominated based on superpixels rather than pixels. Note that we may extend our model instantiation by using 3D supervoxel instead of 2D superpixel. We choose the latter in this work due to its demonstrated reliability and efficiency during the experiments.

6.1.1.3 Superferns feature extraction

With an instantiated spot model $\mathbf{S} = \{I, E, R\}$, the next step is to extract a discriminative and efficient feature representation. Since a spot generally has darker interior than its exterior, it makes sense to define features based on the computationally efficient intensity differences between pixels in the interior and exterior. Difference-based fern features have shown great success in computer vision [90]. Ferns compute the intensity difference between a *subject pixel* and another pixel with a certain offset w.r.t. the subject pixel. Using the same offset in different images leads to feature correspondence among these images.

For our problem, the spot center $m(I)$ can be regarded as the subject pixel, and its intensity is the average intensity of all interior pixels $m(\mathbf{G}(I))$. We then randomly generate h 3D offsets $O = \{\mathbf{o}_i\}_{i=1}^{h}$ with a uniform distribution, whose center is the spot center and radius is τ. Finally, the feature set is computed as $F = \{f_i\}_{i=1}^{h}$, where $f_i = \mathbf{G}(m(I) + \mathbf{o}_i) - m(\mathbf{G}(I))$. While f_i is efficient to compute, $\mathbf{G}(m(I) + \mathbf{o}_i)$ is the intensity of a single pixel, which can be noisy, specially in in-vivo MRI and lead to low discriminability of f_i. Thus, it is desirable to replace it with the average intensity of all pixels within an exterior superpixel. However, the exterior superpixels around different spots have no correspondence, and, as a result, f_i for different spots also have the correspondence issue.

To address this issue, we present an approach to exploit the average intensity without losing correspondence information. The new feature, termed as “superferns", is similar to F except it replaces the single pixel-based intensity with the average intensity of the superpixel, i.e., $F' = \{f_i'\}_{i=1}^{h}$, where $f_i' = m(\mathbf{G}(V)) - m(\mathbf{G}(I)), \forall\, m(I) + \mathbf{o}_i \in V$. Note that it is possible to have the same feature at two different offsets due to them being in the same superpixel, i.e., $f_i' = f_j'$. This is not an issue because this equality may not be true for other spots, hence the feature distributions of f_i'

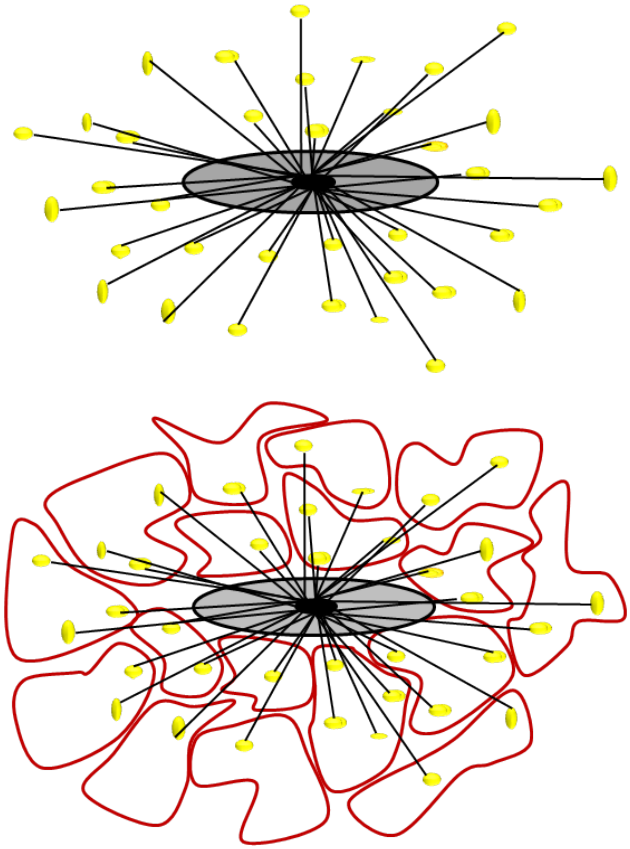

Figure 6.2: Ferns vs. Superferns.

and f'_j are not the same, and they contribute differently to the classification.

Features are also needed for the region context R. Given its role of supporting region-dependent classifiers, we find that simple features work well for R, e.g., the mean and standard deviation of pixel intensities in R, $F_r = (m(\mathbf{G}(R)), \sigma(\mathbf{G}(R)))$.

6.1.1.4 Partition-based bayesian classification

Having computed the feature $F_s = (F, F_r)$ for a set of spots and non-spots, we now present our approach to learn an accurate two-class classifier. Since different local regions have different appearance, we partition the brain region into N_0 partitions, learn a set of N_0 classifiers each for one region, and fuse them via a probabilistic Bayesian formulation. Specifically, for any spot

candidate **S**, its probability of being a spot is

$$P(F_s) = \sum_{i=1}^{N_0} P(F_s, r_i) = \sum_{i=1}^{N_0} P(F_s|r_i)P(r_i), \tag{6.1}$$

where r_i represents the i^{th} partition, $P(r_i)$ is the probability of **S** belonging to r_i, and $P(F_s|r_i)$ is the conditional probability of a spot at r_i.

We learn $P(r_i)$ using the well-know Gaussian Mixture Models (GMM) technique. By collecting F_r for all training samples, we perform GMM to estimate N_0 component Gaussian densities, each considered as one partition. During the testing, $\{P(r_i)\}_{i=1}^{N_0}$ is obtained by evaluating F_r of the testing sample w.r.t. each component densities. In order to learn $P(F_s|r_i)$, we group all training samples into N_0 groups based on their respective maximum $\{P(r_i)\}$, and train the $P(F_s|r_i)$ using the standard implementation of Bayesian Networks in [91]. During the test, for a testing candidate spot, GMM enables a soft partition assignment, and its final probability of being a spot is the weighted average.

6.1.2 Experimental results

In this section we design experiments to investigate answers to the following questions: (i) how does our approach perform and compare with the previous approaches using both *in vivo* and *in vitro* data? (ii) how does the discriminating potential of superferns quantitatively compares with the fern features? (iii) how diverse is the classifier ensemble created by our proposed approach?

6.1.2.1 Experimental setup

The ROC, and Area under the Curve (AUC) are used as the evaluation metrics. For the 5-scans *in vivo* data (G_A), we adopt a leave-two-out scheme such that our testing set always contains one

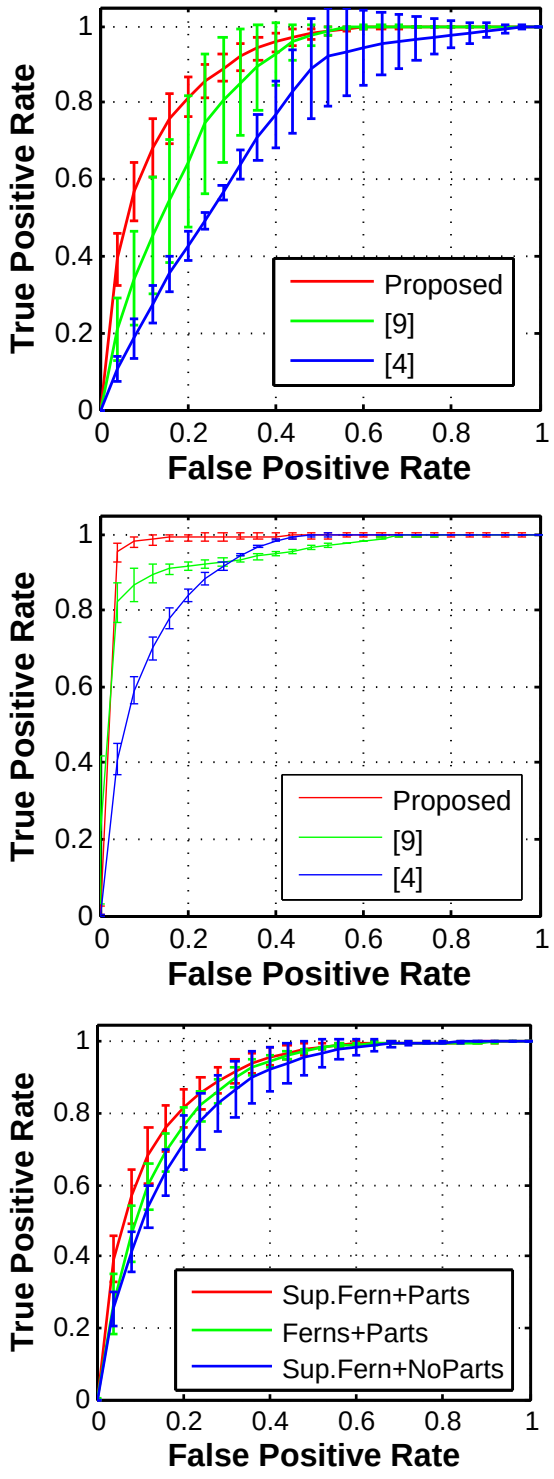

Figure 6.3: Detection performance comparisons and with various components.

labeled and one spotless scan. This creates six pairs of training and testing sets, which allows us to compute the error bar of ROC. For the 4-scans *in vitro* data, three pairs of training and testing sets

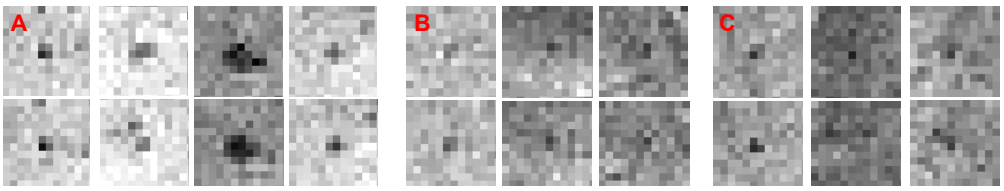

Figure 6.4: Spot detection examples: (a) true detection, (b) false negative, (c) false alarm.

are formed such that the naive scan always remains in the testing set accompanied by every other scan once. We implement the prior work of [92] and [16] and use them as the baselines, since they are the most relevant examples of MRI cell detection using learning-based and rule-based methods. We experimentally determine $\zeta = 2$, $\tau = 9$, h varies from $200-2000$ and q from $20-60$ depending on the size of brain regions.

6.1.2.2 Performance and comparison

As shown in Fig. 6.3 (a,b), the proposed method outperforms two baselines with an average AUC of 98.9% *(in vitro)* and 89.1% *(in vivo)*. The improvement margin is especially larger at lower FPRs, which are the main operation points in practice. Further, Fig. 6.3(c) shows that with *in vivo* data, by using ferns instead of superferns or by making no partitions of the brain region, we observe a decrease in performance to 85.3% and 87.1%, respectively.

Fig. 6.4 shows three types of spot detection results with our method. Each column represents two consecutive slices of one spot. The appearance and shape variations among the spots clearly show the challenge of this problem.

6.1.2.3 Superferns vs. ferns

To further illustrate the strength of the novel superferns feature, we compare the discriminating potential of superferns with ferns, *regardless* the classifier design. Information gain is a standard tool to measure the worth of a feature, where a higher gain indicates its higher discriminating

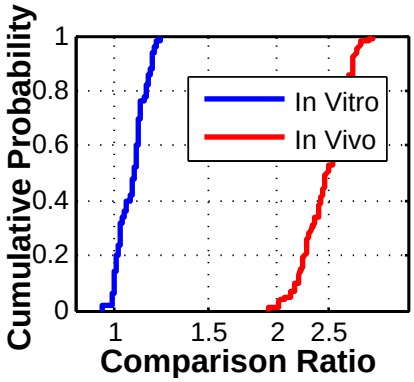

Figure 6.5: Superferns vs. Ferns

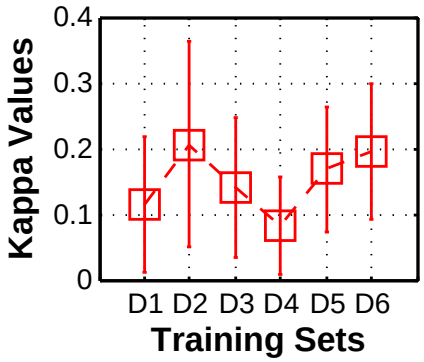

Figure 6.6: Classifier diversity analysis

potential. Given a set of 50 randomly generated offsets $\mathbf{o}_i$, we calculate their superferns features on the *in vitro* training data including both spots and nonspots, which allows us to compute the information gains A_s of each offset or superfearn. The same offsets are applied to the ferns features and results in their information gains A_f. Then we compute the ratio of two information gain, $\frac{A_s(i)}{A_f(i)}$, for $\mathbf{o}_i$, and collectively their cumulative density function (CDF) is shown in Fig. 6.5. Using 100 random offsets, the same experiment is repeated for the *in vivo* data. The fact that almost all ratios are larger than 1 shows the superiority of superferns.

6.1.2.4 Diversity analysis

Our classification framework includes an ensemble of classifiers, one for each partition. Since diverse discriminative features are utilized in different partitions, learning on disjoint partitions should favor high diversities among classifiers, which is an strong indicator for effective classification. To evaluate the diversity of our classifier ensemble, we use the standard Cohen's kappa value as [93], which ranges from 0 to 1, with a lower value indicating a higher diversity. For each of six in vivo training sets, we compute $\frac{N_0(N_0-1)}{2}$ kappa values, each between a pair of classifiers learned on different partitions. Fig 6.6 shows their mean and standard deviation for each training set. Based on the study in [93], we consider our kappa values to be very low, indicating the high

diversity in our learned ensemble.

6.2 CNN ranking with intuitive approach

This section provides supplementary information on the experiments and results of the intuitive approach for CNN ranking which was proposed in Chap. 4.

6.2.1 Experimental setup

6.2.1.1 Target task

Since many medical applications specifically suffer from the lack of large scale annotated data, in this study, an existing, real world MRI database [18] was utilized as a target task. In this thesis, this set is denoted by G_A. This database has three different sets of labeled MRI scans pertaining to rat brains. The injected stem cells appear as dark spots in these images. From each scan about $100,000$ non-spot patches and 5000 spot patches were extracted as mentioned before. These patches were obtained directly from the authors in [18]. In the experiments below, all patches from a single scan (single set) were used for training and the patches from the remaining two scans were independently utilized for testing, generating a total of 6 testing scenarios. In all the experiments, the Area Under the Curve (AUC) was utilized for summarizing classification accuracy.

6.2.1.2 Source task

The focus of this study is to rank a set of *given* source tasks in order to determine their transferability for a fixed target task. Therefore, 25 diverse source tasks were *arbitrarily* designed using the publicly available, standard ImageNet database. Fourteen of these were binary classification tasks, while the number of classes ranged from 5 to 20 for the remaining source tasks. Note that the goal here is to be agnostic to the data characteristics of a source and only utilize the weights learned by

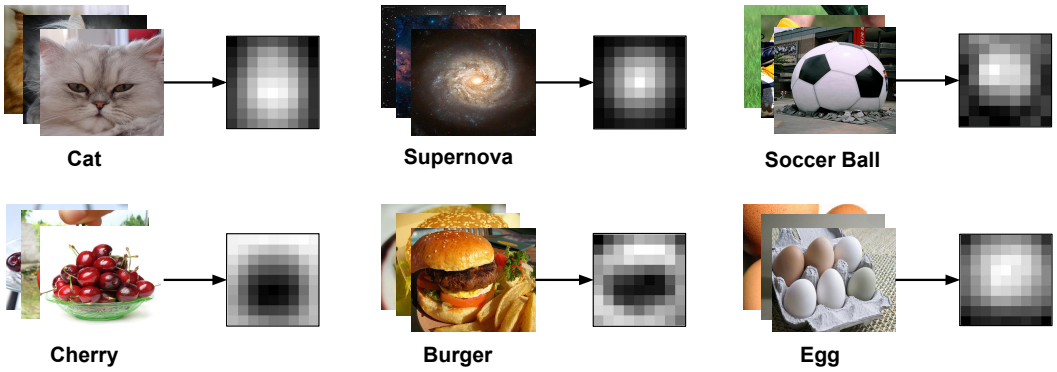

Figure 6.7: Transforming source images to 9×9. Transformed, average images for different entities are shown here.

the source CNN to assess its transferability to the target domain.

In the following sections, experiments are designed to study the following questions: (1) How well does the proposed approach rank the sources from best to worst? (2) What is the difference in performance when the best ranked source is used for transfer learning in comparison to the worst ranked source? (3) What is the gain in classification accuracy when results are compared against a CNN without any transfer learning? (4) How does the size of the target training data impact the performance gain? (5) What role does the choice of layers, that are transplanted, have on transfer learning?(6) Does the information fusion of sources provide robustness against ranking errors? (7) Can the negative impact of transfer learning be predicted in advance, based on the source task's ranking score? (8) What does the ranking score of a source task tell us about its data characteristics?

6.2.2 Results and discussion

6.2.2.1 Impact of size of target training set

In this experiment, we compare the following: (1) The performance of transfer learning when using the source that was ranked the best against the source that was ranked worst by the proposed

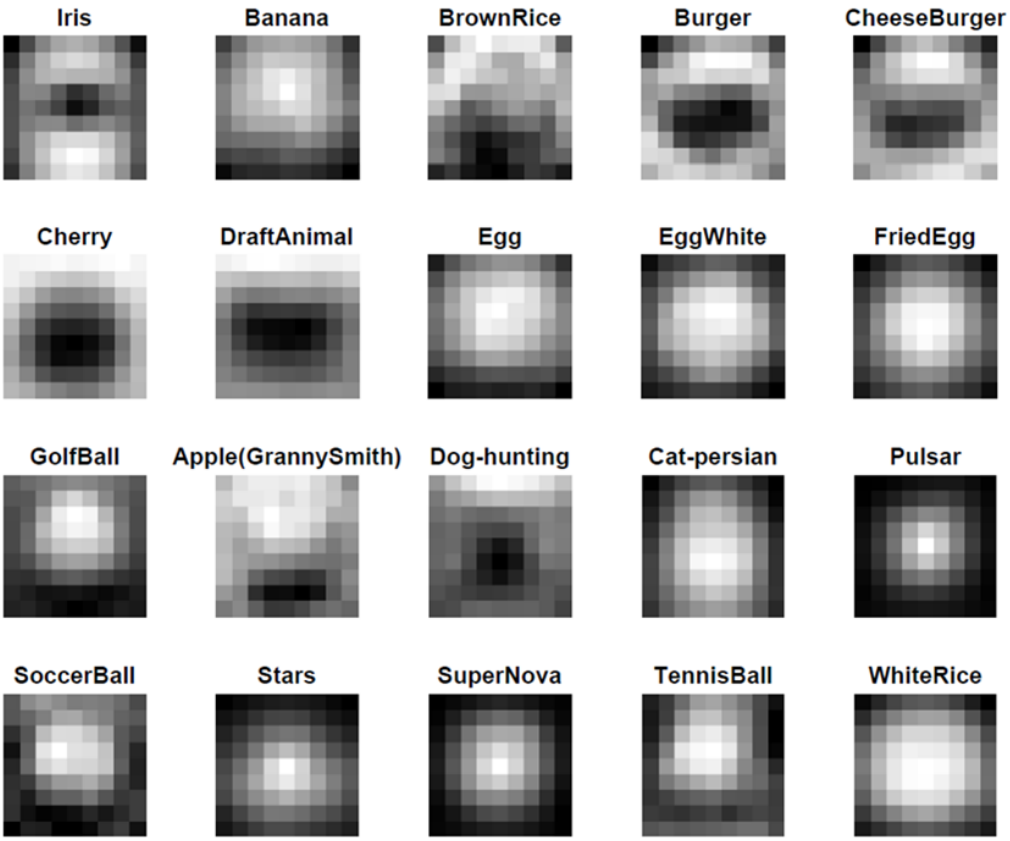

Figure 6.8: Source entities and their corresponding transformed average images.

source ranking approach. (2) Performance of best and worst ranked against a baseline CNN that was only trained using target training data X with no transfer learning. (3) Performance of the aforementioned CNNs when using a different proportion of target training data. Training was accomplished with 12 different percentage values that ranged from 5% to 60% of the training set in increments of 5. Fig. 6.9 shows the results on three different testing scenarios. Fig. 6.9(left) indicates a performance gain of about $\sim 35\%$ with respect to the baseline when only 5% of the target training data is used! We further observe that the performance gain is more significant when the training data is small in size which is precisely the scenario envisioned in this study.

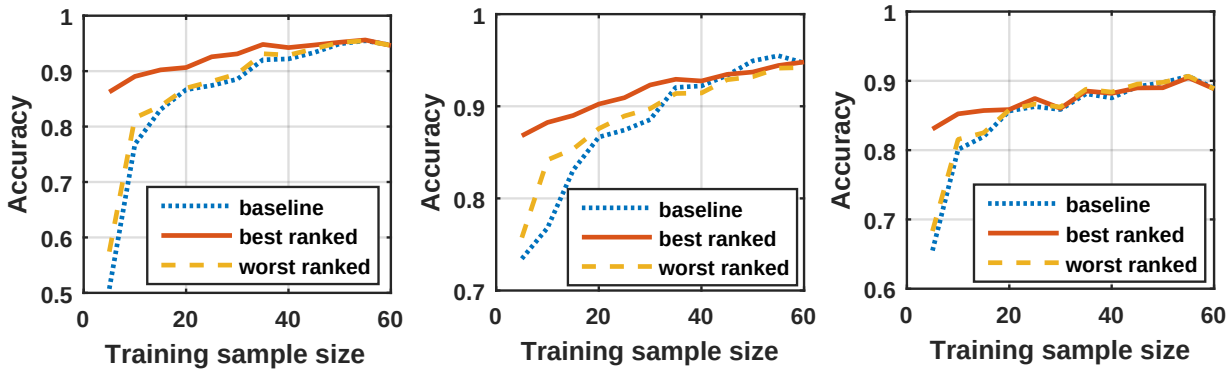

Figure 6.9: Comparison of empirical results on three of the six testing scenarios. Note the performance gain on datasets with smaller amounts of training data and the efficacy of the ranking metric.

6.2.2.2 Correlation between source ranking and performance gain

In Fig. 6.10 (A), the x-axis represents the ranking score of a source task that is computed using the proposed approach, with the top-ranked source having the largest value. The y-axis shows the normalized sum of the overall performance gain achieved by using that source in all the aforementioned 12×6 scenarios (12 different sized training sets, 6 test scenarios). This figure (6.10A) depicts the overall correlation, when utilizing training set sizes ranging from 5% to 60%. However, it is observed that such a correlation is significantly high when the size of the target training data is small, as can be seen in Fig. 6.10 (B,C,D,E). Performance gain is measured in terms of the difference between the Area Under the Curve (AUC) values. Since the criterion was specifically designed for small target training sets, this result is desired. From (F to I) in Fig. 6.10, the training data size increases and it can be seen that performance gain begins to decrease as the target training data is now sufficient for the spot detection problem.

6.2.2.3 Layers to be transferred

Fig. 6.11(A) denotes the classification accuracy when (a) only the most general layer (weights from the first convolutional layer only) is transferred from the source CNNs, and (b) all three

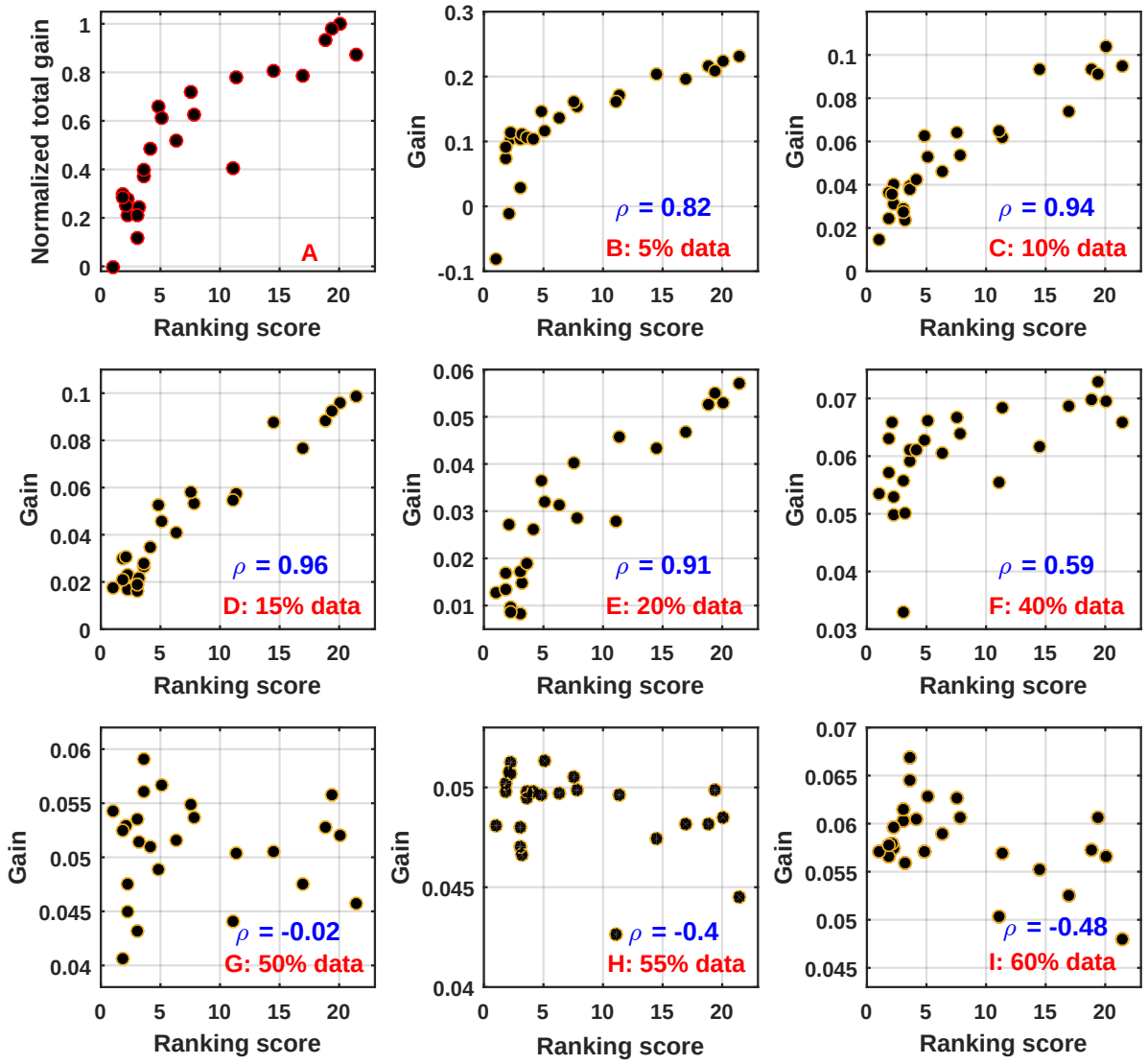

Figure 6.10: Correlation between ranking score and performance gain.

convolutional layers are transferred from the *best* ranked source. The shaded region represents the area between the curves plotted when only 1 layer was transferred from all 25 source CNNs (indicated by L1). Experimental results on all 6 testing scenarios clearly show that for sources with higher ranking scores, transferring all layers result in superior performance. For example, Fig. 6.11(A) shows that the shaded region lies completely under the best ranked source when all three layers of the source CNNs are transferred. In the future, we would like to utilize the ranking scores to determine the number of layers that can be transferred.

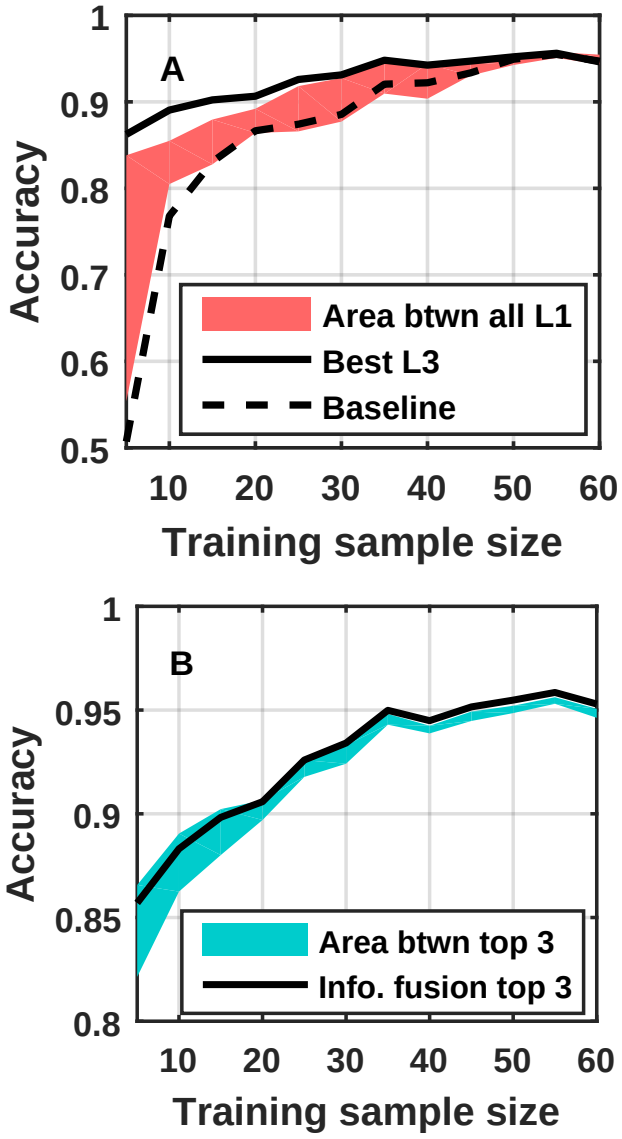

Figure 6.11: (A) Performance gain analysis w.r.t transferring layers. L1 indicates that only 1 convolutional layer was transferred and L3 that all 3 convolutional layers were transfered. The red regions shows the area spanned by 20 different sources, while the black line shows only the best ranked source out of 25. (B) Benefit of information fusion. (C) Correlation of ranking score with number of classes.

6.2.2.4 Benefit of information fusion

Transfer learning can involve transferring information from multiple source CNNs, Z, simultaneously, rather than from a single CNN only. Let two source CNNs be N_i and N_j, respectively. Let N_{ti} and N_{tj}, respectively, be the updated CNNs after transfer learning. The CNNs are updated using

the training data, from the target task. The output of each CNN is the set of probabilities indicating the posterior probability that the given input belongs to a particular class (label). Consider a test sample s. Then, each CNN will predict the class labels of the input data differently, the respective probabilities can be denoted as $P(s|N_{ti})$ and $P(s|N_{tj})$, respectively. These two expressions can be combined as:

$$P(s) = \zeta_i P(s|N_{ti}) + \zeta_j P(s|N_{tj}). \tag{6.2}$$

Using the ranking scores, the weights can be computed as:

$$\zeta_i = \frac{E_i}{E_i + E_j}, \quad \zeta_j = \frac{E_j}{E_i + E_j}. \tag{6.3}$$

For d sources, this approach can be extended such that $\sum_{k=1}^{d} \zeta_k = 1$.

Fig. 6.11(B) shows that the fusion approach can overcome any potential errors in ranking the sources. The shaded region displays the area between the top-3 ranked source CNNs. The result of the fused performance is plotted as a line which clearly stays near the top of this area. In cases, where 'poor' sources are ranked higher, using a fusion approach can prove to be more reliable.

Chapter 7

Conclusion

This work presented the first comprehensive study on learning based, automated spot detection and quantification in MRI. This work highlighted and addressed a number of challenges in this context. To utilize intelligent machine learning and computer vision approaches, the first annotated MRI database was developed for spot detection. An extensive study was conducted by designing a diverse set of learning based approaches which were evaluated using both *in vitro* and *in vivo* MRI scans. Evaluation was also performed against a known number of spots in *in vitro* MRI scans. The impact of resolution change in MRI was also studied. Further, in many medical applications such as this, it is challenging to collect a large volume of annotated data. In this study, we also investigated how accurate convolutional neural network architectures can be learned using transfer learning schemes despite using very limited training data. In fact, more than 35% improvement in accuracy was observed when training was conducted only with 5% of the available training data. In this context, a theoretical framework was also presented which can be also generalized to other related tasks. In addition, we also demonstrate that the labeling process of a medical expert can be incorporated into the classification framework.

It is important to note that MRI-based cell tracking has remained largely phenomenological for its history, starting in the late 80's. Moving forward, automated spot detection for MRI-based cell tracking would prove useful across a broad spectrum of research tracks. For example, Walczak et al, infused neural stem cells via the carotid artery in an effort to target stroke lesions [94]. High resolution *in vivo* and *in vitro* MRI appear to show small clusters of cells, perhaps even

single cells, distributed in the brain as a function of the intervention. Only qualitative analysis was performed on this imaging data; automated spot detection would have enabled quantitative metrics of cell numbers. Another application would be for the evaluation of transplanted islets encapsulated with iron oxide nanoparticles within alginate microspheres. These imaging features, typically are individual hypointensities, examples being [95] and [96]. In both cases, only qualitative or semi-quantitative data were compiled, without a direct enumeration of transplanted and surviving grafts. A last example would be for enumeration of kidney glomeruli in conjunction with the use of cationized ferritin as a contrast agent [97].

The general use of MRI-based cell tracking and this specific approach to quantifying this data has some limitations. Still, MRI of magnetically labeled cells only detects the iron, not the cell itself, and this method is still unable to distinguish live cells from dead cells. Further, if more than one cell generates a particular spot in the MRI, then the calculated cell number would be inaccurate. In this work, only 67% of spots were resultant from individual cells, the other 33% from 2 or 3 cells. It remains an open question as to how accurate an automated spot detection algorithm for MRI-based cell tracking needs to be in order to provide useful clinical information. However, we do not feel that heterogeneous magnetic cell labeling is a significant problem. Indeed, cells with more internalized iron would have darker and larger spots on MRI, while cells with less internalized iron would have lighter and smaller spots. However, our automated quantification algorithm can account for differences in spot size and intensity to compensate for heterogeneous cell labeling.

For future work, several different studies have been suggested at the end of Chap. 4 and Chap. 5. In addition to these, it will be interesting to explore the efficacy of the proposed approach using the ground truth obtained with histology. Such a ground truth can also be utilized to evaluate the labeling performance of a medical expert. Another interesting direction would be to utilize a hier-

archical classification approach where the different classifiers are exploited in multiple layers. A classifier in each layer can reject some of the spot candidates and transfer the other candidates to the next layer. This will allow classifiers in deeper layers to specialize in detecting highly challenging spot candidates. Further, obtaining high resolution MRI can be time-consuming. Therefore, another interesting direction of research would be to explore CNN architectures that can perform accurate mapping of low resolution MRI to a higher resolution.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893. IEEE, 2005.

[2] Aude Oliva and Antonio Torralba. Building the gist of a scene: The role of global image features in recognition. *Progress in Brain Research*, 155:23–36, 2006.

[3] Haitham Bou Ammar, Eric Eaton, Matthew E Taylor, Decebal Constantin Mocanu, Kurt Driessens, Gerhard Weiss, and Karl Tuyls. An automated measure of mdp similarity for transfer in reinforcement learning. In *Workshops at the AAAI Conference on Artificial Intelligence*, 2014.

[4] Clinical trial: Outcomes data of adipose stem cells to treat parkinson's disease, website https://clinicaltrials.gov/ct2/show/nct02184546 (first received: July 3, 2014 last updated: June 17, 2015 last verified: June 2015).

[5] Clinical trial: A study to evaluate the safety of neural stem cells in patients with parkinson's disease, website https://clinicaltrials.gov/ct2/show/nct02452723 (first received: May 18, 2015 last updated: March 10, 2016 last verified: February 2016).

[6] Clinical trial: Umbilical cord tissue-derived mesenchymal stem cells for rheumatoid arthritis, website https://clinicaltrials.gov/ct2/show/nct01985464 (first received: October 31, 2013 last updated: February 4, 2016 last verified: February 2016).

[7] Clinical trial: Cx611-0101, eascs intravenous administration to refractory rheumatoid arthritis patients, website https://clinicaltrials.gov/ct2/show/nct01663116 (first received: August 5, 2011 last updated: March 5, 2013 last verified: February 2013).

[8] Clinical trial: Evaluation of autologous mesenchymal stem cell transplantation (effects and side effects) in multiple sclerosis, website https://clinicaltrials.gov/ct2/show/nct01377870 (first received: June 19, 2011 last updated: April 24, 2014 last verified: August 2010).

[9] Clinical trial: Stem cell therapy for patients with multiple sclerosis failing alternate approved therapy- a randomized study, website https://clinicaltrials.gov/ct2/show/nct00273364 (first received: January 5, 2006 last updated: March 21, 2016 last verified: March 2016).

[10] Clinical trial: Pilot study of redirected autologous t cells engineered to contain humanized anti-cd19 in patients with relapsed or refractory cd19+ leukemia and lymphoma previously treated with cell therapy, website https://clinicaltrials.gov/ct2/show/nct02374333 (first received: February 23, 2015 last updated: February 23, 2016 last verified: February 2016).

[11] Clinical trial: Genetically modified t-cells in treating patients with recurrent or refractory malignant glioma, website https://clinicaltrials.gov/ct2/show/nct02208362.

[12] Jonathan R Slotkin, Kevin S Cahill, Suzanne A Tharin, and Erik M Shapiro. Cellular magnetic resonance imaging: nanometer and micrometer size particles for noninvasive cell localization. *Neurotherapeutics*, 4(3):428–433, 2007.

[13] Erik M Shapiro, Kathryn Sharer, Stanko Skrtic, and Alan P Koretsky. In vivo detection of single cells by mri. *Magnetic Resonance in Medicine*, 55(2):242–249, 2006.

[14] Rong Zhou, Djaudat Idiyatullin, Steen Moeller, Curt Corum, Hualei Zhang, Hui Qiao, Jia Zhong, and Michael Garwood. Swift detection of spio-labeled stem cells grafted in the myocardium. *Magnetic Resonance in Medicine*, 63(5):1154–1161, 2010.

[15] Yijen L Wu, Qing Ye, Danielle F Eytan, Li Liu, Bedda L Rosario, T Kevin Hitchens, Fang-Cheng Yeh, Chien Ho, et al. Magnetic resonance imaging investigation of macrophages in acute cardiac allograft rejection after heart transplantation. *Circulation: Cardiovascular Imaging*, 6(6):965–973, 2013.

[16] Ihor Smal, Marco Loog, Wiro Niessen, and Erik Meijering. Quantitative comparison of spot detection methods in fluorescence microscopy. *IEEE Transactions on Medical Imaging*, 29(2):282–301, 2010.

[17] Yuki Mori, Ting Chen, Tetsuya Fujisawa, Syoji Kobashi, Kohji Ohno, Shinichi Yoshida, Yoshiyuki Tago, Yutaka Komai, Yutaka Hata, and Yoshichika Yoshioka. From cartoon to real time mri: in vivo monitoring of phagocyte migration in mouse brain. *Scientific reports*, 4:6997, 2014.

[18] Muhammad Jamal Afridi, Xiaoming Liu, Erik Shapiro, and Arun Ross. Automatic in vivo cell detection in MRI. In *Medical Image Comuting and Medical Assisted Interventions*, pages 391–399. Springer, 2015.

[19] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lars Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[20] Muhammad Jamal Afridi, Xiaoming Liu, and J Mitchell McGrath. An automated system for plant-level disease rating in real fields. In *International Conference on Pattern Recognition*, pages 148–153, 2014.

[21] Alexander Sorokin and David Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition workshops*, 2008.

[22] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[23] Zhenguo Li, Xiao-Ming Wu, and ShihFu Chang. Segmentation using superpixels: A bipartite graph partitioning approach. In *Computer Vision and Pattern Recognition*, 2012.

[24] Zhihui Hao, Qiang Wang, Haibing Ren, Kuanhong Xu, Yeong Kyeong Seong, and Jiyeun Kim. Multiscale superpixel classification for tumor segmentation in breast ultrasound images. In *International Conference on Image Processing*, pages 2817–2820, 2012.

[25] Brian Fulkerson, Andrea Vedaldi, and Stefano Soatto. Class segmentation and object localization with superpixel neighborhoods. In *International Conference on Computer Vision*, 2009.

[26] Joseph Tighe and Svetlana Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Europearn Conference on Computer Vision*, pages 352–365, 2010.

[27] Han Liu, Yanyun Qu, Yang Wu, and Hanzi Wang. Class-specified segmentation with multiscale superpixels. In *Asian Conference on Computer Vision Workshops*, pages 158–169, 2013.

[28] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[29] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *advances in Neural Information Processing Ssystems*, pages 1097–1105, 2012.

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[31] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[32] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[36] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016.

[37] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

[38] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *International Conference on Computer Vision*, pages 37–45, 2015.

[39] Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Pose induction for novel object categories. In *International Conference on Computer Vision*, pages 64–72, 2015.

[40] Mingsheng Long, Jianmin Wang, Michael Jordan, and Yue Cao. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, 2015.

[41] Hao Chen, Dong Ni, Jing Qin, Shenli Li, Xin Yang, Tianfu Wang, and Pheng Heng. Standard plane localization in fetal ultrasound via domain transferred deep neural networks. *IEEE Journal of Biomedical and Health Informatics*, 19, 2015.

[42] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In *Computer Vision and Pattern Recognition workshop*, 2015.

[43] Etai Littwin and Lior Wolf. The multiverse loss for robust transfer learning. *Computer Vision and Pattern Recognition*, 2016.

[44] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 2013.

[45] Sachin Sudhakar Farfade, Mohammad J Saberian, and Li-Jia Li. Multi-view face detection using deep convolutional neural networks. In *International Conference on Multimedia Retrieval*, pages 643–650. ACM, 2015.

[46] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Computer Vision and Pattern Recognition*, pages 1717–1724, 2014.

[47] Hal Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.

[48] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application to early detection of fatigue. *Transactions on Knowledge Discovery from Data*, 6(4):18, 2012.

[49] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition*, pages 2066–2073. IEEE, 2012.

[50] John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, pages 120–128. ACM, 2006.

[51] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. Transfer learning for activity recognition: A survey. *Knowledge and Information Systems*, 36(3):537–556, 2013.

[52] Kyle D Feuz and Diane J Cook. Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (FSR). *Transactions on Intelligent Systems and Technology*, 6(1):3, 2015.

[53] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[54] Wenyuan Dai, Qiang Yang, GuiRong Xue, and Yong Yu. Boosting for transfer learning. In *International Conference on Machine Learning*, pages 193–200, 2007.

[55] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in NLP. In *ACL*, volume 7, pages 264–271, 2007.

[56] Xuejun Liao, Ya Xue, and Lawrence Carin. Logistic regression with an auxiliary data source. In *International Conference on Machine learning*, pages 505–512. ACM, 2005.

[57] Pengcheng Wu and Thomas G Dietterich. Improving svm accuracy by training on auxiliary data sources. In *International Conference on Machine Learning*, page 110. ACM, 2004.

[58] Sinno Jialin Pan, James T Kwok, and Qiang Yang. Transfer learning via dimensionality reduction. In *AAAI Conference on Artificial Intelligence*, volume 8, pages 677–682, 2008.

[59] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *Transactions on Neural Networks*, 22(2):199–210, 2011.

[60] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *International Conference on Machine learning*, pages 759–766. ACM, 2007.

[61] Tatiana Tommasi, Francesco Orabona, and Barbara Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Computer Vision and Pattern Recognition*, pages 3081–3088. IEEE, 2010.

[62] Yi Yao and Gianfranco Doretto. Boosting for transfer learning with multiple sources. In

Computer Vision and Pattern Recognition, pages 1855–1862. IEEE, 2010.

[63] Lilyana Mihalkova, Tuyen Huynh, and Raymond J Mooney. Mapping and revising markov logic networks for transfer learning. In *AAAI Conference on Artificial Intelligence*, volume 7, pages 608–614, 2007.

[64] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Self-taught clustering. In *International Conference on Machine Learning*, pages 200–207. ACM, 2008.

[65] Dikan Xing, Wenyuan Dai, Gui-Rong Xue, and Yong Yu. Bridged refinement for transfer learning. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 324–335. Springer, 2007.

[66] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *advances in Neural Information Processing Systems Workshop*, volume 898, 2005.

[67] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *International Conference on World Wide Web*, pages 751–760. ACM, 2010.

[68] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *International Conference on Computer Vision*, pages 2200–2207, 2013.

[69] Sebastian Thrun and Tom M Mitchell. Learning one more thing. Technical report, DTIC Document, 1994.

[70] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science and Business Media, 2012.

[71] Zhongqi Lu, Yin Zhu, Sinno Jialin Pan, Evan Wei Xiang, Yujing Wang, and Qiang Yang. Source free transfer learning for text classification. In *AAAI Conference on Artificial Intelligence*, pages 122–128, 2014.

[72] Muhammad Jamal Afridi, Arun Ross, and Erik M. Shapiro. L-CNN: Exploiting labeling latency in a CNN learning framework. In *International Conference on Pattern Recognition*, 2016.

[73] Muhammad Jamal Afridi, Arun Ross, Xioaming Liu, Margaret Bennewitz, Dorela Shuboni, and Erik M. Shapiro. Intelligent and automatic cell detection and quantification in MRI. Magnetic Resonance in Medicine.

[74] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *advances in Neural Information*

Processing Systems, pages 487–495, 2014.

[75] Dan Foti, Greg Hajcak, and Joseph Dien. Differentiating neural responses to emotional pictures: evidence from temporal-spatial PCA. *Psychophysiology*, 46(3):521–530, 2009.

[76] MD Grima Murcia, MA Lopez-Gordo, Maria J Ortíz, JM Ferrández, and Eduardo Fernández. Spatio-temporal dynamics of images with emotional bivalence. In *Artificial Computation in Biology and Medicine*, pages 203–212. Springer, 2015.

[77] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009.

[78] Vladimir Vapnik, Akshay Vashist, and Natalya Pavlovitch. Learning using hidden information (learning with teacher). In *International Joint Conference on Neural Networks*, pages 3188–3195. IEEE, 2009.

[79] Jixu Chen, Xiaoming Liu, and Siwei Lyu. Boosting with side information. In *Asian Conference on Computer Vision*, pages 563–577. Springer, 2012.

[80] Ziheng Wang and Qiang Ji. Classifier learning with hidden information. In *Computer Vision and Pattern Recognition*, pages 4969–4977, 2015.

[81] Viktoriia Sharmanska, Novi Quadrianto, and Christoph Lampert. Learning to rank using privileged information. In *International Conference on Computer Vision*, pages 825–832, 2013.

[82] Lior Wolf and Noga Levy. The svm-minus similarity score for video face recognition. In *Computer Vision and Pattern Recognition*, pages 3523–3530, 2013.

[83] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H Lampert. Learning to transfer privileged information. *arXiv preprint arXiv:1410.0389*, 2014.

[84] Walter J. Scheirer, Samuel E. Anthony, Ken Nakayama, and David D. Cox. Perceptual annotation: Measuring human vision to improve computer vision. *PAMI*, 36, August 2014.

[85] Eric P Xing, Andrew Y Ng, Michael I Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. *advances in Neural Information Processing Ssystems*, 15:505–512, 2003.

[86] Sugato Basu, Mikhail Bilenko, and Raymond J Mooney. A probabilistic framework for semi-supervised clustering. In *KDD*, pages 59–68. ACM, 2004.

[87] Herman Kamper, Weiran Wang, and Karen Livescu. Deep convolutional acoustic word embeddings using word-pair side information. *arXiv preprint arXiv:1510.01032*, 2015.

[88] Ruth Janning, Carlotta Schatten, and Lars Schmidt-Thieme. Hnnp-a hybrid neural network plait for improving image classification with additional side information. In *ICTAI*, pages 24–29. IEEE, 2013.

[89] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition*, pages 2097–2104, 2011.

[90] Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *Transactions on Pattern Analysis and Machine Intelligence*, 32(3):448–461, 2010.

[91] Remco R Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, and David Scuse. Weka manual for version 3-7-8. *Hamilton, New Zealand*, 2013.

[92] Yuki Mori, Ting Chen, Tetsuya Fujisawa, Syoji Kobashi, Kohji Ohno, Shinichi Yoshida, Yoshiyuki Tago, Yutaka Komai, Yutaka Hata, and Yoshichika Yoshioka. From cartoon to real time MRI: in vivo monitoring of phagocyte migration in mouse brain. *Scientific reports*, 4, 2014.

[93] Juan José Rodriguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630, 2006.

[94] Piotr Walczak, Jian Zhang, Assaf A Gilad, Dorota A Kedziorek, Jesus Ruiz-Cabello, Randell G Young, Mark F Pittenger, Peter CM van Zijl, Judy Huang, and Jeff WM Bulte. Dual-modality monitoring of targeted intraarterial delivery of mesenchymal stem cells after transient ischemia. *Stroke*, 39(5):1569–1574, 2008.

[95] Dian R Arifin, Steffi Valdeig, Robert A Anders, Jeff WM Bulte, and Clifford R Weiss. Magnetoencapsulated human islets xenotransplanted into swine: a comparison of different transplantation sites. *Xenotransplantation*, 23(3):211–221, 2016.

[96] Ping Wang, Christian Schuetz, Prashanth Vallabhajosyula, Zdravka Medarova, Aseda Tena, Lingling Wei, Kazuhiko Yamada, Shaoping Deng, James F Markmann, David H Sachs, et al. Monitoring of allogeneic islet grafts in nonhuman primates using mri. *Transplantation*, 99(8):1574–1581, 2015.

[97] Edwin J Baldelomar, Jennifer R Charlton, Scott C Beeman, Bradley D Hann, Luise Cullen-McEwen, Valeria M Pearl, John F Bertram, Teresa Wu, Min Zhang, and Kevin M Bennett. Phenotyping by magnetic resonance imaging nondestructively measures glomerular number and volume distribution in mice with and without nephron reduction. *Kidney international*, 2015.