Deep Learning with PCA Wavelet Networks

CS39440 Major Project Report

Author: Samuel Spurling ([sas90@aber.ac.uk](mailto:sas90@aber.ac.uk))

Supervisor: Dr Bernie Tiddeman. My Supervisor (bpt@aber.ac.uk)

29th March 2022

Version 1.0 (Draft)

This report is submitted as partial fulfilment of a BSc degree in  
Computer Science and Artificial Intelligence (With Integrated Year in Industry) (GG47)

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Wales, UK

Declaration of originality

I confirm that:

* This submission is my own work, except where clearly indicated.
* I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
* I have read the regulations on Unacceptable Academic Practice from the University’s Academic Registry (AR) and the relevant sections of the current Student Handbook of the Department of Computer Science.
* In submitting this work, I understand and agree to abide by the University’s regulations governing these issues.

Name …………………………………………

Date ……………………………………………

Consent to share this work

By including my name below, I hereby agree to this project's report and technical work being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name …………………………………………

Date ……………………………………………

Acknowledgements

I am grateful to…

I’d like to thank…

Abstract

In this report we propose a novel application for PCA wavelet networks (PCAWN) [1]. The PCAWN decomposes an image using invertible 2D wavelet filters-banks and 1x1 filters learnt through principal component analysis to control the size of decompositions. The reconstruction is accomplished using the inverted filter bank and an approximately inverted PCA. Previously the approach has shown competitive results for linear inverse problems in computer vision such as compressive sensing, super-resolution, and in-painting. The task set is image segmentation. This is a task where there is an object in an image and the model must construct a mask over the object of interest. This problem is important in fields such as medical imaging. We use 2 variants of the PCAWN, a fully convolutional and a fully connected approach. In experiments, there will be a comparison of the PCAWN approach to the U-net model, a standard architecture for image segmentation, to compare the PCAWN models performance. The results of the experiments demonstrate that PCAWN has great potential when applied to the task of image segmentation. Possible future work looks at using PCAWN as regularisation in the U-net architecture and replacing PCA with a different kind of invertible dimensionality reduction such as Kernel PCA or UMAP.

Contents

[1 Background, Analysis & Process 6](#_Toc100868108)

[1.1 Background 6](#_Toc100868109)

[1.2 Analysis 6](#_Toc100868110)

[1.2.1 Fundamentals 6](#_Toc100868111)

[1.2.2 Architectures 6](#_Toc100868112)

[1.2.3 Research Questions 6](#_Toc100868113)

[1.3 Process 7](#_Toc100868114)

[2 Experiment Methods 8](#_Toc100868115)

[3 Software Design, Implementation and Testing 9](#_Toc100868116)

[3.1 Design 9](#_Toc100868117)

[3.1.1 Overall Architecture 9](#_Toc100868118)

[3.1.2 Some detailed design 9](#_Toc100868119)

[3.1.3 User Interface 9](#_Toc100868120)

[3.1.4 Other relevant sections 9](#_Toc100868121)

[3.2 Implementation 10](#_Toc100868122)

[3.3 Testing 10](#_Toc100868123)

[3.3.1 Overall Approach to Testing 10](#_Toc100868124)

[3.3.2 Automated Testing 10](#_Toc100868125)

[3.3.3 Integration Testing 11](#_Toc100868126)

[3.3.4 User Testing 11](#_Toc100868127)

[4 Results and Conclusions 12](#_Toc100868128)

[4.1 Issues with experiments 13](#_Toc100868129)

[4.1.1 Memory errors 13](#_Toc100868130)

[4.2 Convolutional Network approach 13](#_Toc100868131)

[5 Critical Evaluation 13](#_Toc100868132)

[6 References 15](#_Toc100868133)

[7 Appendices 16](#_Toc100868134)

[A. Third-Party Code and Libraries 17](#_Toc100868135)

[B. Code Samples 18](#_Toc100868136)

# Background, Analysis & Process

## Background

The project extends the PCAWN method [1] to the task of image segmentation, so naturally PCAWN was a key part of the background research. The motivation for this was to understand the novel approach to linear inverse problems and apply my findings to an image segmentation approach. Convolutional scattering wavelet networks [2] is another method which shares similarities to PCAWN, especially on methods for mitigating the issue of large channels after multiple applications of wavelet filter banks. To understand the segmentation task, there was research of U-net [3], DeepLabV3 [4] and a comparison of segmentation methods [5].

## Analysis

### Fundamentals

#### Principal Component Analysis

#### Wavelets

#### Segmentation

### Architectures

#### PCAWN

#### Scattering Wavelet Networks

#### U-Net Architecture

#### DeeplabV3 Architecture

The DeepLabV3 is state of the art for semantic segmentation. The approach proposes the use of atrous convolutions, used in a method called Atrous Spatial Pyramid Pooling (ASPP) [6]. The word “trous” in French means holes, which is the principle of atrous convolutions. This method uses convolutional filters with a dilation rate which determines the padding between values in the kernel used for the convolution. Atrous convolutions have the benefit of capturing dense features whilst maintaining the resolution of resultant feature maps. Deep convolutional networks that use max-pooling and striding do not maintain resolution of feature maps, meaning information is lost.

The formula for calculating an atrous convolution is:



Where y is the output feature map, x is the image, i is a location in x and y, r is the atrous rate and w is the kernel. By increasing the rate, we increase the number of zeros between consecutive values in the filter. For example, an atrous value of 1 is a conventional filter. An atrous rate of 2 pads one zero between every value in the filter.





Atrous Rate = 1

Atrous Rate = 2

The reason for including details on DeeplabV3 and atrous convolutions is that the ASPP method could be applied to future PCAWN work. Atrous convolutions have been used with undecimated wavelet networks before successfully [7] and would be an interesting edit to the PCAWN architecture.

### Research Questions

There were several initial questions posed:

* Can you reimplement the network?
* Can you apply the network to a novel task?
  + Image segmentation
  + Crowd counting
* Can another dimensionality reduction be used instead of PCA?
* What are the effects of changing network parameters?

Of these questions, I have completed 3, and would suggest the remainder for a future project.

#### Can you reimplement the network?

This task is necessary to complete for the success of the rest of the project. Simply, the task is to reimplement the network for a new dataset.

#### Can you apply the network to a novel task? (Image Segmentation)

This task was the main focus of the project. There are a number of technical aspects to answering this question which were expected to take a long time, which is why the remaining questions have been delegated to future research.

#### What are the effects of changing network parameters?

This task is a natural extension of the segmentation task. Some experimentation would be required to determine the best parameters for segmentation. There are several parameters to vary through experiments. For the model, the parameters to edit include layer count, percentage of channels kept, different architectures and different activation functions. For the dataset, the editable parameters are image resolution and training data size.

## Process

You need to describe briefly the life cycle model or research method that you used. You do not need to write about all of the different process models that you are aware of. Focus on the process model or research method that you have used. It is possible that you needed to adapt an existing method to suit your project; clearly identify what you used and how you adapted it for your needs.

For the research-oriented projects, there needs to be a suitable process for the construction of the software elements that support your work.

# Experiment Methods

This section should discuss the overall hypothesis being tested and justify the approach selected in the context of the research area. Describe the experiment design that has been selected and how measurements and comparisons of results are to be made.

You should concentrate on the more important aspects of the method. Present an overview before going into detail. As well as describing the methods adopted, discuss other approaches that were considered. You might also discuss areas that you had to revise after some investigation.

You should also identify any support tools that you used. You should discuss your choice of implementation tools or simulation tools. For any code that you have written, you can talk about languages and related tools. For any simulation and analysis tools, identify the tools and how they are used on the project.

For the parts of your project that need some engineering (hardware, software, firmware, or a mixture) to support the experiments, include details in your report about your design and implementation. You should discuss with your supervisor whether it is better to include a different top-level section to describe any engineering work. In this template, Chapter 3 is suggested as a place for that discussion.

# Software Design, Implementation and Testing

This could be one chapter or a few chapters. It should define and discuss the software that is developed to support the research that is being conducted. For example, if your research involves running experiments, what software are you creating to support that work? What functionality is required? What design will be used? What implementation issues are there and what testing is used?

Even though a research project is investigating specific research questions, it is still necessary for you to discuss the software that you develop. Research has a habit of generating bits of software that can exist for several years and need future modification. Therefore, you need to be able to discuss the technical issues as well as the research approach.

## Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

You should also identify any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

Some example sub-sections may be as follows, but the specific sections are for you to define.

### Overall Architecture

### Some detailed design

#### Even more detail

### User Interface

### Other relevant sections

## Implementation

This section should discuss issues you encountered as you tried to implement your experiments. What were the results of running the experiments? What conclusions can you draw from these results?

During the work, you might have found that elements of your experiments were unnecessary or overly complex; perhaps third-party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

If you had multiple experiments to run, it may be sensible to discuss each experiment in separate sections.

## Testing

Detailed descriptions of every test case are definitely not what is required in this section; the place for detailed lists of tests cases is in an appendix. In this section, it is more important to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Provide information in the body of your report and the appendix to explain the testing that has been performed. How does this testing address the requirements and design for the project?

How comprehensive is the testing within the constraints of the project? Are you testing the normal working behaviour? Are you testing the exceptional behaviour, e.g. error conditions? Are you testing security issues if they are relevant for your project?

Have you tested your system on “real users”? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

Whilst testing with “real users” can be useful, don't see it as a way to shortcut detailed testing of your own. Think about issues discussed in the lectures about until testing, integration testing, etc. User testing without sensible testing of your own is not a useful activity.

The following sections indicate some areas you might include. Other sections may be more appropriate to your project.

### Overall Approach to Testing

### Automated Testing

#### Unit Tests

#### User Interface Testing

#### Stress Testing

#### Other Types of Testing

### Integration Testing

### User Testing

# Results and Conclusions

This section should discuss issues you encountered as you tried to implement your experiments. What were the results of running the experiments? What conclusions can you draw from these results? What graphs or other information have you assessed regarding your experiments? Discuss those.

During the work, you might have found that elements of your experiments were unnecessary or overly complex; perhaps third-party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

If you had multiple experiments to run, it may be sensible to discuss each experiment in separate sections.

## Issues with experiments

### Memory errors

By far the most prevalent error was out of memory errors. The cause of this ranged from inefficient programming, insufficient RAM, and poor memory management. To solve the inefficient programming and memory management issues we removed unnecessary objects. The insufficient RAM issue was solved initially by using cloud-based platforms, and then by purchasing a new laptop, upgrading the RAM from 8GB to 16GB.

Memory errors meant initial experiments were throttled at a 128x128 image resolution. This in turn limited the “count” parameter of the PCAWN. This parameter controlled the number of layers in the head, and the inverse head. Every layer reduced the resolution of the composition by half. A 128x128 image would have a feature map with a 4x4 resolution after passing through the head component of a network with 5 layers. This caused an error with the convolutional filters used in reconstruction, as the resolution is too small.

The fully convolutional network had notably fewer memory problems compared to the fully connected version. The cause of this was the calculation of the A matrix, which uses computationally intense methods, such as calculating the pseudo-inverse. The A matrix for the fully connected method was larger than the fully convolutional matrix, meaning the memory usage was much higher for the fully connected variant.

## Convolutional Network approach

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **training data size** | **count** | **keep%** | **resolution** | **dice mean test** | **dice mean train** | **dice std test** | **dice std train** | **linear inverse train time** | **img train time** | **seg train time** |
| 4000 | 3 | 0.3 | 128 | 0.358 | 0.348 | 0.218 | 0.222 | 168.371 | 242.668 | 242.668 |
| 2000 | 3 | 0.3 | 128 | 0.360 | 0.353 | 0.218 | 0.220 | 83.499 | 124.600 | 124.600 |
| 6000 | 3 | 0.2 | 128 | 0.564 | 0.557 | 0.203 | 0.205 | 251.632 | 297.255 | 297.255 |
| 6000 | 3 | 0.1 | 128 | 0.565 | 0.558 | 0.204 | 0.206 | 226.381 | 355.888 | 355.888 |
| 6000 | 3 | 0.1 | 64 | 0.565 | 0.558 | 0.204 | 0.206 | 260.870 | 327.403 | 327.403 |
| 6000 | 3 | 0.3 | 256 | 0.564 | 0.557 | 0.203 | 0.205 | 256.319 | 327.596 | 327.596 |
| None | 3 | 0.3 | 128 | 0.591 | 0.584 | 0.196 | 0.194 | 328.039 | 433.187 | 433.187 |
| None | 4 | 0.3 | 128 | 0.602 | 0.595 | 0.182 | 0.183 | 441.355 | 783.941 | 783.941 |

This experiment tests the ability of the convolutional network approach to PCAWN. The results are sorted by the mean dice score. We can make a number of deductions from the results about what factors improve the performance of the model.

Training data size gives a significant improvement in the score (“None” means the full dataset was used, with 100 samples reserved for testing).

Keep percent does not have a statistically significant effect on the dice score. However, there is a considerable effect on the initial training time of the networks, a higher keep percent increases the training time.

Count also has an effect, however it is not statistically significant unless

# Critical Evaluation

Examiners expect to find a section addressing questions such as:

* Were the requirements correctly identified?
* Were the design decisions correct?
* Could a more suitable set of tools have been chosen?
* How well did the software meet the needs of those who were expecting to use it?
* How well were any other project aims achieved?
* If you were starting again, what would you do differently?

Other questions can be addressed as appropriate for a project.

The questions are an indication of issues you should consider. They are not intended as a specification of a list of sections.

The evaluation is regarded as an important part of the project report; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree.

There will be good things in the work and aspects of the work that could be improved. As you write this section, identify and discuss the parts of the work that went well and also consider ways in which the work could be improved.

In the latter stages of the module, we will discuss the evaluation. That will probably be around week 9, although that differs each year.

# References

|  |  |
| --- | --- |
| [1] | M. G. Bernard Tiddeman, "Principal Component Wavelet Networks for Solving Linear," *Symmetry,* vol. 13, no. 6, p. 1083, 2021. |
| [2] | S. M. Joan Bruna, "Invariant Scattering Convolution Networks," *IEEE transactions on pattern analysis and machine intelligence,* vol. 35, no. 8, pp. 1872-1886, 2013. |
| [3] | O. P. F. T. B. Ronneberger, " U-net: Convolutional networks for biomedical image segmentation.," in *International Conference on Medical image computing and computer-assisted intervention*, Cham, 2015. |
| [4] | Y. H. Ş. a. G. U. S. C. Yurtkulu, "Semantic Segmentation with Extended DeepLabv3 Architecture," in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, 2019. |
| [5] | S. e. a. Ghosh, ""Understanding deep learning techniques for image segmentation."," *ACM Computing Surveys (CSUR) ,* vol. 52, no. 4 , pp. 1-35, 2019. |

# Appendices

The appendices are for additional content that is useful to support the discussion in the report. It is material that is not necessarily needed in the body of the report, but its inclusion in the appendices makes it easy to access.

If you have used any 3rd party code, i.e. code that you have not written yourself such as libraries, then you must include Appendix A. In that appendix, you will provide details of the 3rd party code that you have used.

For most other items, it would be better to include them in your technical submission instead of including them as an appendix. For example:

* If you have developed a Design Specification document as part of a plan-driven approach for the project, then it would be appropriate to include that document in the technical work. In this report, you would highlight the most interesting aspects of the design, referring your reader to the full specification for further detail.
* If you have taken an agile approach to developing the project, then you may be less likely to have developed a full requirements specification at the start of the project. Perhaps you used stories to keep track of the functionality and the ‘future conversations.’ If it isn’t relevant to include all those stories in the body of your report, you could detail those stores in a document in the technical work.
* If you have used manual testing, then include a document in the technical work that records the tests that have been done. In this report, you would talk about the use of those tests.

Documents included in the technical work or in the appendices are supporting evidence of the work done. Where you include documents, this report should refer to the documents. You should not be relying on detailed study of those documents in order to understand what is written in this report.

Speak to your supervisor or the module coordinator if you have questions about this.

* 1. Third-Party Code and Libraries

If you have made use of any third-party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. If third-party code or libraries are used, your work will build on that to produce notable new work. The key requirement is that we understand what your original work is and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

The following is an example of what you might say.

**Apache POI library** – The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the client’s existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [5]. The library is released using the Apache License [6]. This library was used without modification.

Include as many declarations as appropriate for your work. The specific wording is less important than the fact that you are declaring the relevant work.

* 1. Code Samples

This is an example appendix. Include as many appendices as you need. The appendices do not count towards the overall word count for the report.

For some projects, it might be relevant to include some code extracts in an appendix. You are not expected to put all of your code here - the correct place for all of your code is in the technical submission that is made in addition to the Project Report. However, if there are some notable aspects of the code that you discuss, including that in an appendix might be useful to make it easier for your readers to access.

As a general guide, if you are discussing short extracts of code then you are advised to include such code in the body of the report. If there is a longer extract that is relevant, then you might include it as shown in the following section.

Only include code in the appendix if that code is discussed and referred to in the body of the report.

Random Number Generator

The Bayes Durham Shuffle ensures that the pseudo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs.

// Some example code here…