

Detection of Wheezes and Breathing Phases using Deep Convolutional Neural Networks

Johan Fredrik Eggen Ravn

INF-3981, Master's thesis in computer science, June 2017



Abstract

A Fully Homomorphic Encryption (FHE) scheme allows you to perform an arbitrary number of computation on encrypted data without decrypting it first. This is a very useful feature as it allows us to perform computations on data without knowing the data itself. Computations on sensitive data can be done in an untrusted environment without compromising privacy. Even though this may sound like the holy grail of cryptography which could solve the worlds IT problems when it comes to security and trust, it have some issues. Unfortunately Fully Homomorphic Encryption(FHE) libraries are very limited in Python. In this paper we have implemented a Python API based of a FHE C++ library called HElib and we will take a look at potential usages and limitations to the library.

Acknowledgements

Morbi sit amet diam condimentum, posuere tellus ut, fringilla massa. Nulla viverra dolor leo, vel cursus erat ornare non. Phasellus vehicula velit eget posuere rutrum. Integer at egestas enim, sed vulputate sapien. Nunc odio metus, mattis in erat at, rhoncus venenatis tortor. Vivamus porttitor molestie risus, vel gravida leo lacinia ac. Phasellus efficitur vehicula risus ut ornare. Pellentesque tincidunt libero ac massa finibus, sed hendrerit nisl pellentesque. Aenean rhoncus, nisi in suscipit aliquet, eros diam tempus nunc, non luctus turpis mauris ut lorem. Nulla sed scelerisque mauris.

Contents

Abstract	i
Acknowledgements	iii
Contents	vi
List of Figures	viii
List of Tables	ix
1 Introduction	1
2 Background	3
2.1 Machine learning	3
2.2 Deep learning	4
2.2.1 Convolutional neural network	4
2.3 Classification vs. detection	5
3 Methods	6
3.1 Dataset	6
4 Results	8
5 Discussion	9
5.1 asdf	9
6 Conclusion	11
6.1 Future Work	11
Bibliography	12
Appendices	14

List of Figures

2.1	Illustration of the convolutional layer . The red box is the input, and blue box is the result. The width and height of the input is preserved, but since the convolutional filter has 32 filters, the depth of the output volume is 32 pixels.	4
2.2	Vizualisation of four different image recognition tasks	5

List of Tables

A.1	asdf	16
-----	------	----

Introduction

When dealing with sensitive data we need to make sure that the data is secure. This means, if the data is leaked due to a security breach, error in the system or any other reasons, we need to make sure the data are unrecognizable for anyone who is not authorized. This is commonly solved by using an encryption algorithm to encrypt the data, making it unreadable to anyone who do not know the private key used during the encryption. As more and more businesses and organizations are moving away from having their own dedicated server to renting server space in large server parks, like a cloud, security are becoming a bigger concern than ever before. Traditionally businesses and organizations would have their own private server which would handle computation and storing of the data. The only people having direct access to the server would be authorized personnel hired by the business or the organization. Meaning the only real threat from a potential attacker would come from outside the business/organization, as any attacks from inside the business/organization would be met with prosecution and lawsuit.

As the trend of cloud services have exploded in the last years this have changed. We no longer have control over the physical location of the server, and we now have to trust the cloud service to uphold access control agreed upon. When storing data on cloud servers we are faced with not only the same security threats as on the private server, but also additional threats specific for cloud computing, like side channel attacks[12], virtualization vulnerabilities and abuse of cloud services. Cloud administrators or unauthorized users might access the data and obtain sensitive information for various motivations. Because of the increased threat when storing data in a cloud environment the data need to be encrypted at all times with our own encryption key, as even the cloud service cannot be trusted.

If we use the cloud service to only store data, then this is pretty simple, but what if we want to do some computation on the data stored? One way to do this is to encrypt the sensitive data and store this on the cloud. To perform a computation the data would be downloaded to a private server and decrypted. The computation is performed and the data is encrypted and uploaded back to the cloud. The problem with this approach, it's inefficient, especially when a lot of computation is done and it also requires a private server to perform the computation. Another approach is to use a Homomorphic Encryption(HE) scheme which enables us to perform computation on encrypted data without decrypting it first. This means, we can perform computation on ciphertexts generating an encrypted result which, when decrypted, matches the result of operations performed on the plaintext. By using Homomorphic Encryption we can perform the computation without ever exposing the data which means we can ensure that the data is secure at all times during the computation.

Why do we want Homomorphic Encryption? As shown in the previous example it allows us to do computation on our data without ever exposing the data during the computation. This essentially means that we can now export the computation work to a third party like a cloud service. We no longer have to pay for expensive servers to perform heavy computations, we can export the job an external server which is much cheaper in term of power consumption, server maintenance and bandwidth cost. This also allows us to pay for exactly the amount of storage and computational power needed, which can dynamically increased when needed. But is also have other benefits like hospital record can be examined without compromising patient privacy and financial data can be analyzed without opening it up to theft.

If Homomorphic Encryption is so great why are not everyone using it already? Even though there are many benefits which comes with using Homomorphic Encryption, there are some issues. First of all, performance. The first Fully Homomorphic Encryption scheme by Craig Gentry[1] was taking 100 trillion times as long to perform computations on encrypted data than plaintext computation. In this paper we will take a look at a FHE library called HElib which is an open source library in C++. With Boost Python we have created a Python API of this library and will use this to see how far the FHE schemes have come and look at practical usages and/or which challenges still remain for FHE schemes.

Background

This chapter describes relevant technical background and relevant work. We will first define the field of machine learning, and then describe the methods that have been used in this thesis. We will explain feature extraction for audio signals using spectrograms. We will also describe recent relevant work that uses the same machine learning methods and feature extractions methods as in this thesis.

2.1 Machine learning

Donec id massa ac leo consequat euismod et a sem. Pellentesque sem justo, vulputate vel neque a, ultrices dapibus ipsum. Vestibulum orci orci, semper ut odio et, luctus condimentum nunc. [1]. Mauris vel interdum nunc. Curabitur mi lectus, rhoncus venenatis ullamcorper quis, mattis volutpat dui. Proin at lectus ac metus ornare lacinia et pretium risus [2].

Accuracy

$$\text{Accuracy} = \frac{TP + TN}{P + N}$$

Aenean rhoncus, nisi in suscipit aliquet, eros diam tempus nunc, non luctus turpis mauris ut lorem. Nulla sed scelerisque mauris.

Recall

Aenean rhoncus, nisi in suscipit aliquet, eros diam tempus nunc, non luctus turpis mauris ut lorem. Nulla sed scelerisque mauris.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Aenean rhoncus, nisi in suscipit aliquet, eros diam tempus nunc, non luctus turpis mauris ut lorem. Nulla sed scelerisque mauris.

Precision

2.2 Deep learning

2.2.1 Convolutional neural network

Aenean rhoncus, nisi in suscipit aliquet, eros diam tempus nunc, non luctus turpis mauris ut lorem. Nulla sed scelerisque mauris. Figure 2.1 shows an illustration of the convolutional layer.

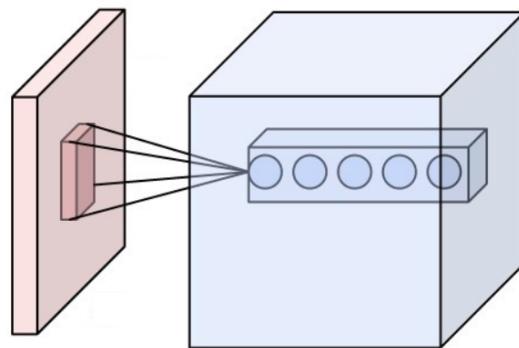


Figure 2.1: Illustration of the convolutional layer . The red box is the input, and blue box is the result. The width and height of the input is preserved, but since the convolutional filter has 32 filters, the depth of the output volume is 32 pixels.

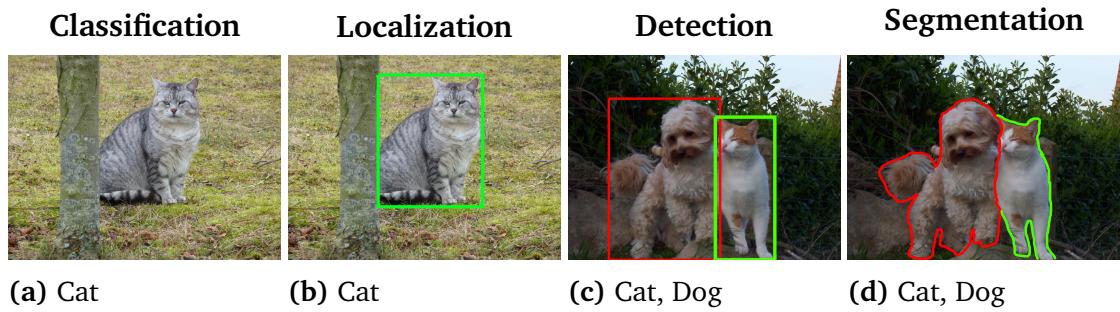


Figure 2.2: Vizualisation of four different image recognition tasks

2.3 Classification vs. detection

- **Classification:** Donec id massa ac leo consequat euismod et a sem. Pellentesque sem justo, vulputate vel neque a, ultrices dapibus ipsum. Vestibulum orci orci, semper ut odio et, luctus condimentum nunc.
- **Object localization:** Donec id massa ac leo consequat euismod et a sem. Pellentesque sem justo, vulputate vel neque a, ultrices dapibus ipsum. Vestibulum orci orci, semper ut odio et, luctus condimentum nunc. the size of the bounding box.
- **Object detection:** Donec id massa ac leo consequat euismod et a sem. Pellentesque sem justo, vulputate vel neque a, ultrices dapibus ipsum. Vestibulum orci orci, semper ut odio et, luctus condimentum nunc.

Methods

In this chapter, we describe the methods used in our approach. We will first describe the dataset, which includes data collection, class distribution and training/test partitioning. We then describe the general design of convolutional neural networks and how we train them. We will describe all experiments performed on the lung sound dataset. The experiments are wheeze detection, breathing phase detection and full file classification. For each experiment, we will describe data preprocessing, model architecture and training details for the best performing model. Finally, we will explain the design and implementation of an prototype that can utilize the capabilities of our lung sound analysis system. The lung sound analysis system is a system that uses the best performing model from each experiment.

3.1 Dataset

Results

Discussion

5.1 asdf

Conclusion

6.1 Future Work

Bibliography

- [1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, pp. 115–118, Feb 2017. Letter.
- [2] “Homesite quote conversion.” <https://www.kaggle.com/c/homesite-quote-conversion>. Accessed: 2017-30-05.

Appendices

tttt

Experiment	d	f	f	f
x	x	d	figure	f
x	x	d	f	f
x	x	d	f	f
x	x	d	f	f
x	x	d	f	f

Table A.1: asdf

