

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

3D Convolutional Neural Network for Liver Tumor Segmentation

Pedro Diogo da Cunha Amorim



Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. João Manuel R. S. Tavares

July 26, 2018

Abstract

Liver cancer is the second most dangerous cancer in the world. Most liver segmentations of Computer Tomography scans are still manually done by medical experts, contributing for longer periods of analysis. Automatic segmentation of the liver and hepatic lesions is an important step towards computer-aided decision support systems. This type of application can produce earlier and more systematic clinical diagnosis, helping medical experts in their decision making, and thus resulting in patients getting earlier prognostics.

As an emerging Computer Vision field, Deep Learning helped define Medical Image Segmentation and Classification, outperforming most other algorithms in many medical challenges, especially with the rise of Convolutional Neural Networks (CNNs). Also, preprocessing a dataset before training is not a trivial step, albeit a very important one when accounting for final results.

In this dissertation, a detailed review on Neural Networks applied to Computer Vision is provided. Also, Volumetric Convolutional Neural Networks are introduced, and proper dataset preprocessing is discussed. Finally, a 3D CNN architecture, V-Net, is implemented and its results analyzed.

Acknowledgments

Não existe mais nobre ensinamento do que as amizades que levo para a vida.

À Faculdade de Engenharia da Universidade do Porto, que foi a minha casa nos últimos 5 anos, deixando em mim parte do seu enorme legado.

Ao professor João Manuel Ribeiro da Silva Tavares, pela orientação prestada durante este projeto.

À Tuna de Engenharia da Universidade do Porto, família cimentada pela amizade, onde aprendi o valor do respeito, palavra, e dedicação.

Aos meus pais, Joaquim Amorim e Goreti Cunha, pela apoio incondicional, e pelos valores que me transmitiram.

À minha família, pelo transparente orgulho que sempre em mim depositaram.

À minha namorada, Luana Valente, por me ensinar o valor da paciência.

A todos os meus amigos, em especial Daniel Almeida, Patrick Sousa, Francisco Próspero, Pedro Abrunhosa, Pedro Neto, Matheus Belucio, João Morgado, e Rui Roxo, pela sua amizade, apoio e motivação.

Diogo Amorim

*“Practice does not make perfect.
Only perfect practice makes perfect.”*

Vince Lombardi

Contents

| | |
|--|------------|
| Abstract | i |
| Acknowledgments | iii |
| Abbreviations | xi |
| 1 Introduction | 1 |
| 1.1 Problem Specification | 1 |
| 1.2 Methodology | 2 |
| 1.3 Research Questions and Thesis | 3 |
| 2 Overview on Computer Vision | 5 |
| 2.1 Neural Network Fundamentals | 5 |
| 2.1.1 Deep Learning | 5 |
| 2.1.2 Implementation problems | 6 |
| 2.2 Convolutional Neural Networks | 7 |
| 2.3 Implementation Tools | 8 |
| 2.4 Literature Review on Convolutional Neural Networks | 8 |
| 2.4.1 Design Considerations in a CNN | 8 |
| 3 3D Implementation of a UNET Based CNN | 11 |
| 3.1 Network Architecture | 11 |
| 3.1.1 U-Net | 11 |
| 3.1.2 V-Net | 12 |
| 3.1.3 Implementation | 12 |
| 3.2 Dataset | 13 |
| 3.2.1 LiTS Dataset | 13 |
| 3.2.2 Preprocessing | 13 |
| 3.3 Results | 16 |
| 3.4 Summary | 17 |
| 4 Conclusion | 19 |
| 4.1 Contributions | 19 |
| 4.2 Future Work | 19 |
| References | 21 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | CT Volume. | 2 |
| 1.2 | Left, Middle: Abdominal liver scan example. Right: Liver/lesion segmentation. (adapted from [31]) | 2 |
| 2.1 | General Regression Neural Network. (adapted from [30]) | 6 |
| 2.2 | ImageNet dataset sample of mammal and vehicle subtrees. (adapted from [8]) . . | 7 |
| 2.3 | Feature visualization from an ImageNet trained CNN. (adapted from [40]) | 8 |
| 3.1 | U-Net: Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. (adapted from [27]) | 12 |
| 3.2 | V-Net (adapted from [23]) | 13 |
| 3.3 | Left: CT Volume. Right: Liver segmentation with two lesions (right) | 14 |
| 3.4 | Slices of an original CT along the z axis. | 14 |
| 3.5 | Left: Original CT. Right: "Windowed CT". | 15 |
| 3.6 | Left: Three raw slices. Right: Interpolated result. | 16 |
| 3.7 | Left: Three raw segmentation slices. Right: Interpolated result. | 16 |

Abbreviations

| | |
|-------|--|
| 2D | Two-Dimensional |
| 3D | Three-Dimensional |
| API | Application Programming Interface |
| BRaTS | Multimodal Brain Tumor Segmentation Challenge 2017 |
| CNN | Convolutional Neural Network |
| CPU | Central Processing Unit |
| CT | Computational Tomography |
| GPU | Graphics Processing Unit |
| HU | Hounsfield Units |
| LiTS | Liver Tumor Segmentation Challenge |
| NN | Neural Network |
| ROI | Region of Interest |

Chapter 1

Introduction

In this chapter, the thesis "3D Convolutional Neural Network for Liver Tumor Segmentation" is introduced.

In section 1.1, liver tumor segmentation is contextualized. In section 1.2, the methodology to tackle this problem is presented as this thesis structure. In section 1.3, the fundamental questions for this thesis are raised.

1.1 Problem Specification

Liver cancer has the second largest mortality rate of any cancer, as is the sixth most common worldwide [10, 32]. Despite this fact, manual measurement is still the most frequent practice in determining tumor shape and size in clinical practice. Thus, accurate automatic liver and lesion segmentation can assist doctors evaluate and plan treatment, while removing a time-consuming process that may increase the patients quality of life.

Computer Vision is a field that has tremendous value in tumor detection. However, traditional methods in liver segmentation have not achieved the same accuracy when compared to other modalities. This is due to liver not being homogeneous in size, shape, and position from subject to subject. Also, in a Computer Tomography scan, contrast between liver and neighboring organs is very poor, and radiologists usually use an injection protocol to enhance tumor visibility, as seen in figure 1.1. This may increase the overall noise in the liver region, and make analysis more complex [24]. Still, computer based approaches make prediction methods fast, repeatable and deterministic. This removes human subjectivity, which helps doctors and researchers compare and discuss results.

Liver Tumor Segmentation is a difficult problem of Computer Tomography (CT) [20]. Segmentation is the process of identifying clusters of information in an image and successfully separate it from the remaining data. There are several algorithms to achieve it, and until recent years, most focused on either using data augmentation to filter noise and enhance image features [38], or analyzing an image part by part using windowing methods [21], and ultimately merging both [2, 29].

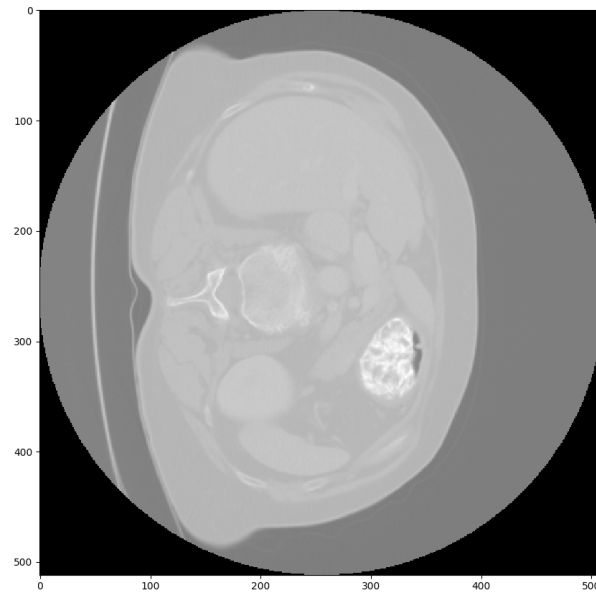


Figure 1.1: CT Volume.

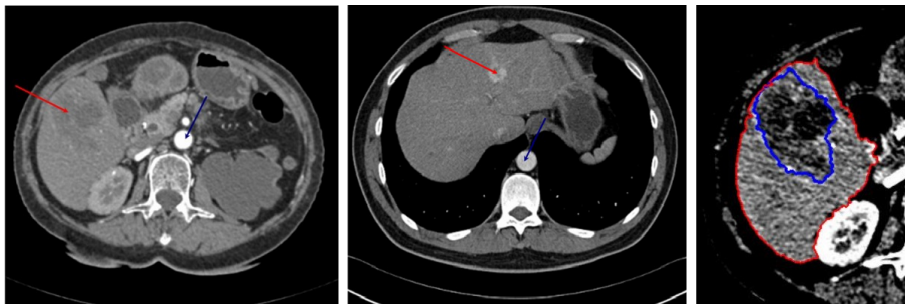


Figure 1.2: **Left, Middle:** Abdominal liver scan example. **Right:** Liver/lesion segmentation. (adapted from [31])

When a successful segmentation is made, the segmented area is called a Region of Interest (ROI), as seen in figure 1.2, which for the purpose of this work will be both the liver, in red, and its respective tumor in blue. The resulting segmentations combined output a 3D model of the liver and its affected tissue, which are undeniably helpful for medical professionals.

1.2 Methodology

In recent years, interest in image segmentation and classification algorithms has risen, as well as the technology to compute them. Nonetheless, another approach to this problem has had a revolutionary impact in the development of Computer Vision solutions: Deep Learning [17]. Either in commercial applications (music preference prediction [34], self driving vehicles [14]), or research

(real time humane pose estimation [5], natural language processing [7]), it has proven itself as a versatile and efficient technology in any context.

In the medical field, deep learning has already achieved surprising results [19, 33]. However, not much has been done for liver tumor segmentation. Current state of the art is attributed to 2D Convolution Neural Networks (CNNs). These are known to ignore volumetric information along the third dimension, as they process Computer Tomographies (CTs) scan slice by slice, essentially meshing individual 2D segmentations together. This project aims to implement a state of the art V-Net based architecture [23], which was extensively used in the Multimodal Brain Tumor Segmentation Challenge 2017 (BRaTS) [22] with good results, and adapt it for liver lesions.

The development process of this dissertation is as follows:

1. **Overview on Computer Vision** (see chapter 2)

In this chapter, all the necessary research for this work is presented, including Computer Vision, Medical Image Segmentation, and Convolutional Neural Networks, with a detailed NN design analysis. A review of the State of the Art is also provided.

2. **3D Implementation of a UNET Based CNN** (see chapter 3)

In this chapter, the CNN foundations will be built upon. It is explained how to extrapolate conventional 2D implementation into 3D. Also present is a description the dataset used, and its preprocessing methods, with correspondent results.

1.3 Research Questions and Thesis

- How can data be preprocessed to improve final results?
- What design choices should be considered when implementing a 3D CNN?
- What is the compromise between 2D and 3D architectures?
- Can 3D CNNs improve performance in Liver Segmentation?

Thesis: 3D Convolutional neural networks outperform 2D counterparts.

Chapter 2

Overview on Computer Vision

In section 2.1 of this chapter, an introduction on Neural Networks is provided, along with how they influenced Image Segmentation in the past decade. In section 2.2, further explanation on how Convolutional Neural Networks became a good alternative to older Vision algorithms can be seen. A description of the Tools being used in this work will also be represented (section 2.3), concluding on a Review of this chapter (section 2.4).

2.1 Neural Network Fundamentals

2.1.1 Deep Learning

For decades, Computer Vision's state of the art was mostly attributed to scripted algorithms [35, 39]. Most focused on human intervention, such as Beasley's work in Semiautonomous Seeded Edge Detection [3], that required a specialist to point out areas of interest, which the algorithm then fine-tunes. Others tried to go one step beyond, removing human intervention altogether, using, for example, genetic algorithms which are based on biological evolutionary patterns [26]. Even though they achieved good results, these were not suitable for generalization problems, as these types of algorithms were computationally heavy, and usually were too fine-tuned for a specific problem. The answer lied within Deep Learning.

Many attempts were made in the past to use Deep Learning technics as a fully autonomous answer for Image Classification. The first used Multilayer Perceptrons, also known as the original form of Neural Networks (NN), were first introduced in 1956 [15]. This type of network, inspired in neural communication, has individual "neurons", arranged in layers, and connected to every neuron in the previous layer (figure 2.1). Every neuron has a weight associated to every connection w_i , and a bias b . After receiving the outputs, x , from the previous layer, it computes a new value according equation 2.1, where K is an activation function designed to induce non-linearity in the system, further improving the expected results results for complex problems.

$$f(x) = K \left(\sum_i w_i x_i + b \right) \quad (2.1)$$

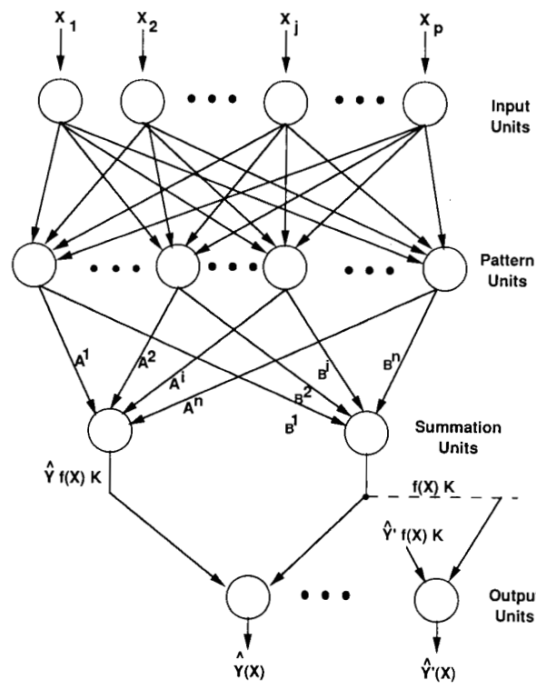


Figure 2.1: General Regression Neural Network. (adapted from [30])

The final step needed was to find a way for this NN to improve its results autonomously, instead of fine-tuning the weights manually, a time consuming approach which was only feasible for very small networks. This development became a reality with Werbos' model for regression [36]. By computing the error in the output of the network, it was possible to adjust each individual weight layer by layer in order to minimize a loss function. Thus, these algorithms could "learn" from previous events and predict future ones [30].

2.1.2 Implementation problems

Despite Deep Learning's immense potential, two major factors may be responsible for its late adoption:

- **Lack of proper Datasets** - Proper datasets are one of the most important factors in deep learning techniques [17]. For image segmentation, pictures have to be properly sized, contain enough variation among them so that the model being trained does not overfit [12], and be in sizable numbers to be a representative sample. Also, for many years storage was limited, and the Internet was not prepared for large data transfers, which made it difficult for researchers to share datasets. Digital cameras were not common either, neither did they produce pictures with sufficient quality for proper feature extraction.
- **Slow Hardware Development** - Because general regression is an intensive process, full image analysis using Deep Learning remained difficult to implement until recently. Lack of

significant GPU and CPU improvements favored traditional methods of image processing, which demanded far less computational power. Also, low dedicated memory further increased the amount of data processed at each time, which resulted in longer training times.

These were also useful for their parallel processing capabilities, which further enhanced the segmentation processing time, after being properly trained through hundreds of iterations.

2.2 Convolutional Neural Networks

As discussed above, large datasets were hard to come by. In 2009, ImageNet [8] was presented as a solution to this problem. This project aimed to help the scientific community have a centralized dataset which could be a benchmark for new Computer Vision applications. It contained 3.2 million full resolution images, subdivided in multiple trees, and could be used for a variety of problems, including image classification, object detection and group clustering. With this new dataset, an annual classification challenge accompanied it. Many different approaches were taken, until, in 2012, a Convolutional Neural Network (CNN) called AlexNet [16] won by a significant lead.

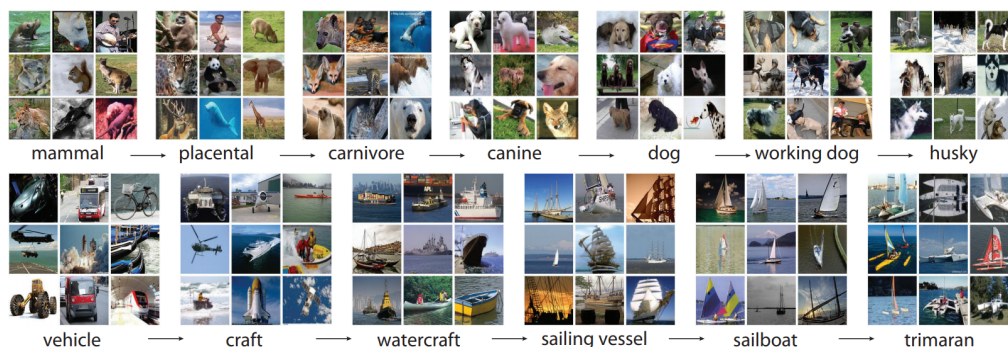


Figure 2.2: ImageNet dataset sample of mammal and vehicle subtrees. (adapted from [8])

In traditional Computer Vision, most of the work consists on hand-engineering filters which, when applied to an image, can extract its features [11]. The more features can be extracted, the more accurate a prediction is. A major setback in this approach is that each feature must be manually engineered in the design process, which makes scaling these types of algorithms hard. Convolutional Neural Networks work in the opposite way. Instead, one chooses how many features the CNN will extract, and during its training, it will automatically extract them [18]. These can be extremely abstract to the human eye, as seen in figure 2.3, but very accurate in these circumstances, which is one of the main advantages of CNN architectures.

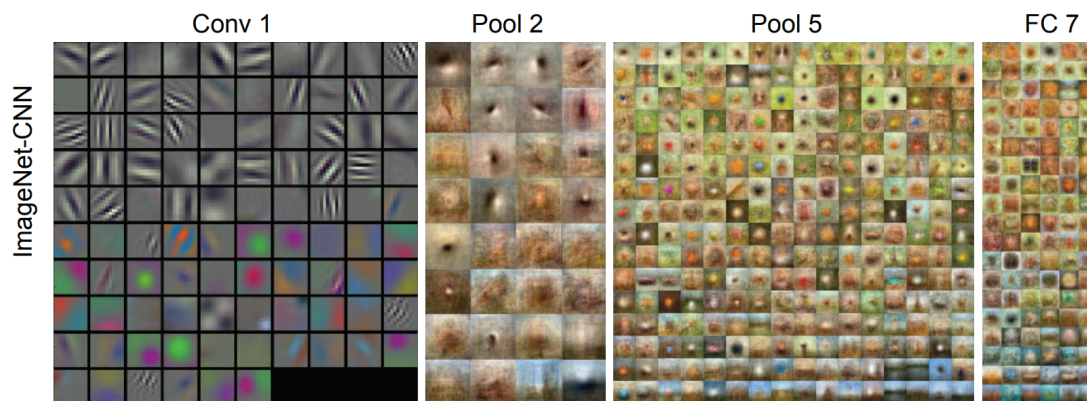


Figure 2.3: Feature visualization from an ImageNet trained CNN. (adapted from [40])

2.3 Implementation Tools

For this project, a computer equipped with two Nvidia GeForce GTX 1080 GPUs, and one Nvidia Titan X was used. As for software, all code was written in Python, using the following tools:

- **CUDA** - CUDA is a parallel computing platform, where developers can increase computing applications by having full control of a Nvidia GPU's power [25]. While the sequential part of the workload runs on a CPU, multi-threaded operations are instead computed on thousands of GPU cores in parallel, optimizing workflow.
- **Tensorflow** - TensorFlow is an open source software library for high performance numerical computation, which offers strong support not only for deep learning, but also many other scientific domains, with its flexible numerical computation core [1].
- **Keras** - Keras is a high-level Neural Networks API, written in Python, and capable of running on top of Tensorflow. It features an user friendly, modular, high level approach to writing code for NNs, thus being the reason it was chosen for this project. [6]
- **Nibabel** - Nibabel is a Python package which can read Nifti files and convert them to data arrays. [4].

2.4 Literature Review on Convolutional Neural Networks

2.4.1 Design Considerations in a CNN

Nowadays, most NNs core code is already written and optimized in libraries like Theano, Coffee, or Tensorflow. Improvements in performance are, in essence, dependent on two factors: the architecture's design, and the hardware it is implemented in. Design is usually restricted by its performance cost, and restricted by lack of memory, or computing power. A description of such compromises will be presented in this subsection.

2.4.1.1 Architectural Configuration

There is a certain debate of which is better regarding performance: a wider, or a deeper network?

- **Depth** - Zagoruyko and Komodakis [37] suggest a wide enough "shallow" network can learn any function given enough data, while performing faster than traditional methods. They also claim that deepening a NN by increasing the number of layers has diminishing returns, whereas performance gains with wide NNs are more linear. However, these types of networks are better at memorizing, and thus, generalization is a problem.
- **Width** - He et al. [13] have a different understanding on this matter. Their findings point out that although diminishing returns are expected when going deeper and increasing the number of layers, by reducing the width of each layer and using a residual model, these models avoid degradation issues, and can outperform their peers. Also, Elden and Shamir [9] have mathematically proven that going deeper has less complexity while exponentially more performance gains for NNs.

Ultimately, even though there are certain configurations that outperform their counterparts, it comes down to the problem in hand. Different NN's have different applications to extract their full potential, and there is still much progress made through empirical evidence.

Chapter 3

3D Implementation of a UNET Based CNN

In section 3.1, this project's architecture is presented. Then, in section 3.2 the is introduced, and an explanation of how it was preprocessed is provided. Finally, in section 3.3, the results are discussed.

3.1 Network Architecture

This project aims to implement a 3D CNN for liver tumor segmentation, and compare its results with a 2D approach. As such, U-Net is used as a 2D CNN benchmark (see subsection 3.1.1), and a V-Net implementation is used for a volumetric approach (see subsection 3.1.2).

3.1.1 U-Net

Image segmentation using traditional sliding-window CNNs require huge amounts of data to achieve proper results. In medical research, datasets are hard to obtain, and they are not always uniform. Ronneberger et al. designed a CNN to adress this very issue: U-Net [27].

U-Net is the most popular CNN architecture for medical image analysis, optimized for data augmentation in order to predict segmentation masks without requiring a large amount of training data. This is especially useful, as medical data is hard to obtain. Many medical applications have used U-Net based architectures to perform modality segmentation, achieving very good results.

As seen in figure 3.1, the U-Net architecture features a contracting path of 2D convolutions to extract context, and a symmetric expanding path which performs deconvolution to retrieve the abstract information and rescale it to the input's size. Its output shape matches the input's, returning a mask that segments the original image. In the case of medical images, each slice is used as a 2D input one by one until the entire volume has been cycled through.

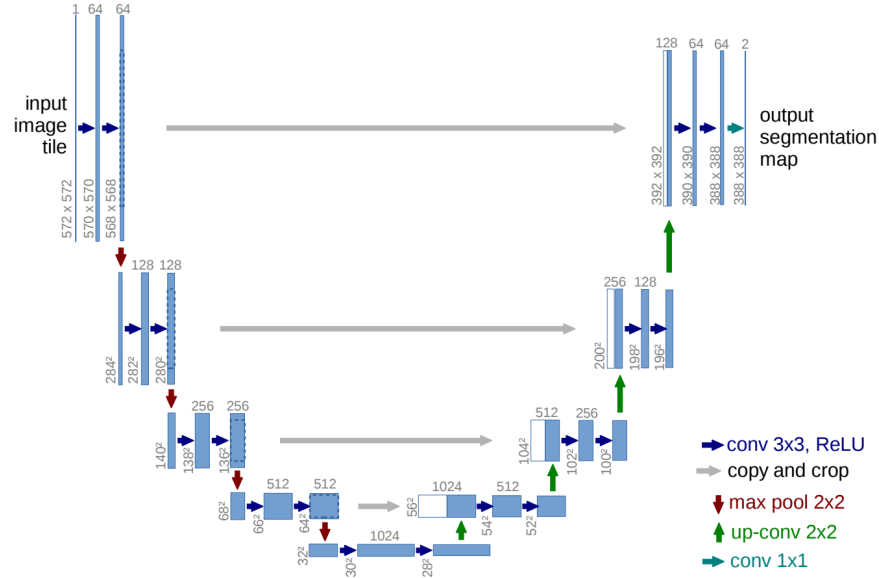


Figure 3.1: **U-Net**: Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. (adapted from [27])

3.1.2 V-Net

U-Net based CNNs have already proven to be a good solution for liver tumor segmentation. Despite this, they have a fundamental design flaw: the lack of volumetric context, as they ignore differences along the z axis. This essentially means that each slice is segmented as an individual image, which could make it harder to segment smaller patches, like a tumor, and reduce performance. Özgün et al. address this issue with V-Net [23]. V-Net is an architecture based on U-Net, where instead of using an image as input, it expects a volumetric shape.

3.1.3 Implementation

This thesis used a V-Net based CNN with the following parameters: Input shapes should be in powers of 2, which are better handled by the algebra optimization libraries. A $128 \times 128 \times 128$ input shape was the biggest our computer could handle without using all available memory. 5 layers were used for depth. 16 base filters for feature extraction were used in the first layer, which are doubled for each layer of convolution they pass through. An initial learning rate = 0.0005 was used. A batch size = 1 was the only option, as when compared to U-Net, we are using 128 different slices with resized resolution 128×128 as input, instead of one with full resolution 512×512 previously used for LiTS (almost tenfold as many pixels). Adam optimizer and Dice loss were also present, and the network trained for a maximum of 500 epochs.

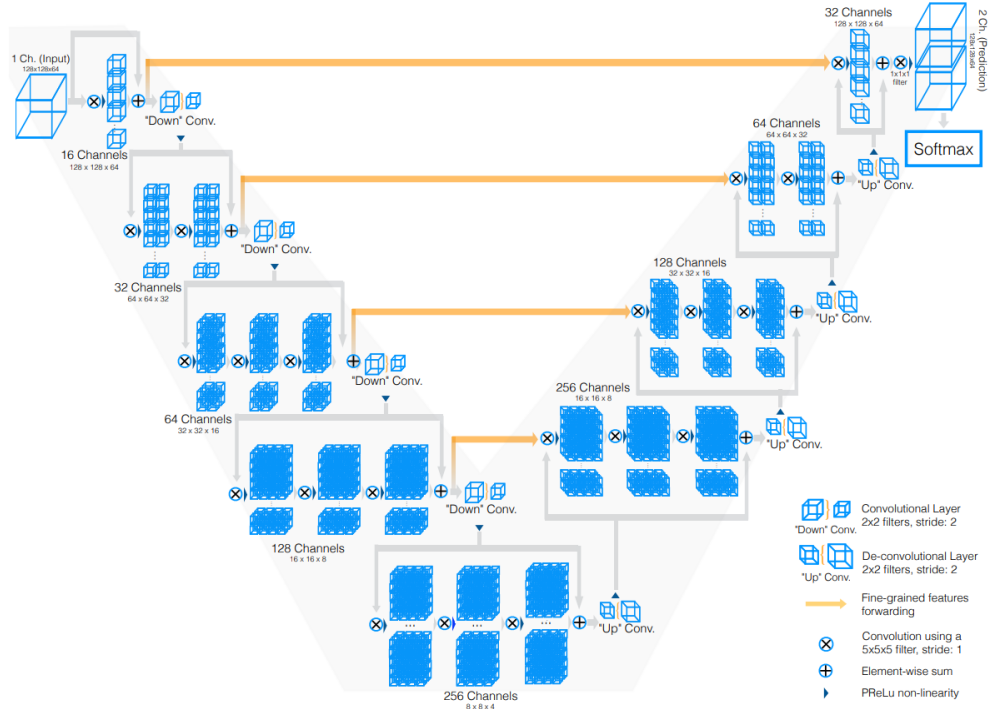


Figure 3.2: V-Net (adapted from [23])

3.2 Dataset

3.2.1 LiTS Dataset

The ISBI 2017 Liver Tumor Segmentation Challenge Challenge¹ (LiTS) dataset was used for this project. It comprises 200 contrast-enhanced abdomen CT scans in the Nifti format, from 6 medical centers, with 130 meant for training, and 70 for further testing. Each scan, represented by a volume, has a correspondent segmentation file as ground truth, containing a rough liver segmentation as Label 1, and a correspondent expert's lesion segmentation as Label 2, as seen in figure 3.3.

Every CT scan is also three-dimensional, each with a fixed 512x512 pixel resolution in the (x, y) plane. However, the z plane, associated to the number of layers, varies from scan to scan, some having as low as 75 layers, while others can surpass 800. This and other problems must be accounted before using this dataset as input to the CNN architecture.

3.2.2 Preprocessing

Although CNNs require significantly less image preprocessing than traditional methods, this is still an essential task that can improve training results. Bellow is an ordered description of each method used during this phase.

¹<https://competitions.codalab.org/competitions/17094>

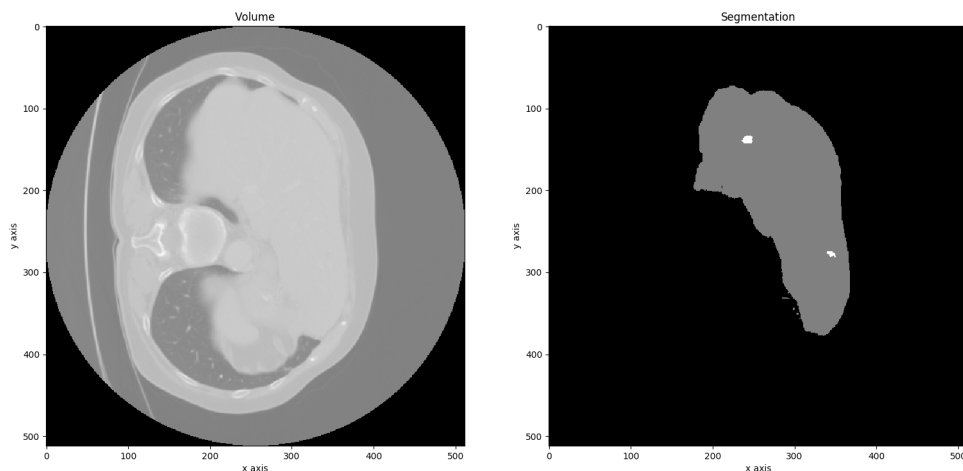


Figure 3.3: **Left:** CT Volume. **Right:** Liver segmentation with two lesions (right)

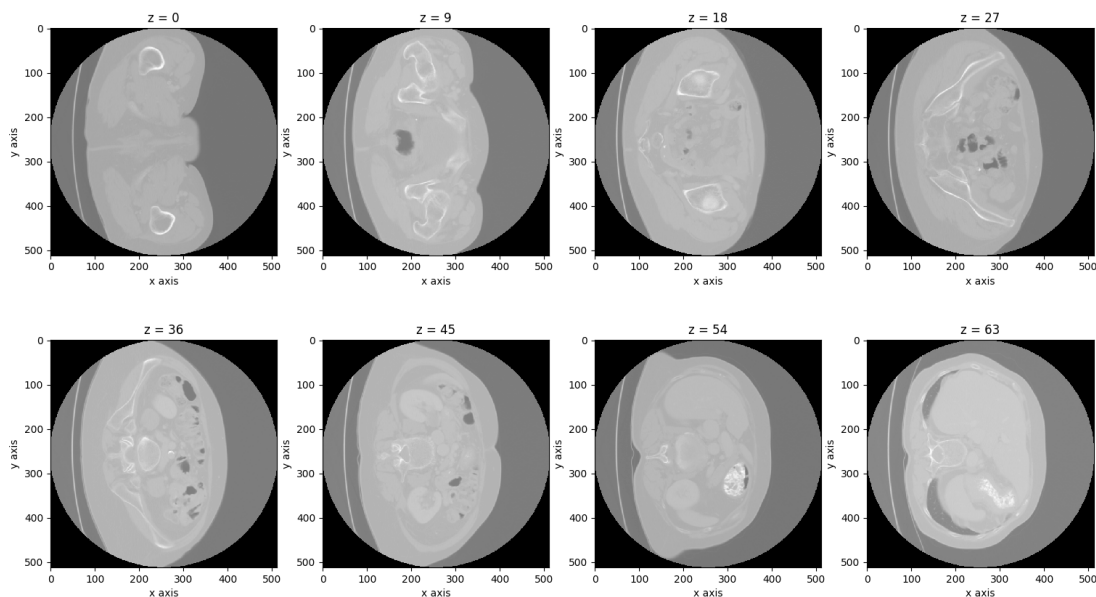


Figure 3.4: Slices of an original CT along the z axis.

3.2.2.1 Cropping

When a dataset is rescaled it is usually compressed, due to memory limitations while training. To prevent significant loss of liver and tumor information, slices which did not contain any segmentation were cropped. This ensures that segmentation data after rescaling will be as dense as possible.

3.2.2.2 Noise Filtering

These volumes are comprised of the raw data of each medical center, and as such, may contain noise. To filter such undesirable information, two technics are used:

- **CT Windowing**

Windowing is an important procedure in a CT scan. Pixel intensity, measured in Hounsfield Units (HU), is a function of the radiodensity, μ , of a substance, as seen in equation 3.1.

$$HU = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water} - \mu_{air}} \quad (3.1)$$

Raw data from the original CT scan has a pixel intensity range around the thousands of HU. However, the value of the "Liver Window" [-62, 238], as given by Sahi et al. [28], is significantly lower. As such, the CT scans range was set based on this new range, less 100 HU to account for all the liver lesions which have lower values, with a final value of [-160, 240].

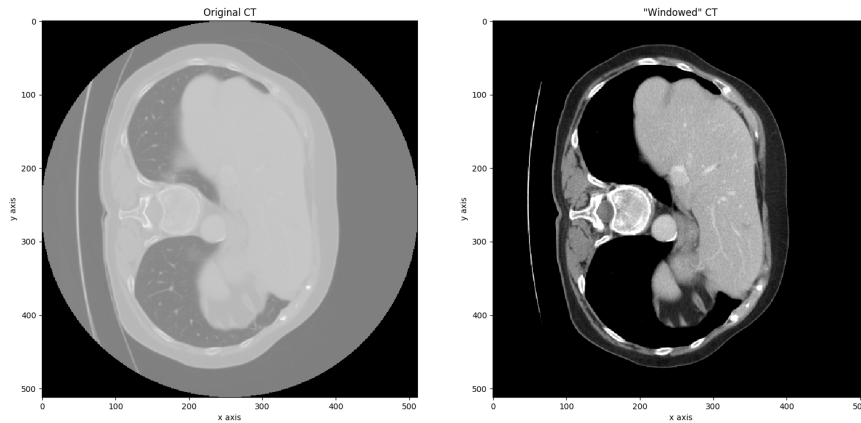


Figure 3.5: **Left:** Original CT. **Right:** "Windowed CT".

In figure 3.5, we can already see significant results. The image became more defined, and the liver and its lesions got highlighted and are more easily identified, as it enhances contrast in soft tissue.

- **Normalization**

When training a NN, weights are multiplied and biases are added to the initial inputs, which are then used for backpropagation. In order to prevent large fluctuations in gradient calculation during this phase, it is advisable to normalize data in preprocessing. To do so, the dataset's mean was subtracted, and its standard deviation was divided. In a statistical sense, normalization removes variation that is believed to be non-causal in prediction of the output. Thus, this variation cannot be used as a predictor by the NN.

3.2.2.3 Rescaling

Each CT scan has a different number of slices by default. A CNN needs a fixed input shape, and so there is a need for rescaling the data. The volumes were resized to (128, 128, 128) shape. Two methods of interpolation were used for this step. Also, the data was rotated 90° in the x-y plane to facilitate viewing.

- **Volumes** - Bilinear interpolation was used, as it is fast to compute and outputs good results (figure 3.6)

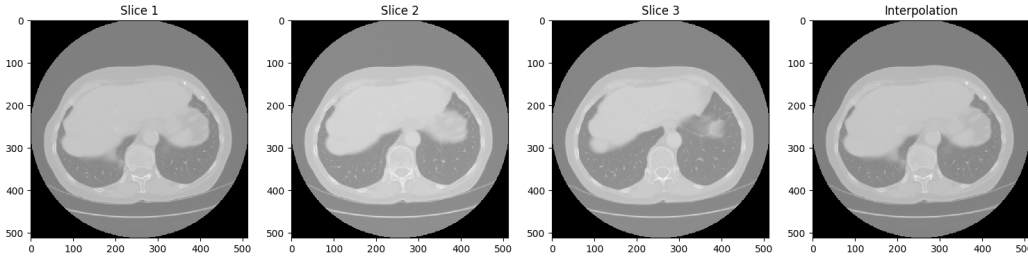


Figure 3.6: **Left:** Three raw slices. **Right:** Interpolated result.

- **Segmentations** - Nearest neighbor interpolation is the only option, as no other preserves the original images pixel values. This ensures that no segmentation slice has non integer values, in this case, 0 for background, 1 for liver, and 2 for lesion (figure 3.7).

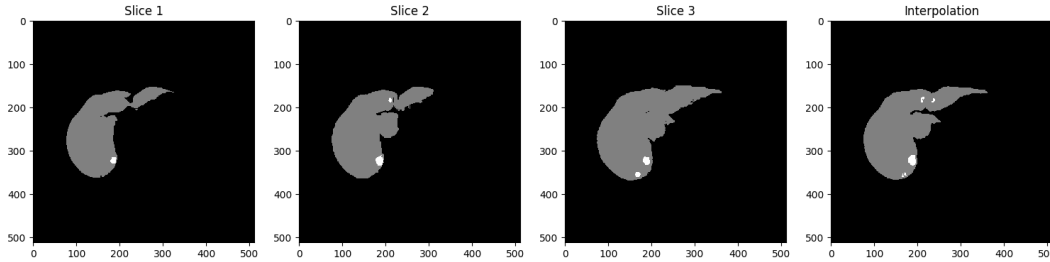


Figure 3.7: **Left:** Three raw segmentation slices. **Right:** Interpolated result.

3.3 Results

Tridimensional processing of a CT scan is very computationally expensive. Because the largest we could rescale the dataset without running out of memory was $128 \times 128 \times 128$ pixels, a quarter of the original resolution, many details were lost. This unfortunately means that Volumetric CNNs

still can not compete with a normal 2D approach, assuming they use full resolution volumes. Nonetheless when comparing results with 2D CNNs using the same downscaled dataset, 3D CNNs outperform them. This is evidence of the importance of the z axis when segmenting CT scans.

3.4 Summary

Plain and Tridimensional CNN architectures are compared, and it can be inferred that while currently 2D methods have the best results, while being faster and more efficient, 3D CNNs can be even more powerful once the hardware necessary to run them at higher resolutions is available.

Chapter 4

Conclusion

In this chapter, this project's contributions are in section [4.1](#), and future work in section [4.2](#).

4.1 Contributions

In this dissertation, a new approach to the hepatic segmentation was discussed. Someone unfamiliar with the problem in hand, medical image segmentation, can grasp its roots, as well as understand how research evolved around it and what may be done in the future using 3D CNNs. This includes:

- Understand the Medical Image Segmentation problem
- Learning Neural Network fundamentals, and what and how a CNN is designed
- Learning important preprocessing techniques regarding medical datasets

Also, for researchers of this field, Volumetric CNNs are introduced and empirical evidence in their favor is shown, hopefully helping this unorthodox method of Deep Learning get more deserved attention.

4.2 Future Work

In the future, 3D Convolutional Neural Networks may gain an advantage once hardware gets more powerful. For now, optimization is key to better run this types of algorithms. Once reliable, 3D CNN's can help many fields, even outside the medical realm.

References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.
- [3] Ryan A Beasley. Semiautonomous medical image segmentation using seeded cellular automaton plus edge detector. *ISRN Signal Processing*, 2012, 2012.
- [4] Matthew Brett, Michael Hanke, Marc-Alexandre Côté, Chris Markiewicz, Satrajit Ghosh, Demian Wassermann, Stephan Gerhard, Eric Larson, Gregory R. Lee, Yaroslav Halchenko, Erik Kastman, Cindee M, Félix C. Morency, moloney, Ariel Rokem, Michiel Cottaar, Jarrod Millman, jaeilepp, Alexandre Gramfort, Robert D Vincent, Paul McCarthy, Jasper J.F. van den Bosch, Krish Subramaniam, Nolan Nichols, embaker, markhymers, chaselgrove, Basile, Nikolaas N. Oosterhof, and Ian Nimmo-Smith. nipy/nibabel: 2.2.0, October 2017.
- [5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, volume 1, page 7, 2017.
- [6] François Chollet et al. Keras. <https://keras.io>, 2015.
- [7] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.
- [8] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.
- [9] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *Conference on Learning Theory*, pages 907–940, 2016.
- [10] Jacques Ferlay, Isabelle Soerjomataram, Rajesh Dikshit, Sultan Eser, Colin Mathers, Marise Rebelo, Donald Maxwell Parkin, David Forman, and Freddie Bray. Cancer incidence and mortality worldwide: sources, methods and major patterns in globocan 2012. *International journal of cancer*, 136(5), 2015.

- [11] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.
- [12] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [14] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015.
- [15] S. C. Kleene. Representation of events in nerve nets and finite automata. In Claude Shannon and John McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, Princeton, NJ, 1956.
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [18] 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.
- [19] Siqi Liu, Sidong Liu, Weidong Cai, Sonia Pujol, Ron Kikinis, and Dagan Feng. Early diagnosis of alzheimer’s disease with deep learning. In *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, pages 1015–1018. IEEE, 2014.
- [20] Suhuai Luo, Xuechen Li, and Jiaming Li. Review on the methods of automatic liver segmentation from abdominal images. *Journal of Computer and Communications*, 2(02):1, 2014.
- [21] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *International journal of computer vision*, 43(1):7–27, 2001.
- [22] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2015.
- [23] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 565–571. IEEE, 2016.
- [24] Mehrdad Moghbel, Syamsiah Mashohor, Rozi Mahmud, and M Iqbal Bin Saripan. Review of liver segmentation and computer assisted detection/diagnosis methods in computed tomography. *Artificial Intelligence Review*, pages 1–41.
- [25] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6(2):40–53, March 2008.

- [26] Mantas Paulinas and Andrius Ušinskas. A survey of genetic algorithms applications for image enhancement and segmentation. *Information Technology and control*, 36(3), 2007.
- [27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [28] Kamal Sahi, Stuart Jackson, Edward Wiebe, Gavin Armstrong, Sean Winters, Ronald Moore, and Gavin Low. The value of “liver windows” settings in the detection of small renal cell carcinomas on unenhanced computed tomography. *Canadian Association of Radiologists Journal*, 65(1):71–76, 2014.
- [29] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
- [30] Donald F Specht. A general regression neural network. *IEEE transactions on neural networks*, 2(6):568–576, 1991.
- [31] Jean Stawiaski, Etienne Decencière, and Francois Bidault. Interactive liver tumor segmentation using graph-cuts and watershed. 01 2008.
- [32] BWKP Stewart, Christopher P Wild, et al. World cancer report 2014. *Health*, 2017.
- [33] Wenqing Sun, Bin Zheng, and Wei Qian. Computer aided lung cancer diagnosis with deep learning algorithms. In *Medical Imaging 2016: Computer-Aided Diagnosis*, volume 9785, page 97850Z. International Society for Optics and Photonics, 2016.
- [34] A. Van Den Oord, S. Dieleman, and B. Schrauwen. Deep content-based music recommendation. 2013. cited By 148.
- [35] Arthur R Weeks. *Fundamentals of electronic image processing*. SPIE Optical Engineering Press Bellingham, 1996.
- [36] Paul Werbos and Paul J. (Paul John. Beyond regression : new tools for prediction and analysis in the behavioral sciences /. 01 1974.
- [37] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- [38] Jun Zhang and Jinglu Hu. Image segmentation based on 2d otsu method with histogram analysis. In *Computer Science and Software Engineering, 2008 International Conference on*, volume 6, pages 105–108. IEEE, 2008.
- [39] Yu Jin Zhang. A review of recent evaluation methods for image segmentation. In *Signal Processing and its Applications, Sixth International, Symposium on. 2001*, volume 1, pages 148–151. IEEE, 2001.
- [40] 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.