UNIVERSITAT DE BARCELONA

End-of-Degree Project

Simultaneous Degrees of Mathematics and Computer Science

Faculty of Mathematics and Computer Science
Universitat de Barcelona

WaveFood: Wavelets CNN Food Segmentation

Author: David Fernández Gómez

Supervisors: Dr. Albert Clop Ponte
Ahmad AlMughrabi
Prof. Petia Ivanova Radeva

Barcelona, January 15, 2026

Dedication

I would like to thank Dr. Albert Clop Ponte, Ahmad AlMughrabi, and Prof. Petia Ivanova Radeva for their always fast replies, for allowing me the opportunity of combining my favourite Mathematics and Computer Science branches and always helping me target my best.

I would like to thank my family, as they have been an unconditional support on my darker days, giving me peace when I wouldn't find it and love when I most needed it. This double degree has been only possible thanks to them and for that I will be forever thankful.

I ought to thank my partner Mar, as she has been my most precious support during almost all the duration of the double degree, sharing classes, studying hours, cries and laughs, both growing always together.

Finally, I would like to acknowledge my own perseverance and commitment to excellence. Always aiming for the best has been a demanding journey, but the effort has truly paid off. I am deeply proud of this thesis, as it represents the culmination of months of my hardest work.

Highlights

- **WaveFood Framework:** We propose a novel Deep Learning framework that integrates differentiable spectral layers, specifically Haar and Zernike transforms, into Convolutional Neural Networks to enhance feature extraction for complex food textures.
- **Memory-Efficient Multi-Level Architecture:** We introduce a Multi-Level Spectral Pyramid strategy that drastically reduces VRAM consumption by approximately **23**%, while maintaining high accuracy, validating spectral features as a resource-efficient alternative to raw spatial data.
- **Rotation-Invariant Feature Superiority:** We empirically demonstrate that the rotation-invariant Zernike bases are far superior for modeling organic food shapes compared to the rigid, axis-aligned Haar Wavelets, which proved unsuitable for the domain.
- **Surpassing a strong baseline:** The proposed Zernike Moments architecture outperforms the robust CCNet baseline in semantic segmentation accuracy on the challenging FoodSeg103 dataset by **0.87**%, **1.74**% and **0.14**% in mIoU, mAcc and aAcc correspondingly.
- **Training Efficiency:** By leveraging efficient spectral projections, the best-performing model accelerates the training process by approximately **22**% compared to the baseline while improving segmentation performance.

Abstract

(English) Semantic food segmentation faces unique challenges due to high intra-class variance and amorphous, organic shapes. While Convolutional Neural Networks (CNNs) are a very powerful tool, they typically require massive computational resources to handle such texture complexities. Our work proposes WaveFood, a novel Deep Learning framework integrating Spectral Image Representation, based on the hypothesis that CNNs benefit from deeper spectral representations than the canonical spatial representation used by humans. Benchmarked against a robust CCNet baseline on the FoodSeg103 dataset, our experiments show that the rigid, axis-aligned Haar Wavelets are unsuitable for food. In contrast, Zernike approaches yield accuracy surpassing the baseline. Notably, the orthonormal Zernike Moments proved superior to the custom MRA wavelet approach, surpassing the baseline's accuracy by **0.87**%, **1.74**% and **0.14**% in mIoU, mAcc and aAcc correspondingly, while reducing training time by approximately **22**%. Furthermore, our Multi-Level architecture significantly lowers VRAM usage by approximately **23**%, maintaining baseline accuracy, validating spectral features as a superior efficiency-performance alternative to raw image data. The source code is available at: [2].

(Catalan) La segmentació semàntica d'aliments presenta reptes únics a causa de l'alta variància intra-classe i les formes orgàniques i amorfes. Tot i que les Xarxes Neuronals Convolucionals (CNNs) són una eina molt potent, sovint requereixen recursos computacionals massius per gestionar aquestes complexitats de textura. El nostre treball proposa WaveFood, un nou *framework* de Deep Learning que integra la Representació d'Imatge Espectral, basat en la hipòtesi que les CNNs es beneficien de representacions espectrals més profundes que la representació espacial canònica utilitzada pels humans. Avaluat en el dataset FoodSeg103 contra un *baseline* robust CCNet, els nostres experiments demostren que la naturalesa rígida de les Wavelets de Haar i la seva alineació amb els eixos són inadequades per al menjar. En canvi, els enfocaments de Zernike proporcionen una precisió que supera la del *baseline* per **0.87**%, **1.74**% and **0.14**% en mIoU, mAcc and aAcc corresponentment. Destaca especialment que els Moments ortonormals de Zernike han resultat superiors a l'enfocament personalitzat de wavelets MRA, superant la precisió del *baseline* i reduint el temps d'entrenament aproximadament un **22**%. A més, la nostra arquitectura Multi-Nivell redueix significativament l'ús de VRAM un **23**% aproximadament, mantenint la precisió del *baseline*, validant les característiques espectrals com una alternativa superior en eficiència i rendiment a les dades d'imatge originals El codi relacionat es pot trobar a: [2].

2020 Mathematics Subject Classification. 68T07, 68U10, 42C40, 65T60, 65D05, 65F20

[2]https://github.com/GCVCG/WaveFood

Contents

Introduction

Computer vision has revolutionized how machines interpret the visual world, achieving remarkable milestones in fields like autonomous driving or medical diagnostics. However, applying these technologies to *Food Computing*[3] [31] reveals a singular challenge: food is chaotic. Unlike cars or buildings, a food dish lacks a rigid structure; it is amorphous, presents complex high-frequency textures, and, crucially, lacks a fixed orientation.

Currently, the gold standard for solving this problem is Semantic Segmentation via Convolutional Neural Networks (CNNs). Formally, *semantic segmentation* is defined as the dense prediction task where every pixel in the input image is assigned a specific class label (e.g., 'pizza' or 'background'), effectively partitioning the image into semantically meaningful regions. Large Neural Network models learn to perform this pixel-wise classification by optimizing millions of parameters. However, this approach comes at a cost: it demands substantial computational resources and relies heavily on data augmentation to learn that a rotated pizza remains a pizza.

Proposed Goal

Our work, we call *WaveFood*, addresses the following hypothesis: deep learning models do not need to learn edge detection or texture extraction from scratch using only spatial convolutions. Instead, we propose using *Spectral Image Representation*, which means decomposing images into frequencies, as a highly efficient feature extractor. To this end, we explore in detail three different mathematical strategies: standard Haar Wavelets [19], Zernike Moments [43] and Zernike Wavelets [15].

We aim to design, implement, and validate architectures that integrate these mathematical transforms as differentiable layers within an existing robust Deep Learning pipeline based on the *CCNet* [24] architecture. We do not aim merely to add theoretical complexity, but to demonstrate practical utility using the *FoodSeg103* dataset, one of the most demanding benchmarks in the food image analysis field.

Main Contributions

This study challenges the intuition that "more complex is better". In an era where AI efficiency is becoming as critical as accuracy, we propose *WaveFood* that looks for new mathematical foundations to redefine new deep learning models, offering architectures that not only improve speed, but also drastically reduce memory consumption while maintaining their accuracy. We demonstrate how a well-chosen mathematical

[3] We use Italic text to highlight key words and concepts.

projection can surpass the performance of heavy baselines while accelerating training by approximately *22%*.

Concretely, this thesis presents the following key contributions:

- **Rigorous Mathematical Formulation:** A comprehensive construction of the 2D Haar Multiresolution Analysis and the Zernike MRA on the unit disk, establishing the theoretical foundations required for their discretization and implementation as neural layers.
- **New WaveFood Architecture:** A novel framework integrating differentiable Haar and Zernike spectral layers into CNNs.
- **Geometric Hypothesis Validation:** Empirical demonstration that the rotation-invariant properties of Zernike bases are superior for modeling organic food shapes compared to the axis-aligned Haar Wavelets.
- **Superior Performance with respect to the State of the art:** Surpassing the CCNet baseline in accuracy by **0.87**%, **1.74**% and **0.14**% in mIoU, mAcc and aAcc correspondingly, while reducing training time by $\sim$ **22**%.
- **Resource Efficiency:** A Multi-Level strategy that maintains baseline accuracy while drastically lowering VRAM consumption by approximately **23**%.

Agile Project Research and Development Methodology

To ensure efficient progress and adaptability given the experimental nature of our work, the development followed an Agile and Lean-inspired workflow. This approach prioritized:

- **Iterative Development:** Rapid prototyping of spectral layers (Haar, Zernike) to validate hypotheses before full-scale training.
- **Continuous Feedback Loops:** Regular 1:1 technical meetings were conducted to align theoretical formulations with implementation details.
- **Active Communication:** Usage of instant communication channels (e.g., Slack) for immediate troubleshooting of hardware constraints (e.g., CUDA compatibility) and code refactoring.

This structured yet flexible approach allowed for the timely identification of bottlenecks and the corresponding fast response.

Structure

The remainder of this thesis is organized into two primary parts: Part I establishes the mathematical foundations, while Part II details the practical implementation of the proposed differentiable layers and analyses their experimental performance.

Specifically, in Part I, we introduce standard wavelets and review the fundamental Multiresolution Analysis (MRA) on the real line $\mathbb{R}$ and its extension to the plane $\mathbb{R}^2$. Subsequently, we examine the adaptation of these concepts to the unit disk $B(0,1)$, focusing particularly on Haar Wavelets, Zernike Moments, and Zernike Wavelets, alongside their respective discrete transforms.

In Part II, we commence with a review of preliminaries in food segmentation and prior spectral approaches, describing the dataset and baseline in detail. We then discuss the implementation methodology, presenting the WaveFood framework and the three proposed architectures. Finally, we report our experimental results, discuss the limitations of the study, and provide concluding remarks and directions for future research.

Part I

Theoretical Background

Chapter 1

Wavelet theory

This chapter establishes the fundamental mathematical theory of wavelets. We begin by defining the construction of the Haar Wavelet, the simplest orthonormal wavelet basis, and prove its completeness in $L^2(\mathbb{R})$, laying the groundwork for more complex spectral analyses.

1.1 Introduction to Wavelets

This section defines the concept of wavelets and introduces the Haar wavelet.

Definition 1.1 (*Integer translates and dilates*)**.** Let $\psi \in L^2(\mathbb{R})$. The *integer translation* of ψ by shift $k \in \mathbb{Z}$ is defined as the function $x \mapsto \psi(x-k)$. The *dyadic dilation* of ψ by scale (or generation) $j \in \mathbb{Z}$ is defined as the function $x \mapsto 2^{j/2}\psi(2^j x)$. From now on, we will refer to the dyadic dilation as only dilation. $\lozenge$

Definition 1.2 (*Wavelet*)**.** A function $\psi \in L^2(\mathbb{R})$ is called a *wavelet* if the family of all its integer translates and dilates

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k) \quad \text{for } j,k \in \mathbb{Z} \tag{1.1}$$

forms an orthonormal basis of $L^2(\mathbb{R})$. If so, the basis formed by this family is called a *wavelet basis.* $\lozenge$

Given the completeness of an orthonormal basis, any $L^2(\mathbb{R})$ function may be reconstructed as follows.

Proposition 1.3 (*Reconstruction formula*)**.** *Given a wavelet $\psi \in L^2(\mathbb{R})$, the following reconstruction formula holds in the L^2 sense:*

$$f(x) = \sum_{j,k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k}(x) \quad \textit{for all } f \in L^2(\mathbb{R}). \tag{1.2}$$

$\lozenge$

Definition 1.4. *The orthogonal wavelet transform* is the map $W : L^2(\mathbb{R}) \to \ell^2(\mathbb{Z}^2)$ that assigns to each $L^2(\mathbb{R})$ function the sequence of its wavelet coefficients:

$$Wf(j,k) := \langle f, \psi_{j,k}\rangle = \int_{\mathbb{R}} f(x)\overline{\psi_{j,k}(x)}dx.$$

$\lozenge$

Lemma 1.5. *One can immediately see the following:*

$$f(x) = \sum_{j,k\in\mathbb{Z}} Wf(j,k)\psi_{j,k}(x) \quad \textit{for all } f \in L^2(\mathbb{R}).$$

◊

Finding wavelets is a very challenging task, as their required properties are highly restrictive. In 1910, Alfréd Haar introduced the first wavelet, the Haar Wavelet [19].

Definition 1.6 (*The Haar Wavelet*)**.** The *Haar wavelet* h is given by

$$h(x) := -\chi_{[0,1/2)}(x) + \chi_{[1/2,1)}(x).^1 \tag{1.3}$$

The family $\{h_{j,k}(x) := 2^{j/2}h(2^j x - k)\}_{j,k\in\mathbb{Z}}$ forms an orthonormal basis for $L^2(\mathbb{R})$ called the Haar basis, as will be proven in the next section. ◊

One can immediately see that $h \in L^2(\mathbb{R})$, as $\int_{\mathbb{R}} h(x)\overline{h(x)}dx = \int_0^1 dx = 1 < +\infty$. Fig. 1.1 shows the Haar wavelet plot, where we can immediately spot its discontinuities.

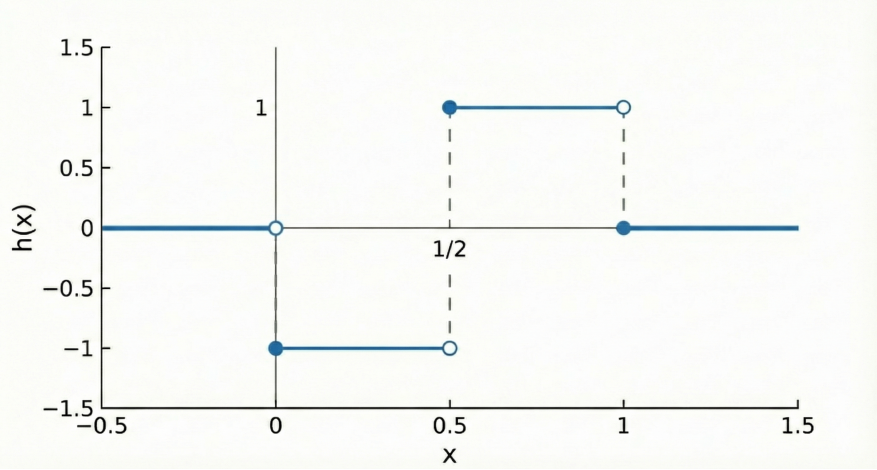


Figure 1.1: Plot of the Haar wavelet $h(x) = -\chi_{[0,1/2)}(x) + \chi_{[1/2,1)}$.

1.2 The Construction of the Haar Wavelet

This section details the step-by-step construction of the Haar Wavelet. We introduce the concept of dyadic intervals, define the Haar family of functions, and establish the necessary operators to rigorously prove its completeness as an orthonormal basis for $L^2(\mathbb{R})$.

1.2.1 The Dyadic Intervals

We establish the preliminary definitions of dyadic intervals, children intervals, and integral averages. These geometric and functional concepts serve as the fundamental building blocks for defining the Haar functions and proving their orthogonality properties in the subsequent subsections.

First and foremost, here there are some basic interval definitions and notations, frequently used in this section.

[1] χ_I denotes the characteristic function of the interval I, which takes the value 1 for $x \in I$ and 0 otherwise.

Definition 1.7. Given an interval $I = [a, b) \subseteq \mathbb{R}$, we define the *left child* of I as $I_l := [a, (a + b)/2)$. The *right child* of I is $I_r := [(a + b)/2, b)$. The *length* of I is $|I|$. Given a locally integrable function f, the *integral average of f over I* is

$$m_I f := \frac{1}{|I|} \int_I f(x)dx.$$

$\Diamond$

Definition 1.8 (*Dyadic Intervals*)**.** An interval $I \subset \mathbb{R}$ is called a *dyadic interval* if there exist $j, k \in \mathbb{Z}$ so that

$$I = I_{j,k} = [k2^{-j}, (k + 1)2^{-j}).$$

We define $\mathcal{D} := \{I_{j,k}\}_{j,k\in\mathbb{Z}}$ as the set of all dyadic intervals in $\mathbb{R}$ and $\mathcal{D}_j := \{I_{j,k}\}_{k\in\mathbb{Z}}$ as the set of dyadic intervals of length 2^{-j}, called the j^{th} *generation*. One can immediately see that each $\mathcal{D}_j$ gives a partition of $\mathbb{R}$. $\Diamond$

Lemma 1.9. *The left child and the right child of a dyadic interval are also dyadic intervals.*

Proof. Given a dyadic interval $I = I_{j,k} \in \mathcal{D}$, one can easily show that $I_l = I_{j+1,2k}$ and that $I_r = I_{j+1,2k+1}$ $\square$

A very important property of this type of intervals is shown in the following proposition.

Proposition 1.10 (*Dyadic Intervals Are Nested or Disjoint*)**.** *If $I, J \in \mathcal{D}$, then either $I \cap J = \emptyset$ or $I \subseteq J$ or $J \subsetneq I$. Moreover, if $J \subsetneq I$, then J is a subset of the left or the right child of I.*

Proof. Given $I, J \in \mathcal{D}$, then there exist $j_1, k_1, j_2, k_2 \in \mathbb{Z}$ so that $I = I_{j_1,k_1}$ and $J = I_{j_2,k_2}$. We can separate 3 different cases:

1. **Case $j_1 = j_2, k_1 = k_2$.** Then trivially $I = J$, so $I \subseteq J$.
2. **Case $j_1 = j_2, k_1 \neq k_2$.** Then the intervals are distinct intervals of the same length.

$$I \cap J = [k_1 2^{-j_1}, (k_1 + 1)2^{-j_1}) \cap [k_2 2^{-j_1}, (k_2 + 1)2^{-j_1}) = \emptyset.$$

3. **Case $j_2 > j_1$.** The interval J is strictly shorter than I. We can rewrite the endpoints of I as follows:

$$I = \left[\frac{k_1}{2^{j_1}}, \frac{k_1 + 1}{2^{j_1}}\right) = \left[\frac{k_1 2^{j_2-j_1}}{2^{j_2}}, \frac{(k_1 + 1)2^{j_2-j_1}}{2^{j_2}}\right).$$

Let $m = 2^{j_2-j_1}$. Then I represents the union of m consecutive dyadic intervals of the j_2^{th} generation:

$$I = \bigcup_{n=0}^{m-1} \left[\frac{k_1 m + n}{2^{j_2}}, \frac{k_1 m + n + 1}{2^{j_2}}\right).$$

Since J is a j_2^{th} generation interval, it may be one of the ones that compose I, thus having $J \subsetneq I$, or it may fall at either side of I, thus having $I \cap J = \emptyset$.

The case $\boldsymbol{j_1 > j_2}$ is symmetric to case 3 (swapping the roles of I and J), leading to $I \subseteq J$ or $I \cap J = \emptyset$. Thus, any two dyadic intervals are either disjoint or one is contained in the other.

Moreover, if $J \subsetneq I$, as shown in the previous lemma, I_l and I_r are also dyadic intervals and they span I. So, J has to be a subset of one of them. □

1.2.2 The Haar Family

We can now define a family of functions that closely relate to the dyadic intervals.

Definition 1.11. Given $I \in \mathcal{D}$, the *Haar function associated to the interval* I is the step function h_I defined by

$$h_I(x) := \frac{1}{\sqrt{|I|}}(\chi_{I_r}(x) - \chi_{I_l}(x)). \tag{1.4}$$

◊

One can immediately notice that the Haar wavelet h from (1.3) is the Haar function $h_{[0,1)}$, associated to the interval $[0, 1)$. Moreover, the Haar basis $\{h_{j,k}\}_{j,k\in\mathbb{Z}}$ is exactly the family of all possible Haar functions $\{h_{I_{j,k}}\}_{j,k\in\mathbb{Z}}$, as proven in the next proposition.

Proposition 1.12. *Given the dyadic interval $I = I_{j,k}$ and its associated Haar function h_I:*

1. *$h_I = h_{j,k}$, where $h_{j,k}$ is the corresponding Haar basis element.*
2. *h_I has 0 integral average over I, i.e., $m_I h_I = 0$.*
3. *h_I has a unit L^2 norm.*

Proof.

1.
$$\begin{aligned} h_{j,k}(x) &= 2^{j/2}h(2^j x - k) = 2^{j/2}\left(-\chi_{[0,1/2)}(2^j x - k) + \chi_{[1/2,1)}(2^j x - k)\right) \\ &= 2^{j/2}\left(-\chi_{[2^{-(j+1)}2k,\, 2^{-(j+1)}(2k+1))}(x) + \chi_{[2^{-(j+1)}(2k+1),\, 2^{-(j+1)}(2k+2))}(x)\right) \\ &= \frac{1}{\sqrt{|I_{j,k}|}}\left(-\chi_{I_{j+1,2k}}(x) + \chi_{I_{j+1,2k+1}}(x)\right) = h_{I_{j,k}} = h_I. \end{aligned}$$

2.
$$\begin{aligned} m_I h_I &= \frac{1}{|I|}\int_I h_I dx = \frac{1}{|I|^{3/2}}\int_I (-\chi_{I_l}(x) + \chi_{I_r}(x))dx \\ &= \frac{1}{|I|^{3/2}}\left(-\int_{I_l} dx + \int_{I_r} dx\right) = \frac{-|I|/2 + |I|/2}{|I|^{3/2}} = 0. \end{aligned}$$

3. As $h_I(x)$ only takes the values $\frac{1}{\sqrt{|I|}}$ and $\frac{-1}{\sqrt{|I|}}$,
$$\langle h_I, h_I\rangle = \int_I h_I^2(x)dx = \frac{1}{|I|}\int_I dx = 1.$$

□

To prove that indeed the Haar family introduced in the Def. 1.6 forms a basis for $L^2(\mathbb{R})$, it suffices to prove that it form an orthonormal basis for $L^2(\mathbb{R})$.

Lemma 1.13. *The Haar family $\{h_I\}_{I\in\mathcal{D}}$ is an orthonormal family.*

Proof. Given $I, J \in \mathcal{D}$ with $I \neq J$, as shown in Prop. 1.10, they are disjoint or one is strictly contained in the other. If I and J are disjoint, then obviously $\langle h_I, h_J\rangle = 0$, as their supports are disjoint. If I is strictly contained in J, then I is contained in J_l or J_r and thus $h_J = \frac{-1}{\sqrt{|J|}}$ on I if it's contained in J_l and $\frac{1}{\sqrt{|J|}}$ otherwise.

$$\langle h_I, h_J\rangle = \int_I h_I(x)h_J(x)dx = \frac{\pm 1}{\sqrt{|J|}}\int_I h_I(x)dx = 0,$$

where at the last step, we've applied the 2nd item of Prop. 1.12. The $J \subsetneq I$ is proven exchanging I and J in the previous proof. This proves that the family $\{h_I\}_{I\in\mathcal{D}}$ is orthogonal. By the 3rd item of Prop. 1.12 we also have the normality. □

If we add to the orthonormality of the system its *completeness*, we will have proven that the Haar family is a basis for $L^2(\mathbb{R})$, as we wanted. The following Theorem gives us this result.

Theorem 1.14. *The Haar family $\{h_I\}_{I\in\mathcal{D}}$ form an orthonormal basis for the $L^2(\mathbb{R})$ space.* ◊

If we were dealing with a finite-dimensional space, given an orthonormal set, we would have to check that the space's dimension and the set's cardinality are the same. Sadly, $L^2(\mathbb{R})$ is an *infinite-dimensional* space and so, that does not work, and we need to prove that the set is *complete*. Thus, we need to prove that for each $f \in L^2(\mathbb{R})$, the following identity holds in the L^2 sense:

$$f = \sum_{I\in\mathcal{D}} \langle f, h_I\rangle h_I. \tag{1.5}$$

If we recall the Prop. 1.3, it makes sense that this is a mandatory condition, as if the Haar family generates a wavelet basis for $L^2(\mathbb{R})$ it must hold the *reconstruction formula* (1.2).

1.2.3 The Expectation and Difference Operators

To be able to prove the *completeness* of the Haar family, we have to introduce two new *operators*: the *expectation* operators P_j and the *difference* operators Q_j.

Definition 1.15 (*Expectation operators*)**.** The *expectation operators* $P_j : L^2(\mathbb{R}) \to L^2(\mathbb{R})$, $j \in \mathbb{Z}$, takes the average value of the given $f \in L^2(\mathbb{R})$ function over dyadic intervals of the j^{th} generation as follows:

$$P_j f(x) := \frac{1}{|I_j|}\int_{I_j} f(y)dy,$$

where $I_j = I_{j,k}(x)$ is the unique dyadic interval of the j^{th} generation that contains x.◊

The new $P_j f \in L^2(\mathbb{R})$ is a *step function* which is constant in each $I \in \mathcal{D}_j$ (of the j^{th} generation and of length 2^{-j}). Given one of those intervals $I \in \mathcal{D}_j$, the value of $P_j f$ over that interval takes the integral average of f over said interval, so for $x \in I, P_j f(x) = m_I f(x)$. Thus, given that all intervals $I \in \mathcal{D}_j$ are disjoint, one can easily show that $P_j f(x) = \sum_{I \in \mathcal{D}_j} m_I f(x)$.

Definition 1.16 (*Difference operators*)**.** The *difference operators* $Q_j : L^2(\mathbb{R}) \to L^2(\mathbb{R})$, $j \in \mathbb{Z}$, takes the difference of detail in the expectation operators of consecutive resolutions as follows:

$$Q_j f(x) := P_{j+1} f(x) - P_j f(x).$$

$\Diamond$

The new $Q_j f \in L^2(\mathbb{R})$ is also *step function* which is constant in each $I \in \mathcal{D}_{j+1}$.

Lemma 1.17. *Given an interval $I \in \mathcal{D}, m_I f = (m_{I_l} f + m_{I_r} f)/2$.*

Proof.

$$m_I f = \frac{1}{|I|} \int_I f = \frac{1}{2} \left(\frac{1}{|I_l|} \int_{I_l} f + \frac{1}{|I_r|} \int_{I_r} f \right) = (m_{I_l} f + m_{I_r} f)/2.$$

$\square$

Proposition 1.18. *For $f \in L^2(\mathbb{R}), Q_j f(x) = \sum_{I \in \mathcal{D}_j} \langle f, h_I \rangle h_I(x)$.*

Proof. Given $x \in \mathbb{R}$, as $\mathcal{D}_j$ forms a disjoint partition of $\mathbb{R}$ there's a unique interval I such that $x \in I \in \mathcal{D}_j$. Thus, remembering the disjoint support of the Haar functions, the right-hand side gets reduced to only the summand of the given interval I. Noting that $|I| = 2|I_r| = 2|I_l|$, the right-hand side equals

$$\begin{aligned} \langle f, h_I \rangle h_I(x) &= \frac{1}{\sqrt{|I|}} \left(\int_{I_r} f - \int_{I_l} f \right) h_I(x) = \frac{\sqrt{|I|}}{2} \left(\frac{1}{|I_r|} \int_{I_r} f - \frac{1}{|I_l|} \int_{I_l} f \right) h_I(x) \\ &= \frac{\sqrt{|I|}}{2} (m_{I_r} f - m_{I_l} f) h_I(x). \end{aligned}$$

Since for $x \in I_r, h_I(x) = 1/\sqrt{|I|}$ and for $x \in I_l, h_I(x) = -1/\sqrt{|I|}$, we can simplify the expression as follows: if $x \in I$, then

$$\langle f, h_I \rangle h_I(x) = \begin{cases} (m_{I_r} f - m_{I_l} f)/2, & \text{if } x \in I_r; \\ -(m_{I_r} f - m_{I_l} f)/2, & \text{if } x \in I_l. \end{cases}$$

Following the Def. 1.16 on the left-hand side of the equation, if $x \in I \in \mathcal{D}_j$, then $P_j f(x) = m_I f(x)$. If $x \in I_r$, then $P_{j+1} f(x) = m_{I_r} f$ and if $x \in I_l$, then $P_{j+1} f(x) = m_{I_l} f$. Hence

$$Q_j f(x) = P_{j+1} f(x) - P_j f(x) = \begin{cases} m_{I_r} f - m_I f, & \text{if } x \in I_r; \\ m_{I_l} f - m_I f, & \text{if } x \in I_l. \end{cases}$$

The averaging property 1.17 implies that

$$m_{I_r} f - m_I f = (m_{I_r} f - m_{I_l} f)/2 = -(m_{I_l} f - m_I f).$$

We conclude that $Q_j f(x) = \sum_{I \in \mathcal{D}_j} \langle f, h_I \rangle h_I(x)$. In fact, this value is constant for $x \in I$. $\square$

1.2.4 The Completeness of the Haar Family

Now, we can use the previous operators to prove the *completeness* of the Haar family. To achieve this, we must show that equation (1.5) holds in the L^2-norm.

Following Prop. 1.18, and given that $\mathcal{D} = \bigcup_{j\in\mathbb{Z}} \mathcal{D}_j$, it suffices to prove that the following equality holds in the L^2 sense:

$$f = \lim_{M,N\to\infty} \sum_{j=-M}^{N-1} Q_j f. \tag{1.6}$$

Note that we can express the partial sums of the difference operators as a telescoping series:

$$\sum_{j=-M}^{N-1} Q_j f = \sum_{j=-M}^{N-1} (P_{j+1}f - P_j f) = P_N f - P_{-M} f \quad \forall f \in L^2(\mathbb{R}).$$

Thus, subtracting f and taking the L^2-norm on both sides, and applying the triangle inequality, we obtain:

$$\begin{aligned}\left\| \sum_{j=-M}^{N-1} Q_j f - f \right\|_2 &= \|(P_N f - P_{-M} f) - f\|_2 = \|(P_N f - f) - P_{-M} f\|_2 \\ &\leq \|P_N f - f\|_2 + \|P_{-M} f\|_2 .\end{aligned}$$

Therefore, if the right-hand side tends to zero as $N, M \to \infty$, then the equality (1.6) holds in the L^2 sense, as we wanted. This is precisely what the next Theorem establishes.

Theorem 1.19. *For any $f \in L^2(\mathbb{R})$, the following limits hold:*

$$\lim_{M\to\infty} \|P_{-M} f\|_2 = 0 \quad \textit{and} \tag{1.7}$$

$$\lim_{N\to\infty} \|P_N f - f\|_2 = 0. \tag{1.8}$$

$\Diamond$

To prove this theorem, first we need to present two lemmas.

Lemma 1.20. *The expectation operators P_j are* uniformly bounded *in $L^2(\mathbb{R})$. In fact, for $f \in L^2(\mathbb{R})$ and for $j \in \mathbb{Z}$,*

$$\|P_j f\|_2 \leq \|f\|_2 .$$

Proof. Given that $\mathcal{D}_j$ is a partition of $\mathbb{R}$ and given that by the Cauchy-Schwarz Inequality [2] we have

$$|P_j f|^2 = \left| \frac{1}{|I|} \int_I f \right|^2 \leq \frac{1}{|I|^2} \left(\int_I 1^2 \right) \left(\int_I |f|^2 \right) = \frac{1}{|I|} \int_I |f|^2,$$

one can easily see that

$$\|P_j f\|_2 = \int_{\mathbb{R}} |P_j f|^2 = \sum_{I\in\mathcal{D}_j} \int_I |P_j f|^2 \leq \sum_{I\in\mathcal{D}_j} \int_I |f|^2 = \int_{\mathbb{R}} |f|^2 = \|f\|_2 .$$

□

[2] It states that in $L^2(\mathbb{R}^n)$ space, the following inequality holds: $\left|\int_{\mathbb{R}^n} f\overline{g}\right|^2 \leq \int_{\mathbb{R}^n} |f|^2 \int_{\mathbb{R}^n} |g|^2$. The general statement can be found in Prop. A.3.

Lemma 1.21. *If g is a* continuous *function and has* compact support *(i.e., $g \in C_c(\mathbb{R})$), then Thm. 1.19 holds for g.*

Proof. Suppose that the function g is continuous and that it is supported on the interval $[-K, K]$[3].

To prove that the Thm. 1.19 holds for g, first we need to prove that g satisfies the two limits (1.7) and (1.8).

1. **Proving (1.7) for g.** If we consider a dyadic generation M large enough such that $K < 2^M$, and if $x \in [0, 2^M) \in \mathcal{D}_{-M}$, then the expectation is taken over an interval much larger than the support of g. Thus:

$$|P_{-M}g(x)| = \left| \frac{1}{2^M} \int_0^K g(t)dt \right|.$$

Applying the Cauchy-Schwarz Inequality:

$$|P_{-M}g(x)| \leq \frac{1}{2^M} \left(\int_0^K 1^2 dt \right)^{1/2} \left(\int_0^K |g(t)|^2 dt \right)^{1/2} \leq \frac{\sqrt{K}}{2^M} \|g\|_2 .$$

The same inequality holds for $x < 0$. Also, if $|x| \geq 2^M$, then $P_{-M}g(x) = 0$, because the interval in $\mathcal{D}_{-M}$ that contains x is disjoint from the support of g. We can now estimate the L^2 norm of $P_{-M}g$:

$$\begin{aligned} \|P_{-M}g\|_2^2 &= \int_{-2^M}^{2^M} |P_{-M}g(x)|^2 dx \leq \int_{-2^M}^{2^M} \left(\frac{\sqrt{K}}{2^M} \|g\|_2 \right)^2 dx \\ &= \frac{K \|g\|_2^2}{2^{2M}} \int_{-2^M}^{2^M} 1 dx = \frac{K \|g\|_2^2}{2^{2M}} \cdot 2^{M+1} = 2^{-M+1} K \|g\|_2^2 . \end{aligned}$$

Given $\epsilon > 0$, by choosing M large enough (which corresponds to $M \to \infty$), we can make $2^{-M+1} K \|g\|_2^2 < \epsilon^2$. Thus, there exists a generation M such that for all $j > M$, $\|P_{-j}g\|_2 \leq \epsilon$.

2. **Proving (1.8) for g.** We assume that g is continuous and supported on the compact interval $[-K, K]$. Let's choose M such that $[-K, K] \subset [-2^M, 2^M]$. Since g is continuous on a compact set, by the Heine-Cantor theorem, g is *uniformly continuous*. So given $\epsilon > 0$, there exists $\delta > 0$ such that:

$$|g(y) - g(x)| < \frac{\epsilon}{\sqrt{2^{M+1}}} \quad \text{whenever } |y - x| < \delta.$$

Now we choose $N > M$ large enough so that $2^{-j} < \delta$ for all $j > N$. Each point x is contained in a unique $I \in \mathcal{D}_j$, with $|I| = 2^{-j} < \delta$. Therefore, for all $y \in I$, we have $|y - x| < \delta$.

Noting that $P_j g(x) - g(x) = \frac{1}{|I|} \int_I (g(y) - g(x))dy$, applying the Integral Triangle Inequality:

$$|P_j g(x) - g(x)| \leq \frac{1}{|I|} \int_I |g(y) - g(x)| dy \leq \frac{1}{|I|} \int_I \frac{\epsilon}{\sqrt{2^{M+1}}} dy = \frac{\epsilon}{\sqrt{2^{M+1}}}.$$

[3] In $\mathbb{R}$, by the Heine-Borel Theorem, if the support of g is compact, it has to be closed and bounded, so there exists $K \in \mathbb{R}$ so that $\text{supp}(g) \subseteq [-K, K]$.

Choosing $j > N > M$, noting that if $|x| > 2^M$, namely outside $\text{supp}(g)$, the approximation error is zero (since both $P_j g(x) = 0$ and $g(x) = 0$), we can calculate the L^2 norm as follows:

$$\begin{aligned}\|P_j g - g\|_2^2 &= \int_{\mathbb{R}} |P_j g(x) - g(x)|^2 dx = \int_{|x|\leq 2^M} |P_j g(x) - g(x)|^2 dx \\ &< \int_{|x|\leq 2^M} \left(\frac{\epsilon}{\sqrt{2^{M+1}}}\right)^2 dx = \frac{\epsilon^2}{2^{M+1}} \int_{|x|\leq 2^M} 1 dx = \epsilon^2.\end{aligned}$$

Thus, we have shown that given $\epsilon > 0$, there is an $N > 0$ such that for all $j > N$,

$$\|P_j g - g\|_2 \leq \epsilon.$$

□

We can now prove Thm. 1.19.

Proof of Thm. 1.19. By Thm. A.25, given $f \in L^2(\mathbb{R})$ and $\epsilon > 0$, we can decompose f as $f = g + h$, where $g \in C_c(\mathbb{R})$ with $\text{supp}(g) \subseteq [-K, K]$ and $h \in L^2(\mathbb{R})$ with $\|h\|_2 < \epsilon/4$.

1. **Proving (1.7) for f.** By Lem. 1.21, there exists $N \in \mathbb{Z}$ so that for all $j > N$,

   $$\|P_{-j} g\|_2 \leq \epsilon/2.$$

   The expectation operators P_j are linear operators [4], so $P_j(g + h) = P_j g + P_j h$. Using the Triangle Inequality and the Lem. 1.20 we check that

   $$\|P_{-j} f\|_2 \leq \|P_{-j} g\|_2 + \|P_{-j} h\|_2 \leq \epsilon/2 + \|h\|_2 \leq \epsilon.$$

2. **Proving (1.8) for f.** By the same Lem. 1.21, there exists $N \in \mathbb{Z}$ so that for all $j > N$,

   $$\|P_j g - g\|_2 \leq \epsilon/2.$$

   Now, using the Triangle Inequaility twice and then applying Lem. 1.20,

   $$\|P_j f - f\|_2 \leq \|P_j g - g\|_2 + \|P_j h - h\|_2 \leq \epsilon/2 + 2\|h\|_2 \leq \epsilon.$$

□

Thus, we've successfully proven that the function defined at (1.3) is indeed a *wavelet*, as stated in the Def. 1.6.

[4] If we check their definition, the linearity of integrals proves it automatically.

Chapter 2

Multiresolutional Analysis based on Wavelets

This chapter generalizes the specific case of Haar wavelets into the formal framework of Multiresolution Analysis (MRA) both for the one- and two-dimensional cases.

2.1 One-dimensional MRAs

In this section, we formalize the general concept of a one-dimensional MRA and detail the specific construction of the Haar MRA.

2.1.1 The MRA Definition

The concept of *Orthogonal Multiresolution Analysis* can be informally explained as the decomposition of the $L^2(\mathbb{R})$ space into closed subspaces that satisfy several important properties. They are called, the *approximation and detail spaces*, namely V_j and W_j where j are integers.

Definition 2.1 (*Multiresolution Analysis*)**.** An *orthogonal multiresolution analysis* (or MRA) consists of a sequence of closed subspaces $\{V_j\}_{j\in\mathbb{Z}}$ of $L^2(\mathbb{R})$, referred to as *approximation spaces*, and a scaling function $\phi \in V_0$, satisfying the following properties:

1. **Nested Subspaces:** The spaces are increasing, such that $V_j \subset V_{j+1}$ for all $j \in \mathbb{Z}$.
2. **Trivial Intersection:** $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$.
3. **Completeness:** The union of the subspaces is dense in the total space, i.e., $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$ [1].
4. **Scale Invariance:** A function $f(x)$ belongs to V_j if and only if its dilation $f(2x)$ belongs to V_{j+1}.
5. **Translation:** The integer translates of the scaling function, $\{\phi(x-k)\}_{k\in\mathbb{Z}}$, form an orthonormal basis for the central space V_0.

$\Diamond$

[1] The overline is the symbol for the closure in the L^2 norm.

Associated with these approximation spaces, we define the subspaces that capture the difference in information between resolutions.

Definition 2.2 (*Detail Spaces*)**.** The *detail spaces* (or wavelet spaces), denoted by $\{W_j\}_{j\in\mathbb{Z}}$, are defined as the orthogonal complement of V_j in V_{j+1}. They satisfy the orthogonal direct sum decomposition:

$$V_{j+1} = V_j \oplus W_j. \tag{2.1}$$

Consequently, W_j contains the detail information required to increase the resolution from level j to $j+1$, such that $V_j \perp W_j$. Note that following Lem. A.11, W_j is a closed subspace of V_{j+1}. Since V_{j+1} is itself a closed subspace of $L^2(\mathbb{R})$, it immediately follows that W_j is also a closed subspace of $L^2(\mathbb{R})$. ◊

2.1.2 The Haar MRA

Using the definitions seen in the previous subsection, we can construct the Haar MRA.

Definition 2.3 (*The Haar MRA*)**.** We define the *Haar MRA* taking the characteristic function $\phi(x) = \chi_{[0,1)}(x)$ as the scaling function. The approximation subspaces V_j are defined as the space of step functions with steps on the dyadic intervals of the j^{th} generation ($\mathcal{D}_j$) that have a finite L^2-norm. ◊

It is not immediately obvious that this construction satisfies all the rigorous requirements of Def. 2.1. We verify this in the following proposition.

Proposition 2.4. *The spaces V_j and the function ϕ defined in Def. 2.3 constitute a valid Multiresolution Analysis.*

Proof. We verify the five conditions of Def. 2.1:

1. **Nested Subspaces:** A step function constant on intervals in $\mathcal{D}_j$ is essentially a step function constant on intervals in $\mathcal{D}_{j+1}$, since any interval $I \in \mathcal{D}_j$ is the union of two intervals in $\mathcal{D}_{j+1}$ (its children) taking the same value. Thus, $V_j \subset V_{j+1}$.

2. **Trivial Intersection:** Recall the expectation operators P_j from Def. 1.15. The range of P_j is exactly V_j. Indeed, by definition, $P_j f$ is constant on any $I \in \mathcal{D}_j$, so $\text{Range}(P_j) \subseteq V_j$. Conversely, for any $g \in V_j$, since g is already constant on dyadic intervals of level j, its average over such intervals is the value of the function itself, implying $P_j g = g$. Thus, $V_j = \{P_j f : f \in L^2(\mathbb{R})\}$.

   If $f \in \bigcap_{j\in\mathbb{Z}} V_j$, then $f \in V_j$ for all j, which implies $f = P_j f$ for all j (specifically as $j \to -\infty$). By equation (1.7), we know that $\lim_{j\to-\infty} \|P_j f\|_2 = 0$. Since $\|f\|_2 = \|P_j f\|_2$ for all j, it follows that $\|f\|_2 = 0$, so $f = 0$.

3. **Completeness:** This relies similarly on equation (1.8). Since $P_j f \in V_j$ as previously shown, and $\lim_{j\to\infty} \|P_j f - f\|_2 = 0$, any function $f \in L^2(\mathbb{R})$ can be approximated arbitrarily well by elements of $\bigcup_{j\in\mathbb{Z}} V_j$. Thus, the union is dense in $L^2(\mathbb{R})$.

4. **Scale Invariance:** The dilation $x \mapsto 2x$ implies that if x traverses an interval of length $2^{-(j+1)}$, then $2x$ traverses an interval of length 2^{-j}. Therefore, a function is constant on intervals of generation $j+1$ if and only if its dilated version is constant on intervals of generation j. Hence, $f(x) \in V_j \iff f(2x) \in V_{j+1}$.

5. **Translation:** The family $\{\phi(x-k)\}_{k\in\mathbb{Z}} = \{\chi_{[k,k+1)}(x)\}_{k\in\mathbb{Z}}$ forms an orthonormal basis for V_0.

   (a) **Orthogonality**: The supports $[k, k+1)$ are disjoint for distinct integers k, so the inner product is zero.

   (b) **Normality:** $\left\|\chi_{[k,k+1)}\right\|_2^2 = \int_k^{k+1} 1dx = 1$.

   (c) **Completeness in V_0:** Let $f \in V_0$. By definition, f is constant on each interval $[k, k+1)$. Thus, f can be written as $f(x) = \sum_{k\in\mathbb{Z}} c_k\chi_{[k,k+1)}(x)$ for some sequence $\{c_k\}$. Since $f \in L^2$, this sequence is in ℓ^2. For any $l \in \mathbb{Z}$:

$$\langle f, \chi_{[l,l+1)}\rangle = \sum_{k\in\mathbb{Z}} c_k\langle\chi_{[k,k+1)}, \chi_{[l,l+1)}\rangle = \sum_{k\in\mathbb{Z}} c_k\delta_{kl} = c_l.$$

   This confirms that the coefficients correspond to the Fourier coefficients with respect to this system, proving it is a basis. In other words, the system satisfies the reconstruction formula, as in a similar already explained case (Thm. 1.14).

□

2.1.3 The Dilation Equation

We know that the integer translates of the scaling function of a MRA form an orthonormal basis of V_0, but something similar happens to the integer translates of the dilates of said function.

Lemma 2.5. *Given a MRA with scaling function ϕ, the family of integer translates and dilates $\{\phi_{j,k}\}_{k\in\mathbb{Z}} = \{2^{j/2}\phi(2^jx-k)\}_{k\in\mathbb{Z}}$ forms an orthonormal basis for V_j.*

Proof. Given $j \in \mathbb{Z}$, we check the required properties:

1. **Orthonormality**: For $k, l \in \mathbb{Z}$, using the change of variables $y = 2^jx$ (so $dx = 2^{-j}dy$):

$$\begin{aligned}\langle\phi_{j,k}, \phi_{j,l}\rangle &= \int_{\mathbb{R}} 2^{j/2}\phi(2^jx-k)2^{j/2}\overline{\phi(2^jx-l)}dx = 2^j\int_{\mathbb{R}}\phi(y-k)\overline{\phi(y-l)}\frac{dy}{2^j}\\ &= \langle\phi_{0,k}, \phi_{0,l}\rangle = \delta_{k,l}.\end{aligned}$$

2. **Completeness in V_j:** Let $f \in V_j$. By the scaling invariance property of the MRA, the function $g(x) := f(2^{-j}x)$ belongs to V_0. Since $\{\phi_{0,k}\}_{k\in\mathbb{Z}} = \{\phi(x-k)\}_{k\in\mathbb{Z}}$ forms an orthonormal basis for V_0, we can expand g as:

$$g(x) = \sum_{k\in\mathbb{Z}} c_k\phi(x-k), \quad \text{where } c_k = \langle g, \phi_{0,k}\rangle.$$

   Now, substituting back to recover f (setting $x \to 2^jx$):

$$f(x) = g(2^jx) = \sum_{k\in\mathbb{Z}} c_k\phi(2^jx-k) = \sum_{k\in\mathbb{Z}} c_k2^{-j/2}\phi_{j,k}(x).$$

To verify that these coefficients are indeed the Fourier coefficients with respect to the new basis, we compute the inner product using the change of variables $y = 2^j x$:

$$\begin{aligned}\langle f, \phi_{j,k}\rangle &= \int_{\mathbb{R}} g(2^j x) 2^{j/2} \overline{\phi(2^j x - k)} dx = 2^{j/2} 2^{-j} \int_{\mathbb{R}} g(y) \overline{\phi(y-k)} dy \\ &= 2^{-j/2} \langle g, \phi_{0,k}\rangle = 2^{-j/2} c_k.\end{aligned}$$

Thus, we have explicitly shown that $f = \sum_{k\in\mathbb{Z}} \langle f, \phi_{j,k}\rangle \phi_{j,k}$, proving that the family is an orthonormal basis for V_j.

□

Note that given an MRA and its scaling function ϕ, we can determine all the approximation spaces. These spaces are generated by the integer translates of the dyadic dilates of the scaling function, as proven in Lem. 2.5.

Given the nested property of the approximation spaces, we can define a fundamental relationship known as the dilation equation.

Definition 2.6 (*The dilation equation*)**.** The scaling function ϕ lies in the central space V_0. Since $V_0 \subset V_1$, it follows that $\phi \in V_1$. By Lem. 2.5, we know that the family $\{\phi_{1,k}(x) := \sqrt{2}\phi(2x-k)\}_{k\in\mathbb{Z}}$ forms an orthonormal basis for V_1. Thus, we can write ϕ in terms of said basis as follows:

$$\phi(x) = \sum_{k\in\mathbb{Z}} h_k \phi_{1,k}(x) = \sqrt{2} \sum_{k\in\mathbb{Z}} h_k \phi(2x-k). \tag{2.2}$$

We call this the *dilation equation* (or *scaling equation*). The sequence of coefficients $\{h_k\}_{k\in\mathbb{Z}}$ belongs to $\ell^2(\mathbb{Z})$ and is called the *low-pass filter* (or scaling filter) of the MRA. ◊

This equation is fundamental because it tells us how to construct the approximation from the finer approximation. In the case of the Haar MRA, this relationship is very simple.

Proposition 2.7 (*The Haar dilation equation*)**.** *The Haar MRA holds the following dilation equation:*

$$\phi = \frac{\phi_{1,0} + \phi_{1,1}}{\sqrt{2}}. \tag{2.3}$$

Proof.

$$\begin{aligned}\frac{\phi_{1,0}(t) + \phi_{1,1}(t)}{\sqrt{2}} &= \frac{\sqrt{2}\phi(2t) + \sqrt{2}\phi(2t-1)}{\sqrt{2}} = \phi(2t) + \phi(2t-1) \\ &= \chi_{[0,1/2)}(t) + \chi_{[1/2,1)}(t) = \chi_{[0,1)}(t) = \phi(t).\end{aligned}$$

□

In fact, given the properties of the scaling function, we can see that this equation is applicable to all the translates and dilates as follows:

Proposition 2.8 (*The generalized dilation equation*)**.** *Given the scaling function $\phi \in V_0$ and the low-pass filter $\{h_n\}_{n\in\mathbb{Z}} \subset \ell^2(\mathbb{Z})$ of a MRA, the generalized dilation equation holds for $j,k \in \mathbb{Z}$:*

$$\phi_{j,k} = \sum_{n\in\mathbb{Z}} h_n \phi_{j+1,n+2k}. \tag{2.4}$$

Proof. Applying the dilation equation (2.2) and substituting the variables:

$$\begin{aligned}\phi_{j,k}(x) &= 2^{j/2}\phi\left(2^j x - k\right) = 2^{j/2}\left(\sqrt{2}\sum_{n\in\mathbb{Z}} h_n \phi(2(2^j x - k) - n)\right) \\ &= \sum_{n\in\mathbb{Z}} h_n 2^{(j+1)/2}\phi\left(2^{j+1}x - (2k+n)\right) = \sum_{n\in\mathbb{Z}} h_n \phi_{j+1,n+2k}(x).\end{aligned}$$

□

Thus, we can extract the generalized Haar dilation equation as follows:

Corollary 2.9 (*The generalized Haar dilation equation*)**.** *The Haar MRA holds the following generalized dilation equation:*

$$\phi_{j,k} = \frac{\phi_{j+1,2k} + \phi_{j+1,2k+1}}{\sqrt{2}}. \tag{2.5}$$

◊

2.1.4 The Projection and Difference Operators

In Sec. 1.2.3, we defined the operator P_j for the Haar case as an expectation operator that computed the average of a function over dyadic intervals. We observed that this operator produced an approximation of the function at the scale 2^{-j}.

In the context of a general Multiresolution Analysis, we can generalize this concept. Since V_j is a closed subspace of $L^2(\mathbb{R})$, we can define P_j not just as an averaging operator, but geometrically as the *orthogonal projection* onto V_j. It is worth noting that in the specific case of the Haar MRA, where V_j is the space of piecewise constant functions, this geometric definition coincides exactly with the averaging definition given previously.

Definition 2.10. Given $f \in L^2(\mathbb{R})$, we define $P_j f$ as the orthogonal projection of f onto the subspace V_j. Since $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$ constitutes an orthonormal basis for V_j, we can express this projection explicitly as:

$$P_j f(x) := \sum_{k\in\mathbb{Z}} \langle f, \phi_{j,k}\rangle \phi_{j,k}(x). \tag{2.6}$$

◊

Since V_j is a *closed subspace* of $L^2(\mathbb{R})$, we can directly apply the *Orthogonal Projection Theorem* (Thm. A.16), which guarantees two fundamental properties:

- First, that $P_j f$ is the unique *best approximation* to f within the subspace V_j in the L^2-norm sense.

- Second, that the approximation error is orthogonal to V_j, which paves the way for the definition of the detail spaces W_j as the orthogonal complement of V_j in V_{j+1}.

Furthermore, the nested property of the MRA ($V_j \subset V_{j+1}$) implies that $P_{j+1}f$ provides a better (or at least equal) approximation than $P_j f$. Just as we did in the Haar analysis, we are interested in capturing the information lost when moving from a finer resolution $j+1$ to a coarser resolution j.

Definition 2.11. We define the *difference operator* Q_j as:

$$Q_j f := P_{j+1} f - P_j f. \tag{2.7}$$

$\Diamond$

While this definition mirrors the one from the Haar wavelet construction, its geometric interpretation in a general MRA is quite powerful: $Q_j f$ represents the orthogonal projection of f onto W_j, the orthogonal complement of V_j in V_{j+1}. This result is proven in the next lemma.

Proposition 2.12. *Given $f \in L^2(\mathbb{R})$, the difference $Q_j f$ is the orthogonal projection of f onto the detail space W_j.*

Proof. First, since $Q_j f = P_{j+1} f - P_j f$, and both terms belong to V_{j+1} (as $V_j \subset V_{j+1}$), it is clear that $Q_j f \in V_{j+1}$.

To prove that $Q_j f$ is the orthogonal projection onto W_j, we must show that $Q_j f \in W_j$, which means showing that $Q_j f \perp V_j$. It suffices to prove that the projection of $Q_j f$ onto V_j is zero, i.e., $P_j(Q_j f) = 0$.

Using the definition $Q_j = P_{j+1} - P_j$ and the linearity of the projection operator:

$$P_j(Q_j f) = P_j(P_{j+1} f - P_j f) = P_j P_{j+1} f - P_j^2 f.$$

We use two key properties of orthogonal projections:

1. **Idempotence:** $P_j^2 = P_j$ (projecting twice is the same as projecting once).
2. **Nested Projections:** Since $V_j \subset V_{j+1}$, projecting onto the larger space V_{j+1} and then onto the subspace V_j is equivalent to projecting directly onto V_j. Thus, $P_j P_{j+1} = P_j$.

Substituting these into our equation:

$$P_j(Q_j f) = P_j f - P_j f = 0.$$

Since $P_j(Q_j f) = \sum_k \langle Q_j f, \phi_{j,k} \rangle \phi_{j,k} = 0$, and $\{\phi_{j,k}\}$ is a basis, it implies that $\langle Q_j f, \phi_{j,k} \rangle = 0$ for all k. Thus, $Q_j f \perp V_j$. Thus, since $Q_j f \in V_{j+1}$ and $Q_j f \perp V_j$, by definition $Q_j f$ belongs to the orthogonal complement W_j.

Finally, to verify it is indeed the projection of f onto W_j, we check if the residual $f - Q_j f$ is orthogonal to W_j. Decomposing f:

$$f - Q_j f = f - (P_{j+1} f - P_j f) = (f - P_{j+1} f) + P_j f.$$

Let $w \in W_j$. Since $W_j \subset V_{j+1}$, $f - P_{j+1} f$ is orthogonal to w (since it is orthogonal to the entire subspace V_{j+1}). Since $W_j \perp V_j$ and $P_j f \in V_j$, $P_j f$ is also orthogonal to w. Thus, $\langle f - Q_j f, w \rangle = 0$ for all $w \in W_j$. $\square$

2.1.5 From MRA to Wavelets

The detail spaces hold an interesting property, stated next.

Proposition 2.13. *The detail spaces are pairwise orthogonal, namely, given $j < k \in \mathbb{Z}$, $W_j \perp W_k$.*

Proof. Assume without loss of generality that $j < k$. Since the MRA spaces are nested, $V_{j+1} \subset V_k$. By definition, $W_j \subset V_{j+1}$, so $W_j \subset V_k$. Also by definition, $W_k \perp V_k$ (in V_{k+1}), which immediately implies that $W_j \perp W_k$. □

We can thus express the approximation subspaces in terms of less accurate approximation spaces and the detail subspaces at the intermediate resolutions. This means that for $n < j$ applying the recursion:

$$V_j = V_{j-1} \oplus W_{j-1} = \cdots = V_n \oplus W_n \oplus W_{n+1} \oplus \cdots \oplus W_{j-2} \oplus W_{j-1}.$$

Thus, given the approximation subspaces $\{V_j\}_j \in \mathbb{Z}$ of an MRA and their density condition (item 3 of 2.1), we have the following corollary:

Corollary 2.14. *Given an MRA, the direct sum of its detail subspaces is dense in L^2, i.e.,*

$$\overline{\bigoplus_{j\in\mathbb{Z}} W_j} = L^2(\mathbb{R}). \tag{2.8}$$

Proof. Given that the infinite intersection of the approximation spaces is trivial (Item 2 of Def. 2.1), we have that $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$. This implies that the residual space V_M vanishes as $M \to -\infty$. Thus, the finite decomposition $V_j = V_M \oplus \bigoplus_{k=M}^{j-1} W_k$ becomes:

$$V_j = \bigoplus_{k=-\infty}^{j-1} W_k.$$

Finally, using the density of the union (Item 3 of said definition) and the fact that they are nested, we immediately see that:

$$L^2(\mathbb{R}) = \overline{\bigcup_{j\in\mathbb{Z}} V_j} = \overline{\bigoplus_{j\in\mathbb{Z}} W_j}.$$

□

An extremely important result, which provides the connection from MRAs to the wavelets, is Mallat's Theorem. Given the scope of this thesis, we will only state it here.

Theorem 2.15 (*Mallat's Theorem*)**.** *[29] Given an orthogonal MRA with scaling function ϕ, there exists a wavelet $\psi \in L^2(\mathbb{R})$ such that for each $j \in \mathbb{Z}$, the family $\{\psi_{j,k} := 2^{j/2}\psi(2^j x - k)\}_{k\in\mathbb{Z}}$ is an orthonormal basis for W_j. Hence, the family $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ forms an orthonormal basis for $L^2(\mathbb{R})$.* ♢

Finally, using all the theory presented in this chapter, we can appreciate a perfect symmetry. Given the scaling function ϕ of an MRA, its integer translates and dilates form a basis of the approximation spaces V_j, with P_j serving as the projection operator onto said spaces. Conversely, Mallat's Theorem guarantees the existence of a wavelet function ψ for the same MRA, whose integer translates and dilates form a basis of the detail spaces W_j, with Q_j serving as the projection operator onto said spaces.

Remark 2.16. For the Haar MRA defined in 2.3, the wavelet ψ guaranteed by Mallat's Theorem is precisely the Haar wavelet $h(x) = -\chi_{[0,1/2)} + \chi_{[1/2,1)}$ introduced in Def. 1.6. ◊

2.1.6 The Wavelet Equation

As stated in Mallat's Theorem (Thm. 2.15), the wavelet ψ lies in the detail space W_0. Since $W_0 \subset V_1$, it follows that $\psi \in V_1$. Just as we did with the scaling function in the dilation equation (Def. 2.6), we can express ψ in terms of the basis of V_1, namely $\{\phi_{1,k}\}_{k\in\mathbb{Z}}$.

Definition 2.17 (*The wavelet equation*)**.** We define the *wavelet equation* as the expansion of the wavelet ψ in the finer scaling basis:

$$\psi(x) = \sum_{k\in\mathbb{Z}} g_k \phi_{1,k}(x) = \sqrt{2} \sum_{k\in\mathbb{Z}} g_k \phi(2x-k). \tag{2.9}$$

The sequence of coefficients $\{g_k\}_{k\in\mathbb{Z}}$ belongs to $\ell^2(\mathbb{Z})$ and is called the *high-pass filter* (or wavelet filter) of the MRA. ◊

This equation is fundamental because it tells us how to construct the details from the finer approximation. In the case of the Haar MRA, this relationship is very simple.

Proposition 2.18 (*The Haar wavelet equation*)**.** *The Haar wavelet h satisfies the following equation:*

$$h = \frac{-\phi_{1,0} + \phi_{1,1}}{\sqrt{2}}. \tag{2.10}$$

Proof.

$$\begin{aligned}\frac{-\phi_{1,0}(t) + \phi_{1,1}(t)}{\sqrt{2}} &= \frac{-\sqrt{2}\phi(2t) + \sqrt{2}\phi(2t-1)}{\sqrt{2}} = -\phi(2t) + \phi(2t-1) \\ &= -\chi_{[0,1/2)}(t) + \chi_{[1/2,1)}(t) = \psi(t).\end{aligned}$$

□

In fact, given the properties of the scaling function, we can see that this equation is applicable to all the translates and dilates as follows:

Proposition 2.19 (*The generalized wavelet equation*)**.** *Given the wavelet function $\psi \in W_0$ and the high-pass filter $\{g_n\}_{n\in\mathbb{Z}} \subset \ell^2(\mathbb{Z})$ of a MRA, the generalized wavelet equation holds for $j, k \in \mathbb{Z}$:*

$$\psi_{j,k} = \sum_{n\in\mathbb{Z}} g_n \phi_{j+1,n+2k}. \tag{2.11}$$

Proof. Analogous to the proof of the generalized dilation equation (Prop. 2.8). □

Thus, we can extract the generalized Haar wavelet equation as follows:

Corollary 2.20 (*The generalized Haar wavelet equation*)**.** *The Haar MRA holds the following generalized wavelet equation:*

$$\psi_{j,k} = \frac{-\phi_{j+1,2k} + \phi_{j+1,2k+1}}{\sqrt{2}}. \tag{2.12}$$

◊

Remark 2.21. Comparing the Haar dilation equation (from Prop. 2.7) and the Haar wavelet equation (from Prop. 2.18), we can identify the coefficients for the Haar filters:

- Low-pass filter (h_k): $h_0 = \frac{1}{\sqrt{2}}, h_1 = \frac{1}{\sqrt{2}}$ (and 0 otherwise).
- High-pass filter (g_k): $g_0 = -\frac{1}{\sqrt{2}}, g_1 = \frac{1}{\sqrt{2}}$ (and 0 otherwise).

This explicit knowledge of the filters is the key to the fast algorithms used in signal processing. ◊

2.2 Two-dimensional MRAs

In this section, we formalize the two-dimensional extension of the one-dimensional MRA and detail the construction of the two-dimensional Haar MRA.

2.2.1 Two-dimensional MRAs Construction

If we want to apply the concept of MRAs to two-dimensional functions, we need to expand the theory from $L^2(\mathbb{R})$ to $L^2(\mathbb{R}^2)$. This is easily managed using the *tensor product*.

Proposition 2.22 (*Two-dimensional wavelets*)**.** *Given a wavelet basis $\{\psi_{j,k}\}_{j,k\in\mathbb{Z}}$ for $L^2(\mathbb{R})$ the family of tensor products*

$$\{\psi_{j,k;i,n}(x,y) := \psi_{j,k}(x)\psi_{i,n}(y)\}_{j,k,i,n\in\mathbb{Z}} \tag{2.13}$$

is an orthonormal basis in $L^2(\mathbb{R}^2)$.

Proof. This is a direct consequence of the Thm. A.37. □

In fact, more generally, Lem. A.35 gives us the following proposition.

Proposition 2.23. *Given an orthonormal basis $\{\phi_n\}_{n\in\mathbb{Z}}$ of a closed subspace $V \subseteq L^2(\mathbb{R})$, then the functions defined on $\mathbb{R}^2$ by*

$$\phi_{m,n}(x,y) = \phi_m(x)\phi_n(y) \quad m,n \in \mathbb{Z}$$

form an orthonormal basis of the tensor product space denoted as $V \otimes V$ i.e., $V \otimes V = \overline{\mathrm{span}(\{\Phi_{\mathrm{m,n}}\}_{\mathrm{m,n}\in\mathbb{Z}})}$. ◊

Thus, we can apply this for every approximation space V_j in a given MRA. As we recall, for every scale or generation $j \in \mathbb{Z}$, the family $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$ is an orthonormal basis of the subspace $V_j \subseteq L^2(\mathbb{R})$. Consequently, the family of tensor products $\{\phi_{j;k,n}(x,y) := \phi_{j,k}(x)\phi_{j,n}(y)\}_{j,k,m\in\mathbb{Z}}$ is an orthonormal basis of the tensor product space $\mathcal{V}_j := V_j \otimes V_j$.

One can see that the spaces $\mathcal{V}_j$ form a new tensor MRA defined in the $L^2(\mathbb{R}^2)$ space, with scaling function

$$\phi(x,y) = \phi(x)\phi(y),$$

where $\phi \subseteq V_0$ in the right-hand side is the scaling function of the original MRA. Thus, as we constructed the detail spaces W_j as the orthogonal complements of V_j in V_{j+1}, we can define the detail spaces $\mathcal{W}_j$ of the tensor MRA as the orthogonal complement of $\mathcal{V}_j$ in $\mathcal{V}_{j+1}$. We can express $\mathcal{W}_j$ in terms of the spaces V_j and W_j as follows.

Proposition 2.24.

$$\mathcal{W}_j = (W_j \otimes W_j) \oplus (W_j \otimes V_j) \oplus (V_j \otimes W_j).$$

Proof. Applying Prop. A.36 repeatedly, we can operate as follows:

$$\begin{aligned}\mathcal{V}_{j+1} &= V_{j+1} \otimes V_{j+1} = (V_j \oplus W_j) \otimes (V_j \oplus W_j) \\ &= (V_j \otimes V_j) \oplus [(V_j \otimes W_j) \oplus (W_j \otimes V_j) \oplus (W_j \otimes W_j)] = \mathcal{V}_j \oplus \mathcal{W}_j.\end{aligned}$$

□

Therefore in the tensor MRA we need *three wavelets* to span the detail space $\mathcal{W}_0$, each of them spanning one of the three tensor spaces that are in the direct sum. We can proceed knowing that ϕ and ψ are the scaling function and wavelet function that span V_0 and W_0 respectively. The wavelets are constructed as:

$$\begin{aligned}\psi^d(x,y) &:= \psi(x)\psi(y) \quad \text{for } (W_0 \otimes W_0), \\ \psi^v(x,y) &:= \psi(x)\phi(y) \quad \text{for } (W_0 \otimes V_0) \text{ and} \\ \psi^h(x,y) &:= \phi(x)\phi(y) \quad \text{for } (V_0 \otimes V_0),\end{aligned}$$

where the d stands for diagonal, v for vertical and h for horizontal wavelets. The direction assigned to each wavelet is the direction of details which the corresponding wavelet is better at detecting.

2.2.2 The Two-dimensional Haar MRA

Using the presented theory, we can construct a two-dimensional Haar MRA as follows.

Definition 2.25 (*The Two-dimensional Haar MRA*)**.** The two-dimensional Haar MRA is generated from the scaling function

$$\phi(x,y) = \phi(x)\phi(y) = \chi_{[0,1)}(x)\chi_{[0,1)}(y) = \chi_{[0,1)^2}(x,y).$$

◊

One can see thus, dilating and translating the scaling function, one can find the approximation spaces.

Definition 2.26. The approximation spaces $\mathcal{V}_j$ for the two-dimensional Haar MRA are the spaces of $\mathbb{R}^2$ functions that are constant in the squares $[2^{-j}k, 2^{-j}(k+1)) \times [2^{-j}n, 2^{-j}(n+1))$ for every $k, n \in \mathbb{Z}$ with finite L^2-norm. ◊

Similarly to the scaling function, we can also construct the three wavelet functions.

Definition 2.27. The three wavelet functions of the two-dimensional Haar are the following:

$$\begin{aligned}\psi^d(x,y) := \psi(x)\psi(y) &= \left(-\chi_{[0,1/2)}(x) + \chi_{[1/2,1)}(x)\right)\left(-\chi_{[0,1/2)}(y) + \chi_{[1/2,1)}(y)\right) \\ &= \chi_{[0,1/2)^2}(x,y) + \chi_{[1/2,1)^2}(x,y) - \chi_{[0,1/2)\times[1/2,1)}(x,y) - \chi_{[1/2,1)\times[0,1/2)}(x,y),\end{aligned}$$

$$\begin{aligned}\psi^v(x,y) := \psi(x)\phi(y) &= \left(-\chi_{[0,1/2)}(x) + \chi_{[1/2,1)}(x)\right)\chi_{[0,1)}(y) \\ &= -\chi_{[0,1/2)\times[0,1)}(x,y) + \chi_{[1/2,1)\times[0,1)}(x,y),\end{aligned}$$

$$\psi^h(x,y) := \phi(x)\psi(y) = \chi_{[0,1)}(x)\left(-\chi_{[0,1/2)}(y) + \chi_{[1/2,1)}(y)\right)$$
$$= -\chi_{[0,1)\times[0,1/2)}(x,y) + \chi_{[0,1)\times[1/2,1)}(x,y).$$

$\Diamond$

To gain a better understanding of the scaling and wavelet functions, we visualize them in Fig. 2.1. The figure illustrates the spatial support and sign changes of the given functions. ψ^v highlights vertical edges, as they change horizontally, ψ^h detects horizontal edges, as they change vertically, and ψ^d captures diagonal details.

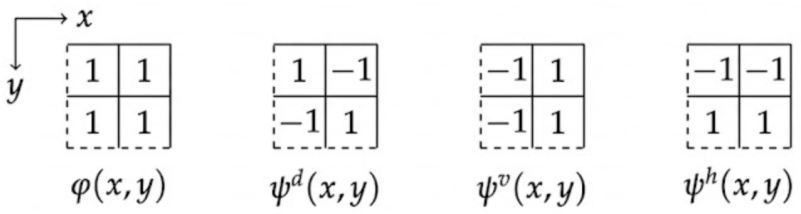


Figure 2.1: Visualization of the 2D Haar MRA scaling and wavelet functions. We visualize them within their support $[0,1)^2$. Here, we assign values to the constant regions, defining the positive axes orientation from top to bottom and left to right.

Following the one- and two-dimensional Haar MRAs, we can construct algorithms to apply them to discrete spaces.

Chapter 3

The Fast Haar Transform and its 2D Construction

In the previous chapters, we define and justify the Haar Wavelet and the Haar MRA, uniquely determining one another. To be able to use this multiresolution analysis in the image world, we need to define how to apply it in the discrete 1D and 2D space.

3.1 The Fast Haar Transform

3.1.1 The FHT Construction

The discretization of a function $f \in L^2(\mathbb{R})$ is a vector $v \in \mathbb{C}^N$ where $N \in \mathbb{Z}$ indicates the number of different samples we take of said function. From now on in this chapter, these values will correspond to the mean value of the function on each of the intervals of $\mathcal{D}_j$, given a generation $j \in \mathbb{Z}$.

Recall that in the Haar MRA (Def. 2.3), the space $V_0 \subset L^2(\mathbb{R})$ is formed by the step functions that are constant in each interval of the form $[k, k+1)$ for every integer k with finite L^2-norm. Thus, we can discretize a function $f \in V_0$ as mentioned earlier, getting one sample in each step, with the following property.

Proposition 3.1. *If $f, g \in V_0$ are supported on $[-K, K]$, discretized as previously mentioned by the vectors $v, w \in \mathbb{C}^{2K}$, the $L^2(\mathbb{R})$ inner product $\langle f, g\rangle_{L^2(\mathbb{R})}$ coincides with the standard dot product $v \cdot w$.*

Proof. By definition $f = \sum_{k=-K}^{K-1} c_k \chi_{[k,k+1)}$ and $g = \sum_{k=-K}^{K-1} d_k \chi_{[k,k+1)}$, with discretization vectors $v = [c_{-K}, \ldots, c_{K-1}]$ and $w = [d_{-K}, \ldots, d_{K-1}]$. We thus get the following:

$$\begin{aligned}
\langle f, g\rangle_{L^2(\mathbb{R})} &= \int_{\mathbb{R}} f\overline{g} = \int_{\mathbb{R}} \left(\sum_{k=-K}^{K-1} c_k \chi_{[k,k+1)}\right) \overline{\left(\sum_{l=-K}^{K-1} d_l \chi_{[l,l+1)}\right)} \\
&= \sum_{k=-K}^{K-1} c_k \overline{d_k} \int_{\mathbb{R}} \chi^2_{[k,k+1)} = \sum_{k=-K}^{K-1} c_k \overline{d_k} \\
&= v \cdot w.
\end{aligned}$$

□

We can generalize this for every approximation space V_j. Since the intervals in $\mathcal{D}_j$ have length 2^{-j}, the inner product will carry a scaling factor.

Proposition 3.2. *If $f, g \in V_j$ are supported on $[-L, L]$ and discretized by vectors $v, w \in \mathbb{C}^N$ (where N is the number of dyadic intervals of length 2^{-j} in the support), the $L^2(\mathbb{R})$ inner product $\langle f, g\rangle_{L^2(\mathbb{R})}$ coincides with the weighted product $2^{-j}(v \cdot w)$.*

Proof. Let the support be partitioned by disjoint intervals $\{I_k\}_{k\in J} \subset \mathcal{D}_j$ where J is a finite subset of $\mathbb{N}$. By definition $f = \sum_{k\in J} v_k \chi_{I_k}$ and $g = \sum_{k\in J} w_k \chi_{I_k}$. Then:

$$\langle f, g\rangle_{L^2(\mathbb{R})} = \int_{\mathbb{R}} f\overline{g} = \sum_{k\in J} v_k \overline{w_k} \int_{I_k} 1\, dx = \sum_{k\in J} v_k \overline{w_k} \cdot 2^{-j} = 2^{-j}(v \cdot w).$$

□

Remark 3.3. The same logic applies to the inner product between functions of the detail spaces W_j. Since functions in W_j are also step functions on intervals of $\mathcal{D}_{j+1}$, their inner products can be computed discretely using the weight $2^{-(j+1)}$. ◊

Thus, we can calculate the projection operator P_j (Def. 2.10) onto the V_j space of the Haar MRA without having to calculate integrals, simply by operating on vectors.

Here is where the Haar general dilation and wavelet equations (Propositions 2.9 and 2.20) come into play. Applying the inner product to them gives us the recurrence relations for the coefficients. Let $a_{j,k} := \langle f, \phi_{j,k}\rangle$ be the approximation coefficients and $d_{j,k} := \langle f, \psi_{j,k}\rangle$ be the detail coefficients,

$$a_{j,k} = \frac{a_{j+1,2k} + a_{j+1,2k+1}}{\sqrt{2}} \quad (\textit{average} \text{ of } f \text{ at scale } j), \tag{3.1}$$

$$d_{j,k} = \frac{-a_{j+1,2k} + a_{j+1,2k+1}}{\sqrt{2}} \quad (\textit{difference} \text{ of } f \text{ at scale } j). \tag{3.2}$$

It is important to distinguish the roles of the derived coefficients. The difference coefficients $d_{j,k}$ are formally known as the *Haar Wavelet Coefficients*. Conversely, the approximation coefficients $a_{j,k} = \langle f, \phi_{j,k}\rangle$ are formally known as the *Haar Moments* of the function f at scale j.

In the context of the FHT, as shown in the example, the Haar Moments are primarily used as intermediate steps to calculate the coarser Wavelet coefficients following the MRA decomposition. However, one might also take an interest in the approximations themselves, viewing them as a lower-resolution representation of the signal.

Remark 3.4 (Haar Moments and Average Pooling)**.** From a signal processing perspective, calculating the Haar moment is equivalent to computing a local average. In the context of Convolutional Neural Networks (CNNs) for computer vision, this operation is mathematically analogous to the widely known *Average Pooling* layer. ◊

Remark 3.5. Regarding Equation (3.1), one might initially question the consistency of the discretization spaces, as the left-hand side involves scale j (dimension N) while the right-hand side involves scale $j + 1$ (dimension $2N$). This apparent mismatch is resolved by the nested property $V_j \subset V_{j+1}$. A function residing in V_j is automatically an element of V_{j+1}. Therefore, theoretically, we can view the left-hand side functions as elements of the finer space $\mathbb{C}^{2N}$ (where the values appear as repeated pairs). When both sides are analysed in this common finer discretization space, the dimensions align and the integration weights ($2^{-(j+1)}$) cancel out. However, for the sake of memory efficiency and given that in the right-hand side we need the correct discretization to continue the recursion, we do not store these repeated pairs. A strictly analogous logic applies to the difference equation (3.2), since $W_j \subset V_{j+1}$. ◊

Thus, we can conclude that we do not need to calculate integrals nor even full vector inner products repeatedly. We can simply apply the recurrence equations (3.1) and (3.2). By starting with the samples at a fine scale J, we can efficiently compute all averages and differences for coarser scales.

Remark 3.6. Complexity Note: This recursive calculation is extremely efficient. Since each coefficient at scale j is computed using only two coefficients from scale $j+1$, the total number of operations is proportional to the total number of samples N. This gives the Fast Haar Transform a linear complexity of $O(N)$, which is faster than the classic Fast Fourier Transform (FFT) complexity of $O(N \log N)$. ◊

3.1.2 Convolutional Formulation of the FHT

In the previous section, we derived the recurrence relations (3.1) and (3.2) to compute the approximation and detail coefficients. While these equations describe scalar operations on pairs of coefficients, efficient algorithmic implementations, particularly in signal processing and Deep Learning, rely on vector operations. Specifically, these recurrences can be re-interpreted as filtering operations.

To formalize this connection, we must introduce the fundamental operations of discrete convolution and sampling stride. We present these definitions in the general n-dimensional case ($\mathbb{Z}^n$), as they apply equally to the 1D signals discussed here and the 2D images we will address later.

Definition 3.7 (Discrete Convolution in $\mathbb{Z}^n$)**.** *Let $x, h \in \ell^2(\mathbb{Z}^n)$ be two sequences of complex numbers indexed by $\mathbb{Z}^n$. Their discrete convolution, denoted by $(x * h)$, is defined as the sequence whose s-th element ($s \in \mathbb{Z}^n$) is given by:*

$$(x * h)[s] = \sum_{m \in \mathbb{Z}^n} x[s-m]h[m] = \sum_{k \in \mathbb{Z}^n} x[k]h[s-k]. \tag{3.3}$$

◊

To optimize the evaluation of the recurrence relations (3.1) and (3.2), we utilize the filter sequences identified in the Haar MRA (Rem. 2.21: the low-pass filter h and the high-pass filter g. The calculation of coefficients is mathematically equivalent to applying these filters to the signal via convolution. However, the standard convolution definition (3.3) inherently reflects the kernel (index $s-m$), while the recurrence relations (which represent inner products) do not. To reconcile this discrepancy and correctly implement the recurrences using the efficient convolution operator, we must pre-process the filters.

Definition 3.8 (Conjugate Flip)**.** *Let $h = \{h_k\}_{k \in \mathbb{Z}^n}$ be a filter sequence. The conjugate flip of h, denoted by $\tilde{h}$, is defined as:*

$$\tilde{h}[k] := \overline{h[-k]}. \tag{3.4}$$

◊

By substituting $\tilde{h}$ into the convolution formula, the flip inherent in the convolution cancels with the manual flip in $\tilde{h}$, resulting in the correct sliding dot product required by the MRA projections.

Finally, the transition from a finer scale V_{j+1} to a coarser scale V_j implies a reduction in resolution. In our recurrence equations, this was visible as taking indices $2k$ and $2k+1$. In signal processing, this is formalized as a stride.

Definition 3.9 (Strided Convolution)**.** *Given a stride integer factor $S \geq 1$, the strided convolution is defined as the evaluation of the standard convolution at indices which are multiples of S. For a signal x and a filter $\tilde{h}$:*

$$y[s] = (x * \tilde{h})[s \cdot S]. \tag{3.5}$$

Mathematically, this is equivalent to performing a full convolution followed by a downsampling *step where we retain only every S-th sample in each dimension.* ◊

This reformulation leads us to the fundamental algorithmic equivalence used in efficient implementations.

Proposition 3.10 (Equivalence of Recurrence and Convolution)**.** *The calculation of Haar approximation and detail coefficients via recurrence relations (3.1)–(3.2) is mathematically equivalent to performing a strided convolution with the conjugate-flipped Haar filters $\tilde{h} = [1/\sqrt{2}, 1/\sqrt{2}]$ and $\tilde{g} = [1/\sqrt{2}, -1/\sqrt{2}]$. Specifically:*

$$a_j = (a_{j+1} * \tilde{h})_{\downarrow 2} \quad \text{and} \quad d_j = (a_{j+1} * \tilde{g})_{\downarrow 2}, \tag{3.6}$$

where $(\cdot)_{\downarrow 2}$ denotes downsampling by a factor of 2 (stride $S = 2$). ◊

3.1.3 A FHT Example

Example 3.11. [*A discrete Haar decomposition*] Let us illustrate the Fast Haar Transform with a simple numerical example. Consider a function f discretized at the finest scale $J = 0$ with $N = 4$ samples: $a_0 = [4, 6, 10, 2]$.

- **Step 1: From scale $J = 0 \to j = -1$.** Following Proposition 3.10, we can compute the next scale using either the scalar recurrence or the filter convolution.

  Method A: Scalar Recurrence (Intuitive) We apply the average and difference formulas to disjoint pairs:

  - Pair $(4, 6) \to a_{-1,0} = \frac{4+6}{\sqrt{2}} = 5\sqrt{2}, \quad d_{-1,0} = \frac{-4+6}{\sqrt{2}} = \sqrt{2}.$
  - Pair $(10, 2) \to a_{-1,1} = \frac{10+2}{\sqrt{2}} = 6\sqrt{2}, \quad d_{-1,1} = \frac{-10+2}{\sqrt{2}} = -4\sqrt{2}.$

  Method B: Convolutional Verification (Algorithmic) Let us verify $d_{-1,0}$ using the convolution definition with the high-pass filter $\tilde{g} = [1/\sqrt{2}, -1/\sqrt{2}]$. The convolution $(a_0 * \tilde{g})$ at index 0 involves the dot product of the signal segment $[4, 6]$ with the filter g:

  $$d_{-1,0} = \left(4 \cdot \frac{-1}{\sqrt{2}}\right) + \left(6 \cdot \frac{1}{\sqrt{2}}\right) = \frac{2}{\sqrt{2}} = \sqrt{2}.$$

  Both methods yield the representation at scale -1:

  $$a_{-1} = [5\sqrt{2}, 6\sqrt{2}], \quad d_{-1} = [\sqrt{2}, -4\sqrt{2}].$$

  Connection to nested spaces: Notice that while a_{-1} has dimension 2, as a function in $V_{-1} \subset V_0$, it represents a step function with double width. If we were to view this result back in the dimension 4 of V_0, the vector would be $[5\sqrt{2}, 5\sqrt{2}, 6\sqrt{2}, 6\sqrt{2}]$. This illustrates the consistency remark above: the function exists in the finer space as repeated values, but we efficiently store only the unique values.

- **Step 2: From scale $j = -1 \to j = -2$.** Now we iterate the process on the approximation vector a_{-1} (the details d_{-1} are stored and left untouched).

  For the pair $(5\sqrt{2}, 6\sqrt{2})$:

  - Average ($a_{-2,0}$): $\frac{5\sqrt{2}+6\sqrt{2}}{\sqrt{2}} = \frac{11\sqrt{2}}{\sqrt{2}} = 11$.
  - Difference ($d_{-2,0}$): $\frac{-5\sqrt{2}+6\sqrt{2}}{\sqrt{2}} = \frac{\sqrt{2}}{\sqrt{2}} = 1$.

Final Result: The Full Haar Wavelet Transform of the original vector is the concatenation of the final approximation and all detail coefficients ordered by scale, representing the decomposition $V_{-2} \oplus W_{-2} \oplus W_{-1}$:

$$\text{Haar}(f) = [a_{-2,0}, d_{-2,0}, d_{-1,0}, d_{-1,1}] = [11, 1, \sqrt{2}, -4\sqrt{2}].$$

Notice how the energy (squared Euclidean norm) is preserved throughout the process, confirming the orthonormal nature of the transform:

- Original energy: $4^2 + 6^2 + 10^2 + 2^2 = 16 + 36 + 100 + 4 = 156$.
- Transformed energy: $11^2 + 1^2 + (\sqrt{2})^2 + (-4\sqrt{2})^2 = 121 + 1 + 2 + 32 = 156$.

$\Diamond$

3.2 The Fast Two-Dimensional Haar Transform

Following the discretization of 1D functions, we can similarly represent functions in $L^2(\mathbb{R}^2)$ as matrices in $\mathbb{C}^{n\times p}$, commonly denoted as images. As established in the MRA construction, the 2D Haar basis is *separable*. This property is computationally fundamental, as it allows us to implement the two-dimensional transform by simply applying the 1D-FHT sequentially along the rows and then the columns of the image.

It suffices to note that this sequential row-column filtering is mathematically equivalent to performing a *convolution with stride $S = 2$ using four distinct 2×2 kernels*. In this section, we derive these four 2D filters directly from the Haar basis definitions.

Remark 3.12. It is crucial to recall Proposition 3.10. While we define the filters here in their natural "basis orientation" (matching the visual features they detect), the actual algorithmic implementation via *convolution* will require the *conjugate flip* (180-degree rotation) of these matrices. $\Diamond$

3.2.1 2D Filter Banks

Recall Rem. 2.21, the 1D Haar analysis filters (normalized) are:

- Low-pass (scaling) filter h: corresponds to the coefficients $[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$.
- High-pass (wavelet) filter g: corresponds to the coefficients $[\frac{-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]$.

Let us define these four 2D filters. Note that the normalization factor becomes $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}} = \frac{1}{2}$.

1. **Approximation Filter (LL):** Generated by applying h to both rows and columns:
$$H_{LL} = h \otimes h = \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}1 & 1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & 1\\1 & 1\end{pmatrix}.$$
This kernel computes the local average of the 2×2 block. Recalling Remark 3.4, convolving with this filter is mathematically equivalent to applying an *Average Pooling* layer.

2. **Vertical Detail Filter (LH):** Generated by applying h to the vertical axis (averaging rows) and g to the horizontal axis (differencing columns):
$$H_{LH} = h \otimes g = \frac{1}{2}\begin{pmatrix}1\\1\end{pmatrix}\begin{pmatrix}-1 & 1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}-1 & 1\\-1 & 1\end{pmatrix}.$$
This kernel detects sudden changes between the image columns (i.e., vertical edges).

3. **Horizontal Detail Filter (HL):** Generated by applying g to the vertical axis (differencing rows) and h to the horizontal axis (averaging columns):
$$H_{HL} = g \otimes h = \frac{1}{2}\begin{pmatrix}-1\\1\end{pmatrix}\begin{pmatrix}1 & 1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}-1 & -1\\1 & 1\end{pmatrix}.$$
This kernel detects sudden changes between the image rows (i.e., horizontal edges).

4. **Diagonal Detail Filter (HH):** Generated by applying g to both axes:
$$H_{HH} = g \otimes g = \frac{1}{2}\begin{pmatrix}-1\\1\end{pmatrix}\begin{pmatrix}-1 & 1\end{pmatrix} = \frac{1}{2}\begin{pmatrix}1 & -1\\-1 & 1\end{pmatrix}.$$
This kernel detects sudden changes between the image diagonals (i.e., diagonal edges).

Remark 3.13. There is a direct equivalence between these filters and the basis functions visualized in Fig. 2.1. The filter coefficients are exactly the values of the functions $\phi, \psi^v, \psi^h, \psi^d$ on the unit square, scaled by a factor of $1/2$. The scaling is due to the energy preservation, as in the 1D-FHT. ◊

Remark 3.14. The main disadvantage of this separable approach is that the resulting analysis remains highly axis-dependent. ◊

3.2.2 The Mallat Decomposition

Similarly to the one-dimensional case, we can decompose the image into the coefficient matrix. To visualize the approximation and detail images in this separable wavelet decomposition, we proceed as follows. Suppose our fine-resolution discrete image lives in $\mathcal{V}_2$; we denote this image as the approximation a_2. The algorithm decomposes this into a coarser approximation $a_1 \in \mathcal{V}_1$, which contains $1/4$ of the original data, and the detail component $d_1 \in \mathcal{W}_1$.

The detail component is the sum of the horizontal, diagonal, and vertical details:

$$d_1 = d_1^h + d_1^d + d_1^v, \tag{3.7}$$

where each part carries $1/4$ of the initial data. Thus, we have the relation $a_2 = a_1 + d_1$.

This process is iterative. We can repeat it for a_1, decomposing it into a coarser approximation $a_0 \in \mathcal{V}_0$ and details $d_0 = d_0^h + d_0^d + d_0^v \in \mathcal{W}_0$. This yields the multi-level representation:

$$a_1 = a_0 + d_0 \implies a_2 = a_0 + d_0 + d_1.$$

This process can be visualized using the Mallat decomposition algorithm shown in Fig. 3.1. At each level, the approximation image (a_i) is further decomposed into a coarser approximation (a_{i-1}) and three detail components $(d_{i-1}^h, d_{i-1}^d, d_{i-1}^v)$, creating a pyramid of features.

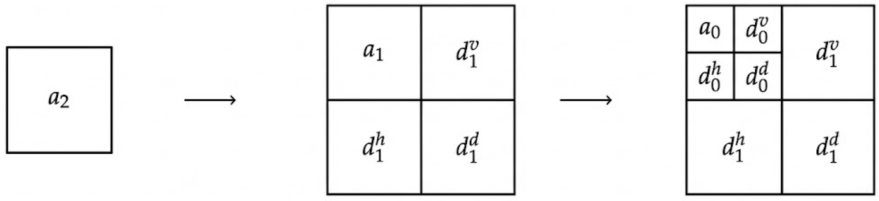


Figure 3.1: Visualization of the Mallat decomposition algorithm. The figure illustrates the 2-scale process, highlighting the specific spatial arrangement of the coefficients at each decomposition step.

Chapter 4

Analysis on the Unit Disk: Zernike Wavelets

While the Haar transform is powerful for detecting edges aligned with cartesian axes, many natural objects (like food on a plate) have curved edges. To analyze such signals effectively, we turn to bases defined on the unit disk $B(0,1) := \{(x,y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$. To be able to construct a kind of wavelet on the unit disk, we focus on the Paper [15] about Zernike Wavelets.

4.1 Zernike Polynomials

4.1.1 Definition and Orthonormal Basis for $L^2(B(0,1))$

Unlike the Haar basis, which is localized in space, Zernike polynomials are global but *localized in frequency*. The Zernike polynomials form a complete orthonormal set in the space of square-integrable functions on the unit disk, i.e., $L^2(B(0,1))$. Unlike the complex definition often used in optics, for better implementation, here we define the real Zernike polynomials directly in terms of trigonometric functions.

Definition 4.1 (*Zernike Polynomials*)**.** Let $\mathbb{N}_0 = \{0\} \cup \mathbb{N}$. The real Zernike polynomial with radial degree $n \in \mathbb{N}_0$ and azimuthal degree $m \in \mathbb{Z}$, where $|m| \leq n$, and $n - |m|$ is even, is defined as:

$$Z_n^m(r,\phi) = \begin{cases} \gamma_n^m R_n^{|m|}(r)\cos(m\phi), & \text{if } m \geq 0; \\ \gamma_n^m R_n^{|m|}(r)\sin(|m|\phi), & \text{if } m < 0. \end{cases} \tag{4.1}$$

The radial parts $R_n^{|m|}(r)$ are denoted as *radial polynomials* and are given by the explicit polynomial formula:

$$R_n^{|m|}(r) = \sum_{s=0}^{(n-|m|)/2} \frac{(-1)^s (n-s)!}{s!\left(\frac{n+|m|}{2}-s\right)!\left(\frac{n-|m|}{2}-s\right)!} r^{n-2s}. \tag{4.2}$$

The normalization constants γ_n^m are chosen to ensure orthonormality over the unit disk:

$$\gamma_n^m = \begin{cases} \sqrt{\frac{n+1}{\pi}}, & \text{if } m = 0; \\ \sqrt{\frac{2(n+1)}{\pi}}, & \text{if } m \neq 0. \end{cases} \tag{4.3}$$

$\Diamond$

One can immediately see that $Z_n^m(r,\phi)$ is a polynomial of degree n in the cartesian variables (x,y).

The explicit definition of the radial polynomials in Equation (4.2) involves factorials of terms involving n. For high radial degrees, the direct computation of these factorials leads to numerical instability and overflow errors. To address this, various recurrence relations exist, but the one we will employ is the relation derived by Kintner as discussed in [36], which allows for the efficient iterative computation of $R_n^m(r)$ recursively on the order n. Note that here m is considered always non-negative.

Proposition 4.2 (*Kintner's Recurrence Relation*)**.** *The radial polynomials $R_n^m(r)$ satisfy the following three-term recurrence relation. For a fixed azimuthal frequency m, the polynomial of order n can be computed from the polynomials of order $n-2$ and $n-4$:*

$$R_n^m(r) = \frac{(K_2 r^2 + K_3) R_{n-2}^m(r) + K_4 R_{n-4}^m(r)}{K_1}, \tag{4.4}$$

where the coefficients K_1, K_2, K_3, K_4 depend solely on n and are given by:

$$K_1 = \frac{n(n^2-m^2)(n-2)}{2}, \quad K_2 = 2n(n-1)(n-2),$$
$$K_3 = -m^2(n-1) - n(n-1)(n-2), \quad K_4 = -\frac{n(n+m-2)(n-m-2)}{2}.$$

This relation holds for $n \geq m+4$. The base cases for the recursion ($n = m$ and $n = m+2$) are computed explicitly using the definition, i.e., r^n and $nr^n - (n-1)r^{n-2}$ respectively. ◊

This recursive method will prove essential for the future Zernike MRA construction.

If needed, the Zernike polynomials can be single indexed.

Definition 4.3 (*ANSI indexing*)**.** The ANSI indexed Zernike polynomials are given by the conversion Z_n^m to Z_j with $j \in \mathbb{N}_0$ as follows:

$$j = \frac{n(n+2)+m}{2}.$$

◊

Their orthonormality is given by the following proposition.

Proposition 4.4 (*Zernike orthonomal basis*)**.** *The Zernike polynomials form an orthonormal family in the $L^2(B(0,1))$ space, i.e., for each $n, n', m, m' \in \mathbb{Z}$ they satisfy the following equality:*

$$\langle Z_n^m, Z_{n'}^{m'} \rangle = \int_0^1 \int_0^{2\pi} Z_n^m(r,\phi) Z_{n'}^{m'}(r,\phi)\, r\, d\phi\, dr = \delta_{n,n'}\delta_{m,m'}, \tag{4.5}$$

where δ is the Kronecker delta.[1]

Proof. Recall the definition of the real Zernike polynomials $Z_n^m(r,\phi) = \gamma_n^m R_n^{|m|}(r)$ $\Theta_m(\phi)$, where $\Theta_m(\phi)$ is $\cos(m\phi)$ for $m \geq 0$ and $\sin(|m|\phi)$ for $m < 0$. The inner product can be separated into radial and angular components as follows:

$$\langle Z_n^m, Z_{n'}^{m'} \rangle = \gamma_n^m \gamma_{n'}^{m'} \left(\int_0^1 R_n^{|m|}(r) R_{n'}^{|m'|}(r)\, r\, dr \right) \left(\int_0^{2\pi} \Theta_m(\phi)\Theta_{m'}(\phi)\, d\phi \right).$$

[1] The appearence of the r term in the integral comes from the coordinate change to polar form.

1. **Angular Orthogonality:** The angular functions are trigonometric. Their orthogonality properties on $[0, 2\pi]$ imply:

   - If $|m| \neq |m'|$, the frequencies differ, so the integral is 0.
   - If $|m| = |m'|$ but $m \neq m'$ (i.e., $m = -m' \neq 0$), one is a sine and the other a cosine. The integral is also 0.

   Thus, the inner product is zero unless $m = m'$. If $m = m'$, the angular integral values are:

$$\int_0^{2\pi} \Theta_m(\phi)^2 \, d\phi = \begin{cases} \int_0^{2\pi} 1^2 \, d\phi = 2\pi, & \text{if } m = 0; \\ \int_0^{2\pi} \cos^2(m\phi) \, d\phi = \pi, & \text{if } m > 0; \\ \int_0^{2\pi} \sin^2(|m|\phi) \, d\phi = \pi, & \text{if } m < 0. \end{cases}$$

   This can be summarized as $\pi(1 + \delta_{m,0})$.

2. **Radial Orthogonality:** Assuming $m = m'$, we look at the radial part. As established by Nijboer [35] (Eq. 2.18):

$$\int_0^1 R_n^{|m|}(r) R_{n'}^{|m|}(r) \, r \, dr = \frac{1}{2n+2} \delta_{n,n'}. \tag{4.6}$$

3. **Normality:** Combining both parts for the case $n = n'$ and $m = m'$, we substitute the normalization constant γ_n^m defined in Eq. (4.3):

$$\|Z_n^m\|^2 = (\gamma_n^m)^2 \cdot \left(\frac{1}{2n+2} \right) \cdot \pi(1 + \delta_{m,0}) = \frac{2n+2}{\pi(1+\delta_{m,0})} \cdot \frac{\pi(1+\delta_{m,0})}{2n+2} = 1.$$

Thus, $\langle Z_n^m, Z_{n'}^{m'} \rangle = \delta_{n,n'} \delta_{m,m'}$. □

Theorem 4.5 (*Zernike Basis*)**.** *The family of Zernike polynomials* $\{Z_n^m\}_{n \geq 0, |m| \leq n}$ *forms a complete orthonormal basis for* $L^2(B(0,1))$.

Proof. By the *Weierstrass Approximation Theorem* generalized to compact subsets of $\mathbb{R}^n$, the set of polynomials in two variables $\mathcal{P} = \text{span}\{x^\alpha y^\beta\}$ restricted to the closed unit disk $B(0,1)$ is dense in $C(B(0,1))$ with respect to the uniform norm $\|\cdot\|_\infty$ (recall that for $P \in \mathcal{P}$, $\|P\|_\infty = \sup\{|P(s)| : s \in B(0,1)\}$).

Since the domain $B(0,1)$ is bounded (and has finite measure π), uniform convergence implies L^2 convergence (i.e., $\|f - P\|_2 \leq \sqrt{\pi} \|f - P\|_\infty$). Consequently, the space of polynomials is also dense in $C(B(0,1))$ with respect to the L^2-norm.

Since $C(B(0,1))$ is dense in $L^2(B(0,1))$ (Thm. A.24), it follows by transitivity that the space of polynomials is dense in $L^2(B(0,1))$.

Finally, since the Zernike polynomials are an infinite orthonormal family that have all the possible degrees, their linear span is identical to the span of all polynomials $\mathcal{P}$. Thus, their closure in the L^2 sense is the whole space $L^2(B(0,1))$, satisfying the completeness condition. □

As stated in Thm. A.20, $L^2(B(0,1))$ is a *Hilbert space*. Consequently, since the Zernike polynomials form a complete orthonormal basis, any function $f \in L^2(B(0,1))$ admits a unique expansion as a Fourier-Zernike series:

$$f(r,\phi) = \sum_{n=0}^{\infty} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} c_{nm} Z_n^m(r,\phi), \tag{4.7}$$

where the coefficients c_{nm} are determined by the orthogonal projection $c_{nm} = \langle f, Z_n^m \rangle$ and will be called as *Zernike Moments*.

4.1.2 The Scaling Spaces V_N

We can create subspaces of $L^2(B(0,1))$ with different families of Zernike polynomials as their orthonormal bases.

Given the multi-index $\alpha = (\alpha_1, \alpha_2) \in \mathbb{N}_0^2$, we define the degree as $|\alpha| := \alpha_1 + \alpha_2$, and for $x = (x_1, x_2) \in \mathbb{R}^2$, we denote the monomial as $x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2}$.

Definition 4.6. Given a $N \in \mathbb{N}_0$, we define V_N as the space of all polynomials in two variables of degree at most N restricted to the unit disk $B(0,1)$, i.e.,

$$V_N := \text{span}\{x^\alpha : \alpha \in \mathbb{N}_0^2, |\alpha| \leq N\}.$$

$\Diamond$

We can determine the dimension of each space V_N simply by counting the number of linearly independent monomials.

Proposition 4.7. *The dimension of the space V_N is given by:*

$$\dim(V_N) = \frac{(N+1)(N+2)}{2}.$$

Proof. The dimension is given by the number of multi-indices satisfying $\alpha_1 + \alpha_2 \leq N$. If we fix the degree of the first variable to be $\alpha_1 = d$ (where $0 \leq d \leq N$), the second variable α_2 can take any integer value from 0 to $N - d$.

This gives exactly $(N - d) + 1$ possibilities for each d. Consequently:

$$\begin{aligned}\dim(V_N) &= \sum_{d=0}^{N} (N - d + 1) = (N+1)(N+1) - \sum_{d=0}^{N} d = (N+1)^2 - \frac{N(N+1)}{2} \\ &= \frac{2(N^2 + 2N + 1) - (N^2 + N)}{2} = \frac{N^2 + 3N + 2}{2} = \frac{(N+1)(N+2)}{2}.\end{aligned}$$

□

These spaces are closed subspaces of $L^2(B(0,1))$ as stated in the next proposition.

Proposition 4.8. *For every $N \in \mathbb{N}_0$, the space V_N is a closed subspace of $L^2(B(0,1))$.*

Proof. Since V_N is a subspace of a Hilbert space with finite dimension (as shown above), by Thm. A.8 it is necessarily closed. □

Converting to polar coordinates, one can immediately see that the polynomials that live in V_N are the ones that have r^α with $\alpha \leq N$. Thus, recalling the definition of the Zernike Polynomials, $Z_n^m \in V_N$ for every $n \leq N$. Following this inclusion, we can state a powerful proposition.

Theorem 4.9. *The family of Zernike polynomials* $\mathcal{Z}_N := \{Z_n^m : 0 \leq n \leq N, |m| \leq n, n - |m| \text{ even}\}$ *constitutes an orthonormal basis for* V_N.

Proof. The set $\mathcal{Z}_N$ consists of orthonormal polynomials contained in V_N (since their degree is at most N). Being a linearly independent set within a finite-dimensional space, it suffices to verify that its cardinality $|\mathcal{Z}_N|$ matches $\dim(V_N)$.

The cardinality is given by the sum over the possible indices:

$$|\mathcal{Z}_N| = \sum_{n=0}^{N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} 1.$$

Let us evaluate the inner sum, which counts the number of valid m indices for a fixed n. We distinguish two cases based on the parity of n:

1. **If n is odd:** The index m must be odd. The possible values are $\{\pm 1, \pm 3, \ldots, \pm n\}$. The number of positive odd integers up to n is $(n+1)/2$. Including the negative counterparts, the total count is $2 \cdot \frac{n+1}{2} = n + 1$.
2. **If n is even:** The index m must be even. The possible values are $\{0, \pm 2, \ldots, \pm n\}$. The number of positive even integers up to n is $n/2$. Including the negative counterparts and zero, the total count is $2 \cdot \frac{n}{2} + 1 = n + 1$.

In both cases, the number of valid m indices is exactly $n + 1$. Consequently, the total cardinality is the sum of an arithmetic progression:

$$|\mathcal{Z}_N| = \sum_{n=0}^{N} (n+1) = 1 + 2 + \cdots + (N+1) = \frac{(N+1)(N+2)}{2}.$$

Since this value coincides with $\dim(V_N)$, the set $\mathcal{Z}_N$ forms a basis for V_N. □

4.2 The Zernike MRA

The definition of a Multiresolution Analysis on the unit disk $B(0,1)$ doesn't come immediately. First we need to define a set of special polynomials generated from the Zernike polynomials that will become key to the MRA definition.

4.2.1 The Kernel Polynomials

Definition 4.10 (*The Kernel Polynomials*)**.** For a given pair (ρ, θ) with $0 \leq \rho \leq 1$ and $0 \leq \theta \leq 2\pi$, the *N-th kernel polynomial* with respect to the L^2-inner product is given by the following expression:

$$K_N(r, \phi; \rho, \theta) := \sum_{n=0}^{N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} Z_n^m(\rho, \theta) Z_n^m(r, \phi). \tag{4.8}$$

◊

We can then extract an important property of the kernel polynomials.

Proposition 4.11. *Given $N \in \mathbb{N}_0$, a pair (ρ, θ) and a polynomial $p \in V_N$, then the following reproducing property holds:*

$$\langle p, K_N(\cdot, \cdot; \rho, \theta)\rangle = \sum_{n=0}^{N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} \langle p, Z_n^m\rangle Z_n^m(\rho, \theta) = p(\rho, \theta). \tag{4.9}$$

Proof. First, using the definition of the L^2-inner product and the linearity of the integral, we expand the expression:

$$\begin{aligned}
\langle p, K_N(\cdot, \cdot; \rho, \theta)\rangle &= \int_0^1 \int_0^{2\pi} p(r, \phi) \left(\sum_{n=0}^{N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} Z_n^m(r, \phi) Z_n^m(\rho, \theta) \right) r \, d\phi \, dr \\
&= \sum_{n=0}^{N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} \left(\int_0^1 \int_0^{2\pi} p(r, \phi) Z_n^m(r, \phi) \, r \, d\phi \, dr \right) Z_n^m(\rho, \theta) \\
&= \sum_{n=0}^{N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} \langle p, Z_n^m\rangle Z_n^m(\rho, \theta).
\end{aligned}$$

Finally, since the Zernike family $\mathcal{Z}_N$ constitutes an orthonormal basis for the space V_N, the expression obtained is precisely the expansion of p in this basis. Since $p \in V_N$, this expansion is exact and equals $p(\rho, \theta)$. □

The Kernel polynomials are the solutions to a specific optimization problem.

Lemma 4.12. *For a given pair $(\rho, \theta) \in B(0, 1)$ and a $N \in \mathbb{N}_0$, the normalized Kernel Polynomial $\left\| \frac{K_N(r,\phi,\rho,\theta)}{K_N(\rho,\theta,\rho,\theta)} \right\|$ is the solution of the following optimization problem:*

$$\min \{\|p\| : p \in V_N, p(\rho, \theta) = 1\}.$$

Proof. Let $p(r, \phi) \in V_N$ with $p(\rho, \theta) = 1$. Then by the given expansion using the Zernike basis

$$p(r, \phi) = \sum_{n=0}^{N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} c_{nm} Z_n^m(r, \phi).$$

Consequently, by the orthonormality of the basis

$$\|p\|^2 = \langle p, p\rangle = \sum_{n=0}^{N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} c_{nm}^2.$$

Now, we apply the $\mathbb{R}$-sums *Cauchy-Schwarz inequality* $\left(\left(\sum a_i b_i\right)^2 \le \sum a_i^2 \sum b_i^2 \right)$:

$$1 = p(\rho, \theta)^2 \le \|p\|^2 \left(\sum_{n,m} Z_n^m(\rho, \theta)^2 \right). \tag{4.10}$$

By the orthonomality of the Zernike basis and the bilinearity of the inner product

$$\begin{aligned}\|K_N(\cdot,\cdot;\rho,\theta)\|^2 &= \langle K_N(\cdot,\cdot;\rho,\theta), K_N(\cdot,\cdot;\rho,\theta)\rangle \\ &= \left\langle \sum_{n,m} Z_n^m(\cdot,\cdot) Z_n^m(\rho,\theta), \sum_{n,m} Z_n^m(\cdot,\cdot) Z_n^m(\rho,\theta) \right\rangle \\ &= \sum_{n,m} Z_n^m(\rho,\theta)^2.\end{aligned} \tag{4.11}$$

We also have by definition

$$K_N(\rho,\theta;\rho,\theta) = \sum_{n,m} Z_n^m(\rho,\theta)^2. \tag{4.12}$$

Thus, combining (4.10), (4.11) and (4.12)

$$\left\| \frac{K_N(\cdot,\cdot,\rho,\theta)}{K_N(\rho,\theta,\rho,\theta)} \right\|^2 = \frac{\|K_N(\cdot,\cdot,\rho,\theta)\|^2}{\left(\sum_{n,m} Z_n^m(\rho,\theta)^2\right)^2} = \frac{1}{\sum_{n,m} Z_n^m(\rho,\theta)^2} \leq \|p\|^2 .$$

□

4.2.2 Zernike Scaling Functions

Given the closed nested subspaces $V_0 \subset V_1 \subset \cdots \subset V_N \subset L^2(B(0,1))$, we aim to define a Multiresolution Analysis (MRA) structure analogous to the classical case on $\mathbb{R}$.

In classical MRAs (like Haar), the space V_j is spanned by translations and dilations of a single scaling function ϕ. However, on the bounded domain of the unit disk, standard translation is not well-defined (as it would move functions out of the domain).

To overcome this, Datta et al. [15] replace the concept of *translation* with *localization* around specific nodes. Instead of shifting a function, we construct a family of functions $\{\phi_{N,j}\}$, denoted as *Zernike Scaling Functions*, that are centered at a discrete set of points $\{P_j\}$ on the disk. These functions form a basis for V_N and inherit the reproducing properties of the Kernel polynomials.

To achieve that, first we need to define the optimal unit disk sampling for the correct localization of the scaling functions, ensuring their numerical stability. We will follow the sampling strategy stated by [39], stated as follows.

Definition 4.13 (*Regular Points: Optimal Concentric Sampling*)**.** For a given radial degree N, we establish $k = \lfloor \frac{N}{2} \rfloor + 1$ concentric circles.

The radii of these circles, denoted as $\{r_j\}_{j=1}^k$, are determined by the zeros of Chebyshev polynomials. Specifically, the approximate optimal radii are given by the polynomial fitting:

$$r_j = 1.1565\,\zeta_j - 0.76535\,\zeta_j^2 + 0.60517\,\zeta_j^3, \quad \text{for } j = 1,\ldots,k, \tag{4.13}$$

where ζ_j correspond to the zeros of the Chebyshev polynomial of the first kind, defined as:

$$\zeta_j = \cos\left(\frac{(2j-1)\pi}{2(N+1)}\right). \tag{4.14}$$

On each circle of radius r_j, we place $n_j = 2N + 5 - 4j$ equally spaced angular nodes. Thus, the final points P_j denoted as regular points are defined in polar coordinates as

$$\left(r_j, \frac{2\pi(s_j-1)}{n_j}\right).$$

The total number of points in this grid matches exactly the dimension of the space V_N, i.e., $\sum n_j = \dim(V_N) = (N+1)(N+2)/2$. ◊

Definition 4.14. Given the Zernike polynomials enumerated using the ANSI indexing 4.3 $\{Z_j\}_{j=0}^{J-1}$ and the set of regular points $\{P_j\}_{j=1}^{J}$, we can construct the following collocation matrix:

$$A_N = [Z_{j-1}(P_i^{(N)})]_{i,j=1}^{J}, \tag{4.15}$$

where its rows correspond to the different evaluation points P_i, while the columns correspond to the different polynomials Z_j. ◊

The distribution of interpolation nodes on the unit disk is critical for the accuracy of the reconstruction. Standard grids (such as uniform polar or Cartesian grids) often result in ill-conditioned collocation matrices, leading to numerical instabilities where the condition number κ_2 grows exponentially with the polynomial order N.

The OCS pattern defined above is constructed specifically to minimize the condition number of the Zernike collocation matrix. It has been shown that this configuration outperforms other patterns, such as spiral sampling, particularly for higher radial orders ($N > 15$), maintaining a low condition number ($\kappa_2 < 100$ even for $N = 30$) [39]. This ensures that the basis of scaling functions constructed upon these points forms a stable and well-conditioned system for the Multiresolution Analysis.

Thus we can now define the scaling functions.

Definition 4.15. Given $N \in \mathbb{N}_0$, let $J = \dim V_N$. Given the set of regular points $\{P_j^{(N)} = (\rho_j^{(N)}, \theta_j^{(N)})\}_{j=1}^{J}$, the scaling functions are defined as

$$\phi_{N,j}(r,\phi) := K_N\left(r, \phi; \rho_j^{(N)}, \theta_j^{(N)}\right) \quad \text{for } j = 1, \dots J.$$

◊

The basis property of these scaling functions is the result of the following theorem.

Theorem 4.16. *Let $N \in \mathbb{N}_0$. Then, the set of regular points $\{P_j^{(N)} = (\rho_j^{(N)}, \theta_j^{(N)})\}_{j=1}^{J}$, the set*

$$\{\phi_{N,j}(r,\phi)\}_{j=1}^{J}$$

is a basis of V_N. ◊

Theorem 4.17. *For a given N, let $\{P_j^{(N)} = (\rho_j^{(N)}, \theta_j^{(N)})\}_{j=1}^{J}$ be a set of regular points as described in Def. 4.13. The set*

$$\left\{K_N(r, \phi; \rho_j^{(N)}, \theta_j^{(N)})\right\}_{j=1}^{J}$$

is a basis of V_N. Consequently, the set of scaling functions $\{\phi_{N,j}(r,\phi)\}_{j=1}^{J}$ is a basis of V_N.

Proof. Let $A_N^{(i)}$ be the matrix obtained from A_N 4.15 by replacing the i-th row with the functions $\{Z_{j-1}(r,\phi)\}_{j=1}^{J}$. Under the choice of points $\{P_j^{(N)}\}_{j=1}^{J}$, the Lagrange interpolating polynomial is uniquely determined, since we have exactly as many points

as the dimension of the space. The i-th fundamental Lagrange polynomial of degree N is given by:

$$l_i(r,\phi) = \frac{1}{\det(A_N)} \det(A_N^{(i)}(r,\phi)).$$

These polynomials are characterized by the property $l_i(P_j^{(N)}) = \delta_{ij}$. This is evident from the determinant definition: if we evaluate l_i at a point P_j with $j \neq i$, the matrix $A_N^{(i)}$ will have two identical rows, resulting in a zero determinant. If evaluated at P_i, the matrix matches A_N, yielding 1.

Note that the size of the set $\{K_N(r,\phi;P_j^{(N)})\}_{j=1}^J$ is the same as the dimension of V_N. Therefore, to show that this set is a basis of V_N, it suffices to show that it is linearly independent. For convenience, let us denote $K_N(r,\phi;P_j^{(N)})$ as K_j. Suppose we have a linear combination summing to zero:

$$\sum_{j=1}^{J} \tau_j K_j = 0. \tag{4.16}$$

For a fixed i, we take the inner product on both sides of Equation (4.16) with the i-th fundamental Lagrange polynomial l_i. By the linearity of the inner product, the summation and scalars τ_j move outside:

$$0 = \left\langle l_i, \sum_{j=1}^{J} \tau_j K_j \right\rangle = \sum_{j=1}^{J} \tau_j \underbrace{\langle l_i, K_j \rangle}.$$

Here we apply the reproducing property of the kernel polynomial (Equation 4.9). Since $l_i \in V_N$, it can be expanded in the Zernike basis as $l_i = \sum c_k Z_k$. Thus, the inner product with the kernel evaluated at P_j acts as an evaluation functional:

$$\langle l_i, K_j \rangle = l_i(P_j^{(N)}).$$

Substituting this back into our equation:

$$0 = \sum_{j=1}^{J} \tau_j l_i(P_j^{(N)}) = \sum_{j=1}^{J} \tau_j \delta_{ij} = \tau_i.$$

Since this holds for each $i = 1, \ldots, J$, we conclude that all coefficients τ_i must be zero, proving that the set is linearly independent. □

A summary of the scaling functions properties are stated in the next theorem.

Theorem 4.18. *1.* ***Inner product:*** *The inner product of the scaling functions gives the following equality:*

$$\langle \phi_{N,i}, \phi_{N,j} \rangle = \phi_{N,i}(P_j). \tag{4.17}$$

2. ***Localization:*** *The scaling function $\phi_{N,j}$ is localized around P_j, i.e., it is the solution to the optimization problem:*

$$\left\| \frac{\phi_{N,j}(r,\phi)}{\phi_{N,j}(P_j)} \right\| = \min \{ \|p\| : p \in V_N, p(P_j) = 1 \}. \tag{4.18}$$

3. ***Basis:*** *The set of scaling functions $\{\phi_{N,j}\}_{j=1}^{J}$ is a basis of V_N.*

4. ***Dual:*** *The dual of the scaling functions $\{\phi_{N,j}\}_{j=1}^{J}$ is given by the fundamental Lagrange interpolating polynomials $\{l_k\}_{k=1}^{J}$ with respect to the regular points $\{P_j\}$.*

5. ***Orthogonality:*** *The scaling function $\phi_{N,j}$ is orthogonal to V_{N-1} with respect to the modified inner product $\langle \cdot, \cdot(\cdot - P_j)\rangle$, i.e., for all $q \in V_{N-1}$:*

$$\langle \phi_{N,j}(\cdot), q(\cdot)(\cdot - P_j)\rangle = 0. \tag{4.19}$$

Proof. 1. Recall that $\phi_{N,i} = K_N(\cdot, \cdot; P_i)$. Using the definition of the inner product and the reproducing property of the kernel polynomial (Eq. 4.9):

$$\langle \phi_{N,i}, \phi_{N,j}\rangle = \langle K_N(\cdot; P_i), K_N(\cdot; P_j)\rangle.$$

Since $\phi_{N,i} \in V_N$, applying the reproducing property for the kernel centred at P_j yields:

$$\langle \phi_{N,i}, K_N(\cdot; P_j)\rangle = \phi_{N,i}(P_j).$$

Note that since we are working with real Zernike polynomials, the kernel is symmetric $K_N(P_j, P_i) = K_N(P_i, P_j)$, so the order of evaluation does not affect the result.

2. This follows immediately from the Lem. 4.12.

3. This is a direct consequence of the previous Thm. 4.17.

4. Let $\{l_k\}_{k=1}^{J}$ be the fundamental Lagrange interpolating polynomials characterized by $l_k(P_j) = \delta_{kj}$. To verify they form the dual basis, we check the biorthogonality condition with the scaling functions:

$$\langle \phi_{N,j}, l_k\rangle = \langle K_N(\cdot; P_j), l_k\rangle.$$

Since $l_k \in V_N$, we can apply the reproducing property of the kernel K_N:

$$\langle l_k, K_N(\cdot; P_j)\rangle = l_k(P_j).$$

By the definition of the Lagrange polynomials, $l_k(P_j) = \delta_{kj}$. Thus, $\{l_k\}$ is indeed the dual basis.

5. Let $q \in V_{N-1}$. We consider the term $f(x) = q(x)(x - P_j)$. Since q has degree at most $N-1$, multiplying by the linear term $(x - P_j)$ increases the degree by exactly 1. Therefore, the resulting function f has degree at most N, implying $f \in V_N$.

Since $f \in V_N$, we can validly apply the reproducing property of the kernel $\phi_{N,j} = K_N(\cdot; P_j)$:

$$\langle \phi_{N,j}, f\rangle = \langle K_N(\cdot; P_j), f\rangle = f(P_j).$$

Evaluating f at the point P_j:

$$f(P_j) = q(P_j)(P_j - P_j) = q(P_j) \cdot 0 = 0.$$

Thus, the orthogonality holds.

□

Theorem 4.19. *Consider a set of regular points $\{P_j\}_{j=1}^J$ as described in Def. 4.13. The following conditions are equivalent regarding the scaling functions $\{\phi_{N,j}\}_{j=1}^J$:*

1. *The scaling functions form an orthogonal set, i.e.,*

$$\langle \phi_{N,k}, \phi_{N,\ell} \rangle = 0 \quad \text{for } k \neq \ell. \tag{4.20}$$

2. *The scaling functions satisfy*

$$\phi_{N,k}(P_\ell) = d_k^{(N)} \delta_{k\ell} \quad \text{for } k, \ell = 1, \ldots, J, \tag{4.21}$$

where $d_k^{(N)} \in \mathbb{R}$.

Proof. We first show that (1) implies (2). From Thm. 4.18 (first property), it is known that:

$$\langle \phi_{N,k}, \phi_{N,\ell} \rangle = \phi_{N,k}(P_\ell).$$

This implies that:

$$\phi_{N,k}(P_\ell) = \begin{cases} 0, & \text{if } k \neq \ell; \\ \langle \phi_{N,k}, \phi_{N,k} \rangle = \|\phi_{N,k}\|^2, & \text{if } k = \ell. \end{cases}$$

Setting $d_k^{(N)} = \|\phi_{N,k}\|^2$ for each $k = 1, \ldots, J$ gives the required result.
The fact that (2) implies (1) is obvious by the same relation $\langle \phi_{N,k}, \phi_{N,\ell} \rangle = \phi_{N,k}(P_\ell)$. □

4.2.3 The Zernike MRA Definition

We now have all the tools required to define the Zernike MRA formally.

Definition 4.20 (*Zernike Multiresolution Analysis*)**.** Following Def. 4.6, the sequence of polynomial subspaces defined on the unit disk establishes a Multiresolution Analysis (MRA) of $L^2(B(0,1))$, which differs from the traditional construction given in Def. 2.1. Specifically, the sequence of subspaces $V_0 \cup \{V_{2^j}\}_{j \in \mathbb{N}_0}$ satisfies the following properties:

1. **Nested Subspaces:** The spaces form an ascending chain, $V_{2^j} \subset V_{2^{j+1}}$ for all $j \in \mathbb{N}_0$, creating a nested sequence starting from V_0:

$$V_0 \subset V_1 \subset \cdots \subset V_{2^j} \subset V_{2^{j+1}} \subset \cdots.$$

2. **Intersection:** The intersection of all subspaces reduces to the space of constant functions:

$$V_0 \cap \left(\bigcap_{j \in \mathbb{N}_0} V_{2^j} \right) = V_0 = \operatorname{span}\{\chi_{B(0,1)}\},$$

where $\chi_{B(0,1)}$ is the characteristic function of the unit disk.

3. **Completeness:** The union of the subspaces is dense in the space of square-integrable functions:

$$\overline{\bigcup_{j \in \mathbb{N}_0} V_{2^j}} = L^2(B(0,1)).$$

4. **Scale Invariance (Dilation):** An analogue to the dilation property holds, relating the polynomial degrees:
$$p(x, y) \in V_{2^j} \iff p(x^2, y^2) \in V_{2^{j+1}}.$$
5. **Translation (Localization):** For $N = 2^j$, the set of scaling functions $\{\phi_{N,r}\}_{r=1}^{J}$ constructed using the regular points (Def. 4.13) constitutes a basis for V_N. This choice of localized basis functions plays the role of integer translates in a standard MRA.

◊

Remark 4.21. The distinctions between this Zernike MRA and the classical formalism are as follows:

- **One-sided sequence:** The nested subspace tower is not bi-infinite (i.e., it does not extend to $j \to -\infty$), as the coarsest space is constrained to be V_0.
- **Non-zero intersection:** Consequently, the intersection is V_0 rather than $\{0\}$. In the context of a bounded domain, this is not a drawback, as one cannot define a scale coarser than the domain itself.
- **Localization vs. Translation:** The basis is generated by localization around nodes on the disk rather than by integer translations.

◊

4.2.4 Zernike Wavelets

Once the multiresolution structure of nested spaces V_N is established, we can define the wavelet spaces as the orthogonal complements that capture the detail information lost when moving from a finer resolution to a coarser one.

Definition 4.22 (*Wavelet Spaces*)**.** For any $N \in \mathbb{N}$, we define the *wavelet space* W_N as the orthogonal complement of V_N in the finer space V_{2N}. That is,
$$V_{2N} = V_N \oplus W_N.$$

Following Thm. 4.9, we can span the detail space in terms of the Zernike basis. Thus, W_N spans the polynomials with radial degrees strictly between N and $2N$:
$$W_N = \operatorname{span}\left\{Z_n^m : N < n \leq 2N, |m| \leq n, n - |m| \text{ even}\right\},$$
the dimension of which is given by $D_N := \dim(V_{2N}) - \dim(V_N) = \frac{3N(N+1)}{2}$.

For the base case $N = 0$, we define the wavelet space W_0 as the orthogonal complement of V_0 in V_1. That is,
$$W_0 := V_1 \ominus V_0 = \operatorname{span}\left\{Z_1^{-1}, Z_1^1\right\},$$
the dimension of which is $D_0 := \dim(V_1) - \dim(V_0) = 2$. ◊

We have explicitly shown the orthonormal bases of these detail spaces (the Zernike polynomials restricted to high frequencies), but for the MRA we also seek the spatial localization of the bases elements. Analogous to the construction of scaling functions via Kernel polynomials, the Zernike wavelets are defined using kernels to ensure this localization. However, instead of a single kernel, they are constructed from the *difference* between kernels at two different scales.

Definition 4.23 (*Zernike Wavelet Functions*)**.** Let $N \in \mathbb{N}$ and let the set $\{Q_j^{(N)} = (\mu_j^{(N)}, \omega_j^{(N)})\}_{j=1}^{D_N}$ be a suitable set of parameter points in the unit disk. The *Zernike wavelet functions* $\{\psi_{N,j}\}_{j=1}^{D_N}$ are defined as:

$$\psi_{N,j}(r,\phi) := K_{2N}(r,\phi;\mu_j^{(N)},\omega_j^{(N)}) - K_N(r,\phi;\mu_j^{(N)},\omega_j^{(N)}). \tag{4.22}$$

Expanding the kernels in terms of Zernike polynomials, this difference captures exactly the frequencies in W_N. Let $J_N := \dim(V_N)$; we have the following representation:

$$\psi_{N,j}(r,\phi) = \sum_{n=N+1}^{2N} \sum_{\substack{m=-n \\ n-|m| \text{ even}}}^{n} Z_n^m(\mu_j^{(N)},\omega_j^{(N)}) Z_n^m(r,\phi) = \sum_{l=J_N}^{J_{2N}-1} Z_l(\mu_j^{(N)},\omega_j^{(N)}) Z_l(r,\phi).$$

For the base case $N = 0$, we similarly define using the points $\{Q_j^{(0)}\}_{j=1}^{D_0}$:

$$\psi_{0,j}(r,\phi) := K_1(r,\phi;\mu_j^{(0)},\omega_j^{(0)}) - K_0(r,\phi;\mu_j^{(0)},\omega_j^{(0)}). \tag{4.23}$$

The expansion for this case contains only the terms of degree $n = 1$ (since the $n = 0$ term cancels out):

$$\begin{aligned}\psi_{0,j}(r,\phi) &= Z_1^{-1}(\mu_j^{(0)},\omega_j^{(0)}) Z_1^{-1}(r,\phi) + Z_1^1(\mu_j^{(0)},\omega_j^{(0)}) Z_1^1(r,\phi) \\ &= Z_1(\mu_j^{(0)},\omega_j^{(0)}) Z_1(r,\phi) + Z_2(\mu_j^{(0)},\omega_j^{(0)}) Z_2(r,\phi).\end{aligned}$$

◊

Remark 4.24 (*Choice of Parameter Points*)**.** Selecting a subset of parameter points that guarantees the linear independence and stability of the wavelet basis is non-trivial. While the paper [15] uses random sampling of D_N points from the J_{2N} regular points of V_{2N}, a robust numerical approach involves computing *Approximate Fekete Points* (AFP) as proposed by Sommariva and Vianello [42].

The procedure operates on the rectangular Vandermonde matrix V constructed by evaluating the basis of W_N on a dense discretization mesh (e.g., the regular points of the finer scale V_{2N}). The goal is to extract a square submatrix with maximum determinant. This is achieved efficiently using a *QR factorization with column pivoting* on the transpose matrix V^T:

$$V^T P = QR,$$

where P is a permutation matrix. The first D_N columns selected by the pivoting strategy correspond to the points in the mesh that maximize the linear independence of the basis functions, thereby minimizing the condition number of the resulting system.◊

These wavelet functions satisfy properties analogous to those of the scaling functions, specifically localization and orthogonality to the approximation space.

Theorem 4.25 (*Properties of Zernike Wavelets*)**.** *Let $\{\psi_{N,j}\}_{j=1}^{D_N}$ be the set of wavelets defined above. The following properties hold:*

1. ***Orthogonality to Scaling Functions:*** *The wavelets are orthogonal to the scaling functions of the coarser scale V_N. For any scaling function $\phi_{N,k} \in V_N$:*

$$\langle \psi_{N,j}, \phi_{N,k} \rangle = 0.$$

2. ***Localization:*** *The wavelet $\psi_{N,j}$ is localized around its parameter point $Q_j^{(N)}$. It is the solution to the minimization problem:*

$$\left\| \frac{\psi_{N,j}(r,\phi)}{\psi_{N,j}(Q_j^{(N)})} \right\| = \min \left\{ \|\psi\| : \psi \in W_N, \psi(Q_j^{(N)}) = 1 \right\}.$$

3. ***Inner Product:*** *The inner product of two wavelets is given by the evaluation of one wavelet at the parameter point of the other:*

$$\langle \psi_{N,j}, \psi_{N,k} \rangle = \psi_{N,k}(Q_j^{(N)}).$$

$\Diamond$

Proof. 1. By definition, $\psi_{N,j} \in W_N$. Since W_N is the orthogonal complement of V_N and $\phi_{N,k} \in V_N$, the inner product is zero by construction. Alternatively, using the kernel definition and its reproducing property:

$$\langle K_{2N}(\cdot; Q_j) - K_N(\cdot; Q_j), \phi_{N,k} \rangle = \phi_{N,k}(Q_j) - \phi_{N,k}(Q_j) = 0,$$

where we used the reproducing property of both kernels on the function $\phi_{N,k}$ (which belongs to both V_{2N} and V_N).

2. This proof mirrors Lem. 4.12 but restricted to the subspace W_N. The function defined as the kernel difference acts as the reproducing kernel for the subspace W_N.

3. Using the kernel definition:

$$\langle \psi_{N,j}, \psi_{N,k} \rangle = \langle K_{2N}(\cdot; Q_j) - K_N(\cdot; Q_j), \psi_{N,k} \rangle.$$

Since $\psi_{N,k} \in W_N \subset V_{2N}$, the first term reproduces $\psi_{N,k}(Q_j)$. Since $\psi_{N,k} \perp V_N$ and $K_N(\cdot; Q_j) \in V_N$, the second term is zero. Thus, the result is $\psi_{N,k}(Q_j)$.

□

4.2.5 Multiresolution Decomposition and Spatial-Frequency Localization

Similarly to the Subsection 2.1.5 we can decompose the approximation spaces V_N as direct sums of the detail spaces W_N. For a given function f and a resolution level N (where N is a power of 2), the best approximation of f in the space V_N is given by the orthogonal projection $f_N = \text{proj}_{V_N}(f)$. Using the wavelet spaces defined in Def. 4.22, we can decompose the approximation space V_N recursively:

$$V_N = V_{N/2} \oplus W_{N/2} = V_{N/4} \oplus W_{N/4} \oplus W_{N/2} = \cdots = V_0 \oplus W_0 \oplus \cdots \oplus W_{N/2}. \quad (4.24)$$

Since V_0 is spanned by constant polynomials, this implies that the projection f_N can be written as a linear combination of wavelets at different scales.

This representation offers clear advantages compared to using the Zernike basis or the scaling functions separately. Regarding the standard Zernike basis, the polynomial degrees represent the frequencies. While a truncation of the Fourier-Zernike series (4.7) gives a good approximation, it lacks spatial information, as one does not know *where* in the disk those frequencies are located.

On the other hand, the scaling functions $\{\phi_{N,j}\}$ provide the necessary spatial localization. However, since they are kernel polynomials, they are composed of all Zernike polynomials up to degree N. Therefore, approximating f with scaling functions gives spatial information but does not provide specific information on the frequency content.

The wavelet decomposition in (4.24) solves this by providing simultaneous localization. In this representation, the *frequency* information is given by the specific subspace W_j (which corresponds to a specific range of polynomial degrees), whereas the *spatial* location is determined by the parameter points Q_j used to construct the wavelets in that subspace.

4.3 The Discrete Zernike Transform

While the previous sections established the theoretical framework in the continuous domain $L^2(B(0,1))$, practical applications involve signals sampled at discrete points. In this section, we formalize the *Discrete Zernike Transform* (DZT).

Let $f(x)$ be a function defined on the unit disk, and let $f \in \mathbb{R}^D$ be the vector of its samples evaluated at an arbitrary set of points $\{P_i\}_{i=1}^D \subset B(0,1)$ (typically with $D \geq \dim(V_N)$):

$$f = [f(P_1), f(P_2), \ldots, f(P_D)]^T.$$

To understand the DZT of a given function, we first examine a simplified approach: projecting the signal onto the global Zernike basis to calculate its Moments. Subsequently, we proceed by projecting it onto the localized Multiresolution Analysis basis to perform the Wavelet Decomposition. Both stages rely on the same linear algebra principles: constructing a basis matrix and inverting it via least squares.

4.3.1 Base Case: Zernike Moments

Given a radial degree $N \in \mathbb{N}_0$, we first aim to approximate $f(x)$ as a linear combination of the Zernike polynomials family $\mathcal{Z}_N$. Let $J = \dim(V_N)$. In the discrete setting, this synthesis equation becomes a matrix-vector multiplication:

$$f \approx \sum_{k=1}^{J} c_k Z_k, \tag{4.25}$$

where $Z_k = [Z_k(P_1), \ldots, Z_k(P_D)]^T$ is the discretized k-th Zernike polynomial evaluated at the sampling points. The coefficients $\{c_k\}_{k=1}^J$ are called the *Zernike Moments*.

This system can be written in matrix form as:

$$f = \mathcal{V}_N c, \tag{4.26}$$

where $\mathcal{V}_N \in \mathbb{R}^{D \times J}$ is the rectangular *Zernike Vandermonde Matrix*, defined as $(\mathcal{V}_N)_{ik} = Z_k(P_i)$.

Since we typically deal with oversampled grids ($D \geq J$), the matrix is not square. To find the coefficient vector c, we employ the *Moore-Penrose pseudoinverse*, denoted as $\mathcal{V}_N^\dagger$. The Zernike moments are obtained by:

$$c = \mathcal{V}_N^\dagger f = (\mathcal{V}_N^T \mathcal{V}_N)^{-1} \mathcal{V}_N^T f. \tag{4.27}$$

This operation provides the unique solution that minimizes the least-squares error $\|\mathcal{V}_N c - f\|_2$.

Remark 4.26. It is worth noting that since the scaling functions $\{\phi_{N,j}\}$ also form a basis for V_N, one could technically perform this approximation using the matrix $\mathcal{M}_N$ (where entries are $\phi_{N,j}(P_i)$) instead of the defined $\mathcal{V}_N$ matrix. However, the Zernike Moments approach is presented here as the fundamental base case. It demonstrates how to extract coefficients for the space V_N using the spectral basis directly. While it lacks the spatial localization of the scaling functions, it avoids the strict necessity of using the specific regular points grid required for the stability of $\mathcal{M}_N$, allowing for more flexible sampling schemes. ◊

4.3.2 Discrete Wavelet Decomposition

We now extend the logic of the previous section to the full Multiresolution Analysis. The goal is to solve the system for the hierarchical decomposition:

$$V_N = V_0 \oplus W_0 \oplus \cdots \oplus W_{N/2}.$$

To do this, we must construct a global basis matrix $\mathcal{B}_{MRA}$. While the block corresponding to V_0 is simply the first column of the $\mathcal{V}_N$ matrix, the blocks for the wavelet spaces W_N require a specific construction based on the difference of kernels.

Recall that a wavelet $\psi_{N,j}$ in subspace W_N is defined by a weighted sum of Zernike polynomials. Specifically, for a sampling point P:

$$\psi_{N,j}(P) = \sum_{k=J_N+1}^{J_{2N}} Z_k\left(Q_j^{(N)}\right) Z_k(P),$$

where $Q_j^{(N)}$ is the corresponding Approximate Fekete Point (from Rem. 4.24). Thus, the matrix block Ψ_N (whose columns are the discretized wavelets) can be expressed as a product of two matrices:

1. A partial Vandermonde matrix $\mathcal{V}^{(W_N)}$ containing only the columns of $\mathcal{V}_{2N}$ corresponding to the frequencies (degrees) present in the detail space W_N (indices $J_N + 1$ to J_{2N}).
2. A coefficient matrix derived from evaluating these polynomials at the parameter points $\{Q_j^{(N)}\}$.

Specifically, let $\mathcal{V}^{(W_N)} \in \mathbb{R}^{D\times D_N}$ be the Vandermonde block on the sample points $\{P_i\}$, and let $\mathcal{Z}^{(Q_N)} \in \mathbb{R}^{D_N\times D_N}$ be the matrix where entries are the Zernike polynomials evaluated at the parameter points $Q_j^{(N)}$. The wavelet block is constructed as:

$$\Psi_N = \mathcal{V}^{(W_N)} \cdot \left(\mathcal{Z}^{(Q_N)}\right)^T. \tag{4.28}$$

Finally, we assemble the full matrix:

$$\mathcal{B}_{MRA} = \left[\Phi_0 \mid \Psi_0 \mid \Psi_1 \mid \cdots \mid \Psi_{N/2}\right].$$

The decomposition of the signal f into the wavelet coefficients d is obtained by applying the Moore-Penrose pseudoinverse to this new basis matrix:

$$d = \mathcal{B}_{MRA}^{\dagger} f. \tag{4.29}$$

This single operation projects the signal onto the entire hierarchy of spaces simultaneously, yielding the coefficients that provide the spatial-frequency localization discussed in the previous section.

This concludes the theoretical framework. We have rigorously constructed the Haar MRA on $\mathbb{R}$ and $\mathbb{R}^2$, as well as the Zernike MRA on the unit disk, presenting the algorithms required to compute their respective moments and wavelets.

A critical distinction emerges from their geometric constructions: while Haar functions are strictly axis-aligned, Zernike polynomials are defined in polar coordinates. This radial definition naturally endows both Zernike Moments and Zernike Wavelets with *rotational invariance*.

These and other key mathematical properties are summarized in Table 4.1. The table evaluates the methods based on characteristics critical for image analysis: *Orthonormality* ensures non-redundant basis functions; *Scale and Rotation Invariance* measure feature stability under geometric transformations; and *Space/Frequency Localization* describes the ability to resolve features simultaneously in specific spatial regions and frequency bands.

Table 4.1: Comparison of Theoretical Properties of Spectral Feature Extractors.

Method	Orthonormality	Scale Inv.	Rotation Inv.	Spatial Loc.	Freq. Loc.
Haar Wavelets	✓	✓	✗	✓	✓
Zernike Moments	✓	✗	✓	✗	✓
Zernike Wavelets	✗	✓	✓	✓	✓

Pt. II will now focus on the practical application of these mathematical foundations, detailing their implementation and integration as differentiable layers within a Convolutional Neural Network for semantic food segmentation.

Part II

Integrating Wavelets and CNNs for Image Segmentation. Application to Food Analysis

Chapter 5

Context and Related Work

This chapter contextualizes our proposed WaveFood framework within the broader landscape of computer vision. We first review the evolution of semantic segmentation in the specific domain of *Food Computing*, highlighting the current shift towards computationally intensive *Transformer-based models*. Subsequently, we examine the integration of Spectral Methods into Deep Learning, with a specific focus on orthogonal moments, identifying the gaps that motivate our efficient CNN-based approach. Finally, synthesizing these insights, we conclude by defining the specific research objectives that drive the design of the WaveFood framework.

5.1 Semantic Segmentation in Food Computing

Food computing has emerged as a crucial field with applications ranging from guiding human behaviour and improving human health to understanding culinary culture [31]. We take special interest in the semantic segmentation of food images, defined as the classification of each pixel into different food categories. Unlike standard object segmentation (e.g., cars, pedestrians), food segmentation presents unique challenges due to the *amorphous nature* of food dishes. Their appearance varies significantly, resulting in high intra-class variance and low inter-class variance, which makes simultaneous category classification and distinct boundary detection particularly difficult.

Semantic food segmentation is also aligned with the objectives of several European research initiatives, as they seek to automate dietary monitoring and food analysis. Specifically, it is an essential step for projects where accurate pixel-wise classification is a prerequisite for extracting high-level nutritional information e.g. food intake monitoring from food pictures taken by a smartphone.

However, image segmentation remains a highly difficult Computer Vision problem. While recent foundation models offer impressive capabilities, there remains a need for efficient, domain-specific feature extractors capable of capturing texture and shape invariances effectively without the computational overhead associated with such massive models.

5.1.1 Evolution of Architectures

Early efforts in the domain adapted classical semantic segmentation architectures to food tasks. While initial approaches relied on hand-crafted features [7], these were quickly superseded by Convolutional Neural Networks (CNNs). Models such as FPN [27], attention-augmented CNNs [4], and the DeepLab family [10] (typically uti-

lizing ResNet [21] backbones) obtained strong performance when training and test distributions were closely matched [13, 4].

More recently, the focus has shifted toward transformer-based backbones, such as Swin [28] and SeTR [54], and high-resolution refinement networks like BiRefNet [55]. These architectures have further improved per-frame mask quality and boundary precision. However, while they perform well on standard top-down images, they are vulnerable to *domain shift*, meaning their accuracy degrades significantly when camera angles, lighting, or food presentation differ from the training set.

5.1.2 State-of-the-Art in Food Segmentation

The field is currently leveraging *Vision Transformers (ViTs)* [45] and large Foundation Models [6] to improve flexibility across visual domains. This includes promptable models and large multimodal models (LMMs).

- **Foundation Models (FoodSAM & FoodLMM)**: The Segment Anything Model (SAM) [25] demonstrated that high-quality masks can be produced from various prompts. Adaptations like *FoodSAM* [26] fuse semantic predictions with SAM proposals to recover fine-grained masks, achieving impressive zero-shot performance, though at a massive computational cost. Furthermore, specialized adaptations like *FoodLMM* [20] show potential for jointly reasoning about ingredient semantics, though their performance on unconstrained video is still nascent.

- **Video and Temporal Consistency (FoodMem)**: To address the limitations of frame-by-frame segmentation, recent methods exploit inter-frame information. Frameworks like XMem [12] and DEVA [11] utilize memory-augmented features to propagate masks across sequences. In the food domain, *FoodMem* [3] represents the current State-of-the-Art on benchmarks like FoodSeg103 [50]. It combines a transformer-based image segmenter (SeTR) [54] with a memory tracker (XMem2) to substantially reduce mask flicker and improve completeness in 360° food videos.

While Transformer-based approaches currently establish the accuracy State-of-the-Art, their quadratic complexity (see Sec. 6.2.2) makes them prohibitive for real-time applications on resource-constrained devices, such as mobile dietary logging. To bridge this gap, our research pivots back to optimizing the fundamental efficiency of *CNNs*. We hypothesize that by replacing standard spatial convolutions with spectrally-rich layers, we can equip lightweight CNN architectures with the texture-awareness and geometric invariance usually reserved for heavier models, thus balancing the trade-off between high-fidelity segmentation and computational feasibility.

5.2 Spectral Methods in Deep Learning

To address the efficiency limitations of standard spatial convolutions, recent research has revisited *Spectral Analysis* (decomposing images into frequency components) as a powerful inductive bias for Deep Learning. In this work, we focus specifically on three spectral transforms: *2D-FHT* (Sec. 3.2), *Zernike Projection*, and *DZT* (Sec. 4.3), corresponding to Haar Wavelets, Zernike Moments, and Zernike Wavelets extractors.

5.2.1 State-of-the-Art in Spectral Architectures

While spectral theory is well-established in signal processing [29], its integration as differentiable layers within modern Deep Learning architectures represents an active area of research.

Haar Wavelets in Deep Learning

Standard spatial downsampling operations, such as Max Pooling, often result in the aliasing or loss of high-frequency texture details. To mitigate this, Haar Wavelets have been integrated into CNNs to perform lossless downsampling, preserving structural information in the frequency domain. Notable works like [52] demonstrate that wavelet-based pooling can significantly improve segmentation boundaries. Furthermore, extremely recent approaches have begun combining multi-level wavelet spectrums with advanced Diffusion Transformers for super-resolution tasks [47], validating the continued relevance of Haar analysis in high-fidelity image reconstruction.

Zernike Moments in Deep Learning

Unlike square wavelets, *Orthogonal Moments* such as Zernike Moments possess intrinsic rotation invariance and high information packing capabilities on the unit disk. Historically used as hand-crafted descriptors [43], recent efforts have successfully integrated them into neural pipelines to enforce rotational equivariance, i.e., a rotation of the input image results in a corresponding rotation of the output feature map. The seminal work on *Harmonic Networks* [48] introduced the concept of using circular harmonics within CNNs to achieve deep translation and rotation equivariance. More recently, the potential of these moments was demonstrated in *MomentsNeRF* [2]. In the context of Neural Radiance Fields [30], the authors established that Zernike Moments are superior for representing high-frequency texture details and reconstructing complex surface reflectance from limited data, outperforming standard grid-based encodings.

Zernike Wavelets: A Research Frontier

While Zernike Moments provide global rotation invariance, they lack local spatial resolution. *Zernike Wavelets* theoretically bridge this gap by offering simultaneous spatial localization and rotational properties. The mathematical foundations for these wavelets on the unit disk have been recently formalized by Datta and Datta [15]. However, their application in discriminative Deep Learning remains a novel and largely unexplored frontier. To the best of our knowledge, no existing CNN architectures fully integrate discrete Zernike Wavelet transforms as end-to-end differentiable layers for semantic segmentation. Our work aims to pioneer this integration, leveraging the theoretical advantages of Zernike Wavelets to capture the complex, amorphous nature of food.

5.3 Our Goal

Our goal is the novel integration of *2D-FHT*, *Zernike Projection* and *DZT* as trainable, differentiable layers within a powerful convolutional segmentation network, a pipeline we denominate *WaveFood*. In the following chapters, the coefficients produced by the previous transforms will be denoted as *Haar Wavelets (HW)*, *Zernike Moments (ZM)* and *Zernike Wavelets (ZW)*. We hypothesize that, compared to the axis-aligned

HW, the circular nature of Zernike basis functions provides a better alignment with the organic structures found in food images, potentially enhancing feature extraction in complex food textures.

Chapter 6

Preliminaries

This chapter establishes the necessary engineering context for the proposed method. While Part I focused on the mathematical construction of wavelets on continuous domains and their discrete algorithms, we now transition to the Deep Learning framework. We formalize the fundamental operations and architectures of Convolutional Neural Networks (CNNs) used as our baseline.

6.1 Fundamentals of Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialized class of neural networks designed to process grid-like topology data, such as images. Mathematically, they are compositions of linear operations (*convolutions*) and non-linear activation functions, optimized to minimize an error metric, denoted as the *loss function*.

In this section, we first introduce the core operations used in CNNs, later we detail the training framework and finally, we present the Encoder-Decoder architecture.

6.1.1 Core Operations

To rigorously define a CNN, we revisit the mathematical operations introduced in Part I, adapting them to the context of learned features.

Learned Convolution vs. Fixed Filters

In Sec. 3.1.2, we defined the discrete convolution (Def. 3.7) using fixed kernels (e.g., Haar's h, g). In a CNN layer, the kernel K is not fixed but contains *trainable parameters* learned via optimization. Typically, this kernel is defined as $K \in \mathbb{R}^{C \times H \times W}$ (where C is the number of input channels, and H, W are the spatial height and width).

In the context of the mathematical convolution definition, which formally involves infinite summations over $\mathbb{Z}^n$, this kernel is interpreted as having *compact support*. This means that the kernel values are assumed to be zero outside the specific spatial dimensions $H \times W$. This property effectively reduces the theoretical infinite definition to a finite sum over the kernel's domain, an operation algorithmically implemented as a *sliding window*.

Crucially, as noted in Rem. 2.21, Deep Learning frameworks implement this operation as a cross-correlation (sliding window without flipping). However, since the weights K are learned from the optimization problem of lowering the loss function, the network automatically learns the kernel in the orientation that best extracts features,

effectively absorbing any necessary "flip" into the learned parameters themselves. Thus, the kernel and the image can be conceptualized as traversing in the same direction.

Stride and Downsampling

As defined in Def. 3.9, the stride S corresponds to the step size of the convolution. In CNN architectures, a stride $S > 1$ acts as a *downsampling* operator, reducing the spatial resolution of the feature map by a factor of S. This reduces computational cost and increases the receptive field of subsequent layers.

Receptive Field and Dilated Convolution

The *receptive field* is the region of the input image that contributes to a specific feature in a deeper layer. While standard downsampling increases the receptive field, it degrades resolution. To expand the receptive field without losing resolution, we employ *dilated convolutions*. If a standard convolution is defined on adjacent pixels, a dilated convolution with rate d inserts $d-1$ zeros between kernel elements, effectively expanding its coverage without increasing the number of parameters. This can be visualized in Fig. 6.1, where this process resembles separating the kernel values.

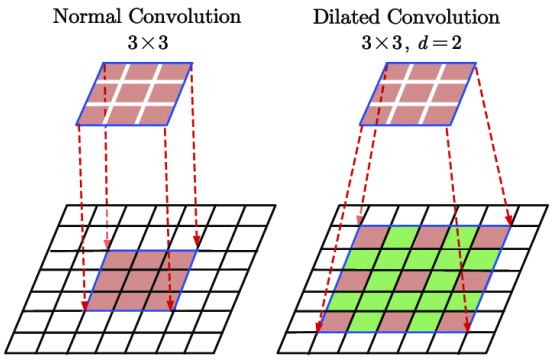


Figure 6.1: Comparison between normal convolution and dilated convolution.

Pooling

Pooling is a non-linear downsampling operation used to reduce spatial dimensions.

- **Average Pooling:** Computes the mean of a local window. As noted in Rem. 3.4, this is mathematically equivalent to the Haar approximation coefficient calculation.
- **Max-Pooling:** Outputs the maximum value within the sliding window.

Upsampling

Since semantic segmentation requires a prediction for every pixel in the original resolution ($H \times W$), the network must eventually recover the spatial dimensions reduced by downsampling. This operation, denoted as *upsampling*, acts as the inverse of pooling.

Skip Connections

Deep architectures often suffer from information loss and optimization difficulties (vanishing gradients, stated in Sec. 6.1.2) as depth increases. A *skip connection* is a structural mechanism that allows the network to bypass intermediate layers, feeding the input of a layer directly to a deeper layer. Two primary variants are essential in segmentation:

- **Residual Connection (Add):** The input x of a given layer is added element-wise to its output $F(x)$, yielding $y = F(x) + x$.
- **Concatenation:** Common in Encoder-Decoder designs (presented in Sec. 6.1.3). High-resolution feature maps from the early layers are concatenated along the channel dimension with the upsampled features of the decoder. This reintroduces fine spatial details (edges, textures) that were lost during pooling, which is critical for accurate boundary segmentation.

6.1.2 Training Framework: Deep Supervision

A Neural Network operates in two distinct modes:

- **Inference:** The forward propagation of an input x through the network layers to produce a prediction $\hat{y}$.
- **Training:** The iterative optimization process where parameters are updated.

Loss Function and Optimization

Training is guided by a *Loss Function* $\mathcal{L}(\hat{y}, y_{gt})$ that quantifies the discrepancy between the network's prediction $\hat{y}$ and the Ground Truth y_{gt} i.e., what the neural network should give in the ideal case of perfect functioning. In semantic segmentation tasks, the standard choice is the *Cross-Entropy Loss*, which measures the information-theoretic divergence between the predicted probability distribution over classes and the true target distribution. Specifically, for an image with N pixels and a set of C semantic classes, the pixel-wise Cross-Entropy loss is defined as:

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log(\hat{y}_{i,c}) \tag{6.1}$$

where:

- N denotes the total number of pixels in the input image (i.e., $H \times W$).
- C is the total number of semantic categories (including background).
- $y_{i,c}$ is the binary indicator (0 or 1) of the ground truth y_{gt}, which takes the value 1 if the pixel i truly belongs to class c, and 0 otherwise.
- $\hat{y}_{i,c}$ represents the predicted probability that pixel i belongs to class c, typically obtained via a Softmax activation function, i.e., $\hat{y}_{i,c} = \frac{e^{z_{i,c}}}{\sum_{j=1}^{C} e^{z_{i,j}}}$, where $z_{i,c}$ is the value obtained via the last CNN layer for class c at pixel i.

The goal of training is to find the set of parameters θ that minimizes this loss. This is achieved via *Gradient Descent*, an iterative optimization algorithm that updates the parameters in the direction opposite to the gradient of the loss function ($\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$, where η is the learning rate and $\nabla_\theta \mathcal{L}$ is the gradient of the loss function regarding the given parameter θ). To compute these gradients efficiently across the multiple layers of the network, we employ *Backpropagation*, an algorithm that recursively applies the chain rule of calculus from the output layer back to the input.

Deep Supervision

In very deep networks, gradients can vanish before reaching the early layers during backpropagation. To mitigate this, we employ *Deep Supervision* [53]. This technique attaches an auxiliary segmentation head to an intermediate stage (layer) of the network. The total training loss is a weighted sum:

$$\mathcal{L}_{total} = \mathcal{L}_{main} + \lambda \cdot \mathcal{L}_{aux}, \tag{6.2}$$

where λ balances the contribution (commonly $\lambda = 0.4$). This auxiliary signal accelerates convergence and improves feature learning in lower layers.

Fig. 6.2 shows the behaviour of this Deep Supervision method applied in the presented Encoder-Decoder architecture, while using more than one auxiliary head in different intermediate steps on a convolutional neural network

6.1.3 The Encoder-Decoder Architecture

Combining the operations defined in Sec. 6.1.1, we can describe the standard topological framework for semantic segmentation: the *Encoder-Decoder* architecture. As illustrated in Fig. 6.2, this architecture typically consists of two distinct paths:

- **The Encoder (Backbone):** Enclosed within the *blue dotted outline* on the left. This contracting path applies a sequence of learned convolutions and downsampling operations (pooling or strides) to progressively reduce the spatial resolution while increasing the feature depth (channels). Its goal is to extract abstract, high-level semantic features from the input image.
- **The Decoder (Head):** Enclosed within the *red dotted outline* on the right. This expanding path utilizes upsampling operations to recover the original spatial resolution ($H \times W$) from the encoded features. Its goal is to spatially localize the semantic features to generate the final pixel-wise segmentation mask.

Crucially, as shown by the *solid upper arrows* in Fig. 6.2, the architecture employs *skip connections* to transfer high-frequency spatial details directly from the Encoder layers to their corresponding Decoder layers via channel concatenation. This ensures that the fine boundaries lost during the encoding downsampling are preserved in the final output. Furthermore, the diagram illustrates several auxiliary Deep Supervision heads via the downward arrows labeled “Loss” that calculate the loss at the output of each decoding stage.

6.2 Baseline Architecture: The CCNet Framework

To validate the proposed spectral layers, we require a robust and standardized baseline. In this work, we adopt the architecture known as *Criss-Cross Network* (CC-

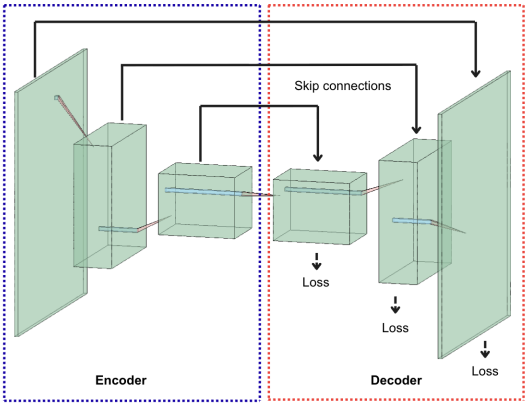


Figure 6.2: Schematic of an Encoder-Decoder architecture implementing Deep Supervision. The diagram visualizes the network layers as 3D green blocks. The contracting Encoder (blue frame) extracts features, while the expanding Decoder (red frame) reconstructs resolution. Key components include skip connections (upper solid arrows) linking corresponding layers and intermediate loss calculation points (downward arrows) for deep supervision.

Net) [24]. It is crucial to clarify that, throughout this thesis, this designation implies a composite model consisting of two distinct parts:

1. **Backbone:** A *ResNet50* [22] acting as the feature extractor (Encoder).
2. **Head:** The *Criss-Cross Attention* module acting as the context aggregator (Decoder).

The complete architecture follows the Encoder-Decoder strategy defined in Section 6.1, implementing Deep Supervision during training.

6.2.1 Backbone: ResNet50

The feature extraction backbone is a variant of the standard Residual Network (ResNet) [21].

The Residual Block

Standard deep networks suffer from degradation problems where accuracy saturates as depth increases. ResNet solves this by introducing the *Residual Block*. Instead of learning a direct mapping $H(x)$, the network learns a residual function $F(x) := H(x) - x$. The original mapping is reconstructed via a skip connection: $y = F(x) + x$. This structure, mathematically defined in Section 6.1.1, allows gradients to flow unimpeded through the network, enabling the training of very deep architectures (e.g., 50 or 101 layers).

To optimize computational efficiency, ResNet50 employs a *Bottleneck Block* design, stacking them in 4 different stages, as shown in Tab. 6.1. As implied by the name, this block follows a "wide-narrow-wide" structure shown in Fig. 6.3:

(a) A 1×1 convolution reduces the channel dimensionality (compression).

(b) A 3×3 convolution processes the spatial features in this lower-dimensional space.

(c) A 1×1 convolution expands the dimensionality back to the original size.

This "sandwich" structure allows for increased depth with fewer parameters compared to standard convolutional blocks.

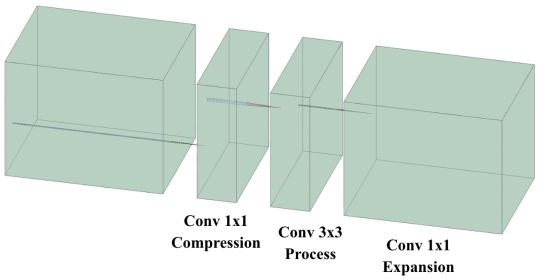


Figure 6.3: Schematic of the ResNet Bottleneck Block. The diagram illustrates the "wide-narrow-wide" channel topology, where features are compressed before the spatial convolution and subsequently expanded to optimize parameter efficiency.

V1c Improvements: Deep Stem and Output Stride

We utilize the *ResNet50-V1c* variant [22], which introduces two critical modifications for semantic segmentation tasks:

- **Deep Stem.** Standard ResNet architectures initiate processing with a single aggressive 7×7 convolution (stride 2) followed by a 3×3 max-pooling layer (stride 2). As shown in Fig. 6.4 (a), this rapidly downsamples the image, potentially discarding fine-grained spatial information essential for segmenting small food ingredients. In contrast, the V1c variant replaces this with a *Deep Stem* (Fig. 6.4 (b)): a stack of three consecutive 3×3 convolutions. The first layer performs the spatial downsampling (stride 2) while reducing channel complexity ($C = 32$), and the subsequent layers refine the features and expand the depth ($C = 64$) before entering the main backbone.

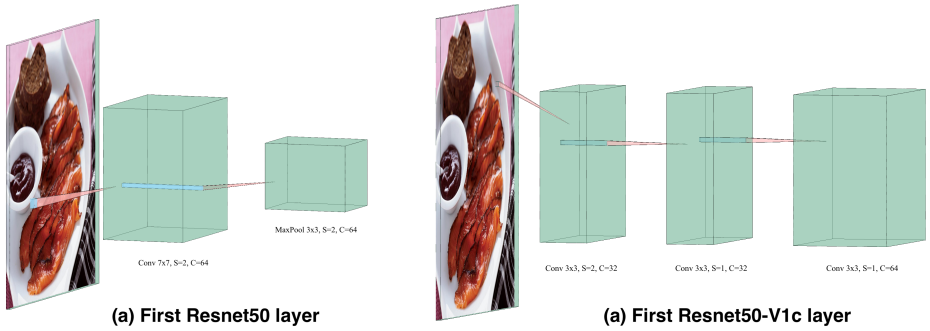


Figure 6.4: Structural comparison between the standard ResNet Stem and the Deep Stem. (a) The standard architecture employs a large 7×7 kernel followed immediately by pooling, causing an abrupt loss of spatial resolution. (b) The *ResNet50-V1c Deep Stem* decomposes this operation into three 3×3 convolutional layers, progressively increasing channels from $32 \rightarrow 64$.

- **Dilated Convolution Strategy.** Image classification networks typically downsample the input by a factor of 32 (Output Stride $OS = 32$). For dense prediction tasks like segmentation, an $OS = 32$ feature map is too coarse. To address this, ResNet50-V1c modifies the final stages (Stage 3 and Stage 4) to stop downsampling (stride 1) while maintaining the receptive field. This is achieved via *Dilated Convolutions* (defined in Sec. 6.1.1). As detailed in Table 6.1, Stage 3 uses a dilation rate $d = 2$, and Stage 4 uses $d = 4$. This strategy results in a final *Output Stride of 8*, providing a feature map with 4 times higher spatial resolution than the standard implementation.

Table 6.1: Specification of the ResNet50V1c Backbone. The **Stride** and **Dilation** columns highlight the strategy to preserve an Output Stride of 8 (1/8 resolution) in the final stages.

Stage	Blocks	Structure	Out Channels	Stride	Dilation	Resolution
Stem	-	3×3 Conv ($\times 3$)	64	2	1	1/2
Stage 1	3	Bottleneck	256	1	1	1/4
Stage 2	4	Bottleneck	512	2	1	1/8
Stage 3	6	Bottleneck	1024	1	2	1/8
Stage 4	3	Bottleneck	2048	1	4	1/8

6.2.2 Segmentation Head: Criss-Cross Attention

Once features are extracted by the backbone, the network must contextualize them. Mathematically, an *Attention Mechanism* is a function that dynamically weights feature vectors based on their similarity, allowing the model to focus on relevant dependencies regardless of their spatial distance.

While *Self-Attention* (central to Transformer models [45]) relates every pixel to every other pixel to capture global context, its computational complexity is quadratic, $O((H \times W)^2)$, making it prohibitive for high-resolution images.

CCNet addresses this bottleneck by introducing the *Criss-Cross Attention* module. Instead of attending to the whole image, each pixel only attends to other pixels in its same *row and column*. By stacking two such modules recurrently:

1. **First Step:** A pixel gathers information from its vertical and horizontal neighbours (sparse context).

2. **Second Step:** Since those neighbors also gathered information from their respective rows/columns, the information propagates to the entire image.

This mechanism achieves global receptive field (dense context) with a significantly reduced complexity of $O((H \times W)(H + W))$. Figure 6.5 illustrates this efficient contextual aggregation strategy, where we only have $(H + W - 1)$ weights different from 0 for each pixel in the image corresponding to their column and row neighbours.

In this work, we adopt the CCNet architecture as our strong baseline implementing *Deep Supervision*. Our objective is to demonstrate that adding spectral feature extraction convolutional layers before the baseline model improves segmentation accuracy without requiring deeper backbones.

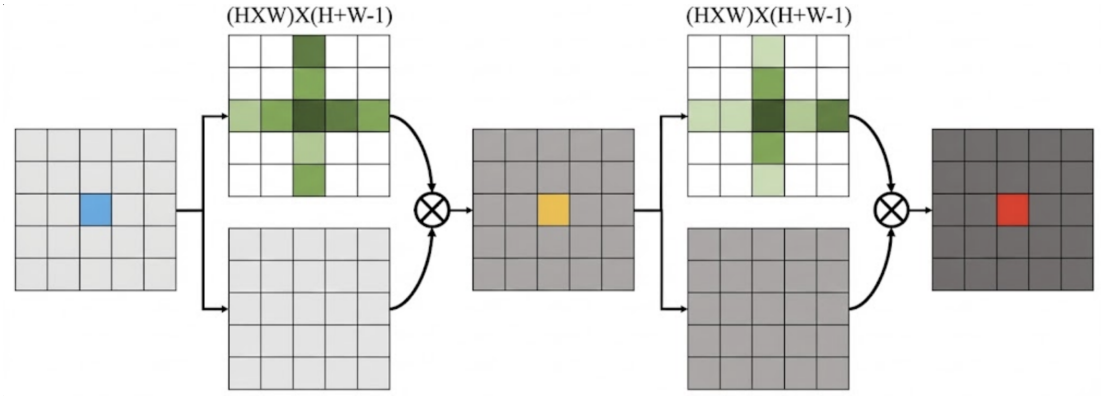


Figure 6.5: Schematic of the Criss-Cross Attention module. The system generates attention maps by restricting dependencies to the criss-cross path (row and column) of each pixel. This approach avoids the massive matrix multiplications of standard global self-attention while effectively aggregating contextual information from the entire image through recurrence.

Chapter 7

WaveFood: A Novel Methodology for Spectral Feature Integration into CNNs

This chapter details the practical implementation of the theoretical concepts established in Pt. I and their integration as differentiable CNN layers in a Deep Learning framework for semantic food segmentation.

7.1 WaveFood Framework Overview

Given the goal of segmenting RGB images entered into the system, our proposed WaveFood framework consists of four sequential processing stages, as illustrated in Fig. 7.1. Without loss of generality, we illustrate our proposal using a sample food image from the *FoodSeg103* dataset (introduced in Sec. 8.1.1):

(a) Spectral Convolutional Layer: This primary stage extracts frequency-domain image features. Depending on the specific configuration, it leverages the 2D-FHT (see Sec. 3.2) or Zernike transforms (see Sec. 4.3). As depicted in the figure, this layer can be instantiated via three distinct integration strategies: Direct Injection, RGB-Fusion, or the Multi-Level architecture (detailed in Sec. 7.4).

(b) ResNet50 Backbone: The spectral features are fed into a pre-trained ResNet50, which acts as a deep feature extractor.

(c) CCNet Head: The Criss-Cross Attention module performs contextual modeling and aggregates global information from the deep features.

(d) Prediction Segmentation Mask: The final stage generates the pixel-wise classification map for the different food classes.

The following sections detail the technical implementation of the framework, spanning from the development environment and preliminary validation to the integration of differentiable modules and the specifics of the proposed architectures.

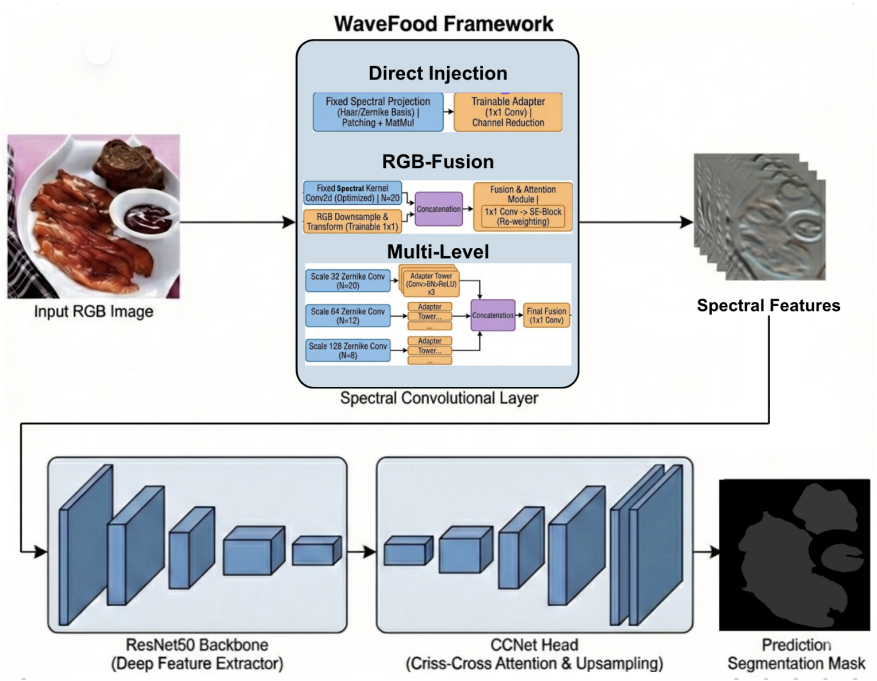


Figure 7.1: Schematic overview of the WaveFood Framework. The pipeline processes an input RGB image through a selectable Spectral Convolutional Layer (configured as Direct Injection, RGB-Fusion, or Multi-Level), followed by a standard Deep Learning baseline (ResNet50 Backbone and CCNet Head) to generate the final segmentation mask.

7.2 Preliminary Integration: Coefficient Extraction and Image Reconstruction

Prior to integrating spectral methods into the neural network architecture, we conducted a critical preliminary analysis to validate the feature extraction and image reconstruction quality. This validation was implemented in a series of `Python` notebooks, isolating the mathematical operations from the learning process to ensure numerical stability.

To facilitate this analysis, we selected a representative high-resolution sample (ID: 00004401) from the *FoodSeg103* dataset (fully described in Sec. 8.1.1), shown in Fig. 7.2. This specific image was chosen for its rich diversity of high-frequency textures and fine-grained details, which are critical for assessing spectral reconstruction fidelity. Furthermore, its native spatial resolution of 1024×1024 pixels was particularly advantageous for the preliminary Haar Wavelet experiments, as it aligns perfectly with the dyadic power-of-two dimensions required by the standard algorithm.

7.2.1 2D-FHT Details

For the Haar Wavelets extraction, the implementation adhered to the procedure described in Sec. 3.2. We utilized `PyTorch` convolutional layers with fixed kernels to apply the LL, LH, HL, and HH filters independently to each channel of the original RGB images. The coefficient extraction was designed to strictly match the Mallat decomposition scheme defined in Fig. 2.1.

Figure 7.2: Benchmark reference sample. This image serves as the standard subject for validating the reconstruction quality of the implemented spectral transforms throughout this chapter.

Visualizing these coefficients presented a challenge due to the dynamic range expansion inherent to the wavelet transform. Specifically, while the input intensity lies in $[0, 1]$, the first-level coefficients expand to $[-1, 1]$, and subsequent levels further increase this range (e.g., $[-2, 2]$). Consequently, we implemented a custom normalization function to map these unbounded ranges back to valid display intervals. Results can be found in Fig. 7.3, where we can appreciate the Mallat decomposition defined in Fig. 2.1, where the top-left component represents the coarse approximation (LL) at the second scale, retaining the global structure. The surrounding quadrants display the sparse detail coefficients (LH, HL, HH), rendered in mid-gray to represent zero values, while bright and dark pixels highlight positive and negative high-frequency edge information, respectively.

Figure 7.3: Normalized visualization of the 2-scale 2D-FHT. The feature map illustrates the hierarchical Mallat decomposition applied to the benchmark image, with coefficients rescaled for visual inspection.

Finally, the image reconstruction was achieved via `PyTorch` transposed convolu-

tional layers, effectively inverting the decomposition process to verify signal integrity.

Remark 7.1. Due to the rich texture content of food images, the resulting Haar coefficient maps are perceptually dense and difficult to validate through simple visual inspection. To ensure that the coefficients were correctly extracted and spatially arranged according to the Mallat decomposition, distinct from merely verifying invertibility through reconstruction, we employed a purpose-built synthetic test image.

As detailed in Fig. 7.4, the input is engineered with four distinct quadrants to isolate specific spectral responses: a low-frequency gradient (top-left), vertical high-frequency stripes (top-right), horizontal high-frequency stripes (bottom-left), and a frequency chessboard (bottom-right). These textures are specifically designed to solely stimulate the LL (Approximation), LH (Vertical Detail), HL (Horizontal Detail), and HH (Diagonal Detail) filters, respectively, confirming the correct spatial disentanglement of the signal frequencies. ◊

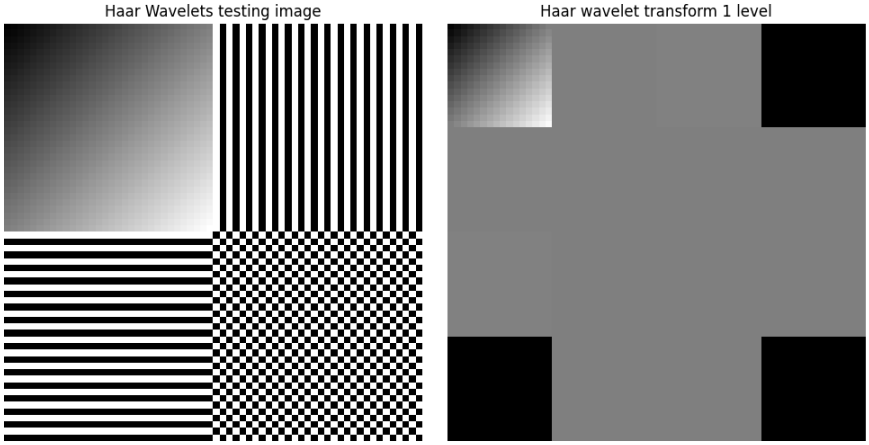


Figure 7.4: Validation of the Haar Wavelet spatial decomposition. The synthetic frequency test pattern (left) isolates specific directional components to verify the correct extraction and spatial arrangement of the Mallat decomposition coefficients (right).

Patch-Convolution Approach

Finally, to overcome the strict dimensional constraints (specifically the requirement for dimensions to be multiples of 2^L) and to maximize the representational capacity of the extracted features, the algorithm was adapted to operate on local *square patches* rather than the full global resolution.

This modification implements a sliding-window strategy, where a fixed-size extraction kernel systematically traverses the input image in a raster-scan order (from top-left to bottom-right). At each step, a local patch of the image is extracted, and the window shifts by a fixed number of pixels, denoted as the *stride* S. By setting this stride to be strictly smaller than the patch size P (e.g., $S < P$), successive windows overlap significantly, ensuring that every pixel is analyzed within multiple distinct local neighbourhoods. This method generates a spatially dense and over-complete representation of the input, significantly increasing the richness of the texture coefficients.

Thus, the global image reconstruction was performed by aggregating the individual inverse-transformed patches, where the final value of each pixel was computed as the

average of the contributions from all overlapping windows.

7.2.2 Zernike Projection and DZT Details

As established in the theoretical framework from Sec. 4.3, Zernike projection and DZT are defined on the unit disk $B(0,1)$. Consequently, our implementation maps the spatial input domain to this unit disk. The sampling points for the Zernike Vandermonde matrix $\mathcal{V}_N$ were selected strictly from the centred inbound "circle" of the image (or patch), masking out corner pixels to avoid boundary artifacts.

To construct $\mathcal{V}_N$, the radial polynomials $R_{nm}(r)$ were computed using the *Kintner recurrence relation* 4.4. This recursive approach prevents the numerical instability associated with high-order factorials in the direct definition. The final polynomials were obtained by multiplying these radial terms by the corresponding angular factors and normalization constants.

Crucially, since the number of pixels in the disk typically exceeds the number of Zernike polynomials (an overdetermined system), the inverse problem relies on a least-squares solution. To resolve this, we calculated the *Moore-Penrose pseudoinverse* ($\mathcal{V}_N^\dagger$), as defined in Sec. 4.3, using `PyTorch`. In our implementation, this matrix $\mathcal{V}_N^\dagger$ is precomputed and stored, reducing the coefficient extraction step to a single matrix-vector multiplication: $c = \mathcal{V}_N^\dagger f$.

Remark 7.2. We experimentally evaluated computing the pseudoinverse via the Normal Equations, i.e., $\mathcal{V}_N^\dagger = (\mathcal{V}_N^T \mathcal{V}_N)^{-1} \mathcal{V}_N^T$. While this method yielded marginally higher SSIM values in specific low-noise cases, it is theoretically prone to numerical instability when the Vandermonde matrix is ill-conditioned. Consequently, we opted for the SVD-based implementation provided by `PyTorch` (`pinv`) to ensure robustness and leverage library-level optimizations. ◊

For the Zernike Wavelets, the challenge shifts from simple projection to finding a discrete set of nodes that allows for stable multi-resolution analysis. Following the methodology described in Sec. 4.3.2, we implemented a node selection algorithm to construct a robust basis:

1. **Regular Points Generation:** We first generated a high-density grid of *regular points* on the unit disk following Def. 4.13. These points provide a uniform covering of the domain and serve as the candidate pool. Fig. 7.5 shows that these points verify that their distribution shifts towards the boundary of the disk as the polynomial order N increases, a characteristic behavior of well-conditioned polynomial interpolation nodes.

2. **Approximate Fekete Points (AFP):** To select the optimal interpolation nodes from this pool, we applied a *QR decomposition with column pivoting* on the transposed Vandermonde matrix constructed from the regular points. The pivots resulting from this decomposition identify the *Approximate Fekete Points* (AFP). These points maximize the determinant of the Vandermonde matrix, thereby minimizing the condition number and ensuring numerical stability.

   As illustrated in Fig. 7.6, the computed AFP for the detail space W_{16} constitute a specific subset of the regular points of the finer space V_{32} (previously shown in Fig. 7.5). It can be observed that the selection algorithm tends to preserve points near the boundary, maintaining the optimal interpolation properties required

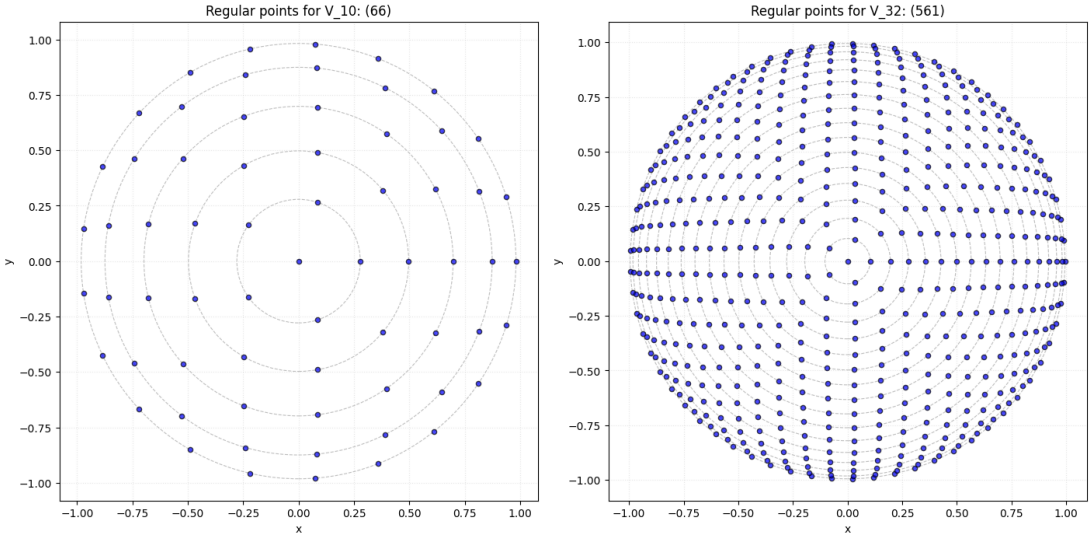


Figure 7.5: Visualization of generated Regular Points on the unit disk. The plots compare the distribution of the Regular Points extracted following Def. 4.13 for a polynomial order of $N = 10$ (left, 66 points) versus $N = 32$ (right, 561 points). As the order N increases, the density of the regular points visibly shifts towards the boundary of the disk. This characteristic clustering at the edges is a critical geometric property for determining well-conditioned interpolation nodes within the domain.

for stable spectral analysis. Preliminary benchmarks showed that this approach yields a slightly higher reconstruction SSIM compared to random sampling.

It is worth noting that for the correct MRA decomposition in the DZT algorithm, N must be a power of 2.

Circular Patching and Minimum Stride

A unique challenge of spectral methods on the disk is the geometric mismatch between the circular domain and the rectangular image grid. As evidenced in Fig. 7.7, applying the transform to the full image frame results in substantial blurring and a significant loss of high-frequency texture details. This demonstrates that the current polynomial order N is insufficient to encode complex spatial information over such a large support. Notably, the ZW reconstruction (b) presents slightly higher degradation and artifacting compared to the ZM baseline (a). These results validate the necessity of adopting a patch-based strategy, reducing the spatial support rather than exponentially increasing N, to mitigate these limitations.

Thus, we applied a convolutional patch technique similar to the FHT approach. However, unlike square patches, a non-overlapping grid of circles leaves significant spatial gaps at the interstices of the grid (specifically at the corners of the underlying squares), resulting in critical information loss. To mitigate this, the stride value (S) relative to the patch radius (R) must be adjusted to ensure sufficient overlap.

The optimal stride was initially approximated empirically, as visualized in Fig. 7.8. Theoretically, the condition for full coverage requires that the critical point (the centroid of four adjacent patches) lies within the circular boundary. The distance from this critical point to the nearest patch centre is $d = S/\sqrt{2}$. Therefore, ensuring full coverage

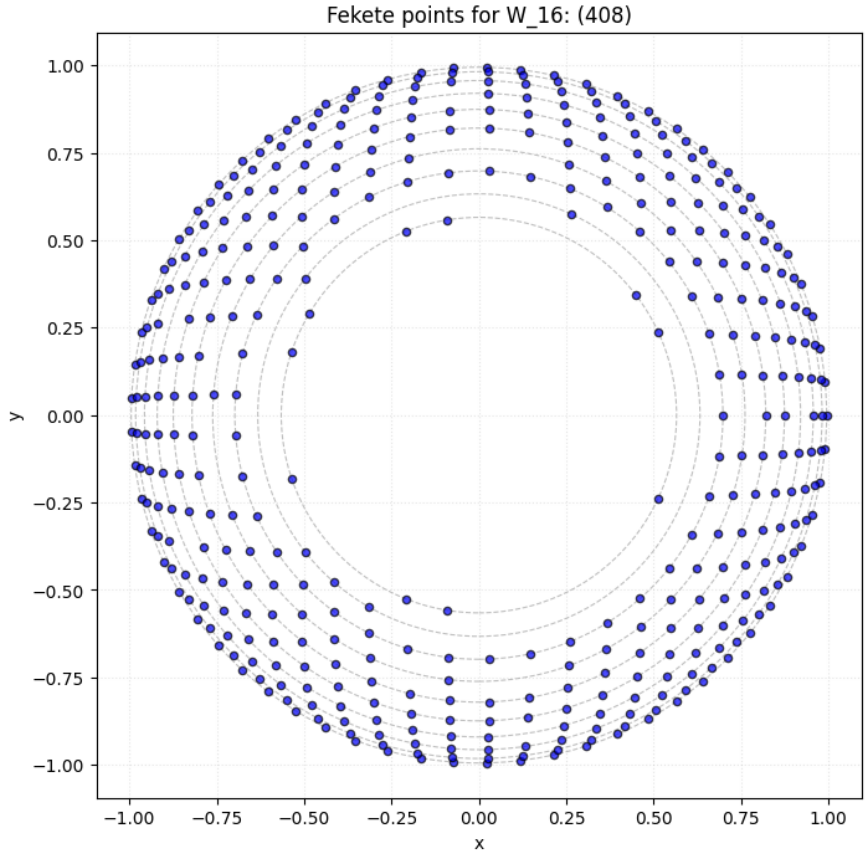


Figure 7.6: Visualization of Approximate Fekete Points (AFP) for W_{16}. The selected nodes (blue) form a subset of the V_{32} regular grid, preferentially clustering near the unit disk boundary to minimize the condition number of the interpolation matrix.

(a) Image reconstruction from Full-image ZM extraction.

(b) Image reconstruction from Full-image ZW extraction.

Figure 7.7: Visualization of spectral reconstruction limitations on the full benchmark image. The inverse transform results using (a) Zernike Moments and (b) Zernike Wavelets exhibit corner information loss due to circular masking and significant texture blurring, illustrating the inadequacy of global projection for high-resolution detail retention.

$(R \geq d)$ implies:

$$S \leq R\sqrt{2} \approx 1.41R. \tag{7.1}$$

In our implementation, we adopted strides with significant overlap $(S < R)$ to generate an over-complete representation, which correlated with substantially higher SSIM values. As shown in Fig. 7.8, while a non-overlapping tiling $(S = 64)$ leaves voids, reducing the stride progressively eliminates them, empirically achieving full pixel-wise coverage at $S = 46$, aligning with the theoretical bound derived in Eq. 7.1.

In the benchmark test image (Fig. 7.2), using Zernike Projection and DZT with patch size $P = 64$ (radius $R = 32$), order $N = 32$, and stride $S = 4$, we achieved SSIM values of approximately 0.95. Furthermore, we verified that increasing N consistently improved the SSIM, confirming the nestedness of the approximation spaces V_N and the method's capability to capture increasingly high-frequency food textures.

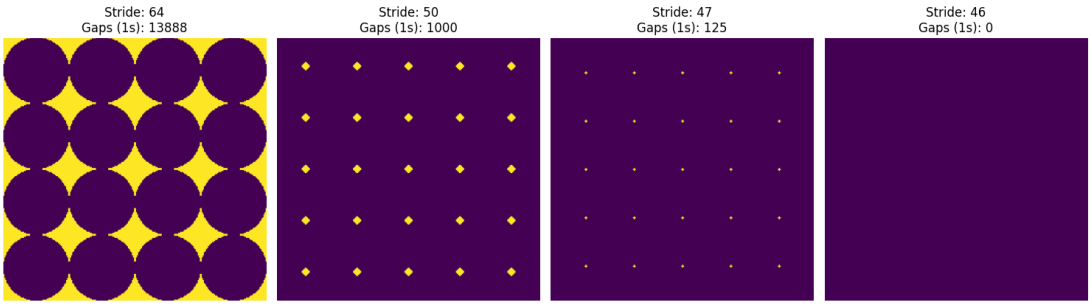


Figure 7.8: Empirical analysis of circular patch coverage. The visualization demonstrates the spatial gaps (yellow) resulting from tiling circular patches of 64 pixels at different stride values (S). Full coverage is achieved when the stride ensures sufficient overlap to cover the interstitial regions $(S = 46)$.

Figures 7.9 and 7.10 illustrate the resulting feature maps for Zernike Moments and Zernike Wavelets, respectively.

For Zernike Moments (Fig. 7.9), the grid displays the responses for the first 32 polynomials (Z_0 to Z_{31}). While Z_0 captures the low-frequency approximation (average intensity), higher indices encode progressively finer spatial frequencies and distinct rotational symmetries. In stark contrast to the limited directional sensitivity of Haar wavelets (restricted to horizontal, vertical, and diagonal), these maps demonstrate a significantly richer diversity of textural and edge orientations, validating the method's capability to encode complex food surface details.

Similarly, the Zernike Wavelet decomposition (Fig. 7.10) follows the multiresolution structure $V_N = V_0 \oplus W_0 \oplus \cdots \oplus W_{N/2}$. The first map (Z_0) represents the coarse approximation on V_0, while subsequent maps display the detail coefficients generated by the wavelets ψ at increasing scales. This hierarchical structure captures specific frequency bands while maintaining the rich edge diversity observed in the moment-based approach.

Remark 7.3. The reconstruction process for overlapping sliding windows involved applying the inverse projection on each patch and subsequently averaging the pixel values in the overlapping regions, as in the 2D-FHT (Sec. 7.2.1). ◊

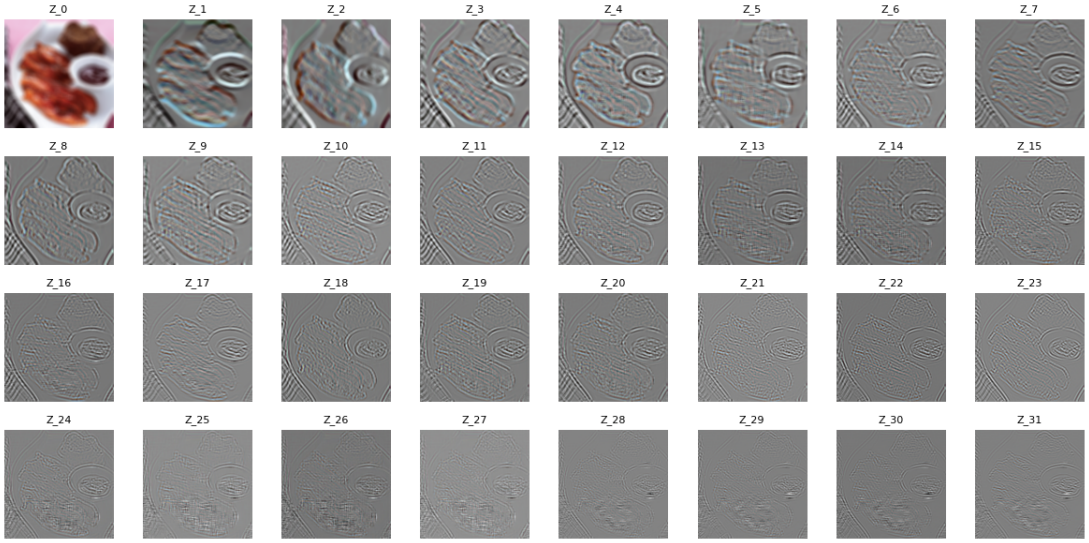


Figure 7.9: Visualization of Zernike Moment feature maps. The grid displays the projected feature responses for the first 32 Zernike polynomials (Z_0 to Z_{31}) ordered by ANSI index, highlighting the diversity of rotational symmetries captured by the basis.

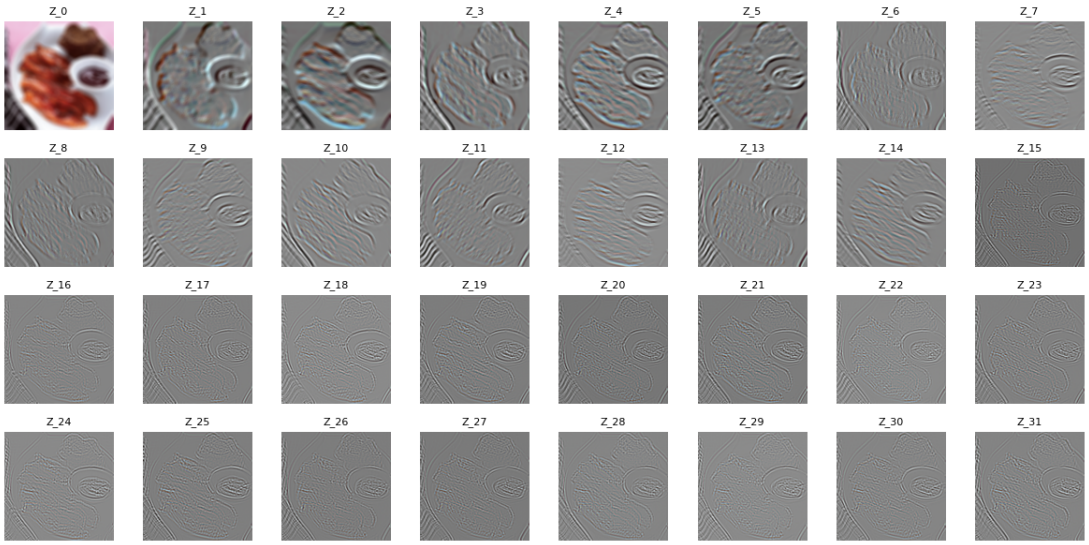


Figure 7.10: Visualization of Zernike Wavelet feature maps. The decomposition illustrates the separation between the coarse approximation (Z_0) and the detail coefficients (Z_k) across multiple scales. **Note:** Labels Z_k denote the sequential index of the wavelet basis function.

7.3 Integration as Differentiable Neural Layers

Based on the preliminary analysis, we implemented the feature extractors as differentiable `nn.Module` layers within the FoodSeg103 repository. Since the outputs of these layers feed directly into the backbone, they must maintain image-like dimensions. This necessitates unfolding the patches and stitching them back together. To achieve this, we compute the mean of the overlapping regions.

7.3.1 2D-FHT Details

As previously mentioned, we implemented the patch convolution approach, which required padding the input images, i.e., adding margins. Recall that the size of the Mallat decomposition matches the size of the image to which it is applied. Consequently, the Mallat decomposition inherits the dimensions of the padded image. This is suboptimal, as minimizing spatial dimensions is crucial for improving time complexity. While this was not an issue in the reconstruction problem, in this approach, the coefficients serve as the layer's output. One might initially consider trimming the edges of the Mallat decomposition; however, this would result in the loss of critical information, as the padding artifacts get redistributed on the lateral image patches (see Fig. 7.11).

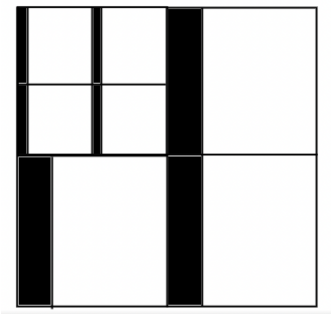


Figure 7.11: Redistribution of padding artifacts (black) on a 2-scale Mallat decomposition within a left-edge image patch. The visualization demonstrates how the zero-padding regions (necessitated by the patch's boundary position) propagate through the multiresolution hierarchy. As the spatial resolution decreases at each scale, the non-informative "void" area is downsampled, occupying a proportional section of the approximation space while introducing sharp discontinuity artifacts at the interface between valid pixels and the padding.

We also identified a flaw in the patch stitching and mean calculation processes: the original method incorrectly mixed different scales. Therefore, we developed a solution to address both the redistribution of padding artifacts and the stitching inconsistencies.

Instead of arranging the 2D-FHT outputs into a Mallat mosaic, we upsampled them to the original patch size and stacked them channel-wise. We excluded the intermediate approximation coefficients (LL), as they are required in their native resolution for the computation of subsequent scales. Consequently, since each RGB channel is processed independently, we obtain the LH, HL, and HH components for every scale, along with the final LL component. This ensures that during the stitching process, patches are assembled coherently, matching scale and filter type. Furthermore, upsampling guarantees that padding artifacts remain confined to the boundary zones, rendering simple edge trimming a valid and effective strategy.

Although the fully convolutional nature of the baseline model theoretically supports an arbitrary number of input channels, the employed ResNet backbone is pretrained to expect standard 3-channel RGB inputs. To ensure compatibility and restore the original dimensionality, we applied a final convolution layer to project the resulting high-dimensional feature map with $C_{in} \times (3 \times S + 1)$ channels, where S is the number of scales, down to 3 channels.

7.3.2 Zernike Projection and DZT Details

In contrast to the Haar approach, the feature extraction process here follows the projection method described in Sec. 4.3. As qualitatively validated in the preliminary analysis in Figures 7.9 and 7.10, the coefficient extraction is implemented to produce spatially consistent tensors.

Specifically, recalling that the coefficients for each patch form a vector $c_{\text{patch}} = \mathcal{V}_N^\dagger f_{\text{patch}}$, the implementation stacks these values across the channel dimension. Thus, the rearrangement creates a high-dimensional feature map of shape $(J, H/S, W/S)$, where $J = (N+1)(N+2)/2$ and S corresponds to the patch extraction stride. Consequently, the distinct spatial feature maps previously visualized in Figs. 7.9 and 7.10 are effectively stacked along this depth axis, forming a single dense descriptor tensor.

Thus, similar to the FHT layer, we included a final convolutional layer to reduce the output channels to 3, ensuring compatibility with the backbone.

This approach initially triggered a warning due to the mismatch between the spatial dimensions of the output feature map and the original image. While this reduction in resolution offered the potential for a faster forward pass compared to the baseline, it also presented a significant risk to accuracy, as the extensive upsampling required to align predictions with the ground truth mask could introduce interpolation errors. Furthermore, we observed that higher stride values paradoxically increased execution times, as the computational overhead of massive upsampling outweighed the benefits of processing smaller feature maps.

Consequently, the chosen strategy involved using a patch stride of 4 pixels in the spectral layers, while simultaneously modifying the ResNet backbone to use a stride of 1 instead of the standard 2 in stage 2 (recall Tab. 6.1). This configuration resulted in a total downsampling factor of 16 (relative to the original input) instead of 8, providing an optimal balance between faster computation times and high segmentation accuracy.

Matrix Multiplication vs. Convolutional Implementation

Initially, for the two proposed layers, coefficient extraction was conceptualized as a matrix multiplication applied to each image patch. However, explicitly extracting these patches (an operation implemented via `PyTorch unfold`) resulted in significant data redundancy and prohibitive VRAM consumption. To address this bottleneck, we reformulated the operation by reshaping the basis matrices into convolutional kernels. This approach eliminates the need to explicitly extract patches from the original image, allowing us to perform coefficient extraction efficiently via a standard convolution operation, drastically reducing memory consumption. This particular improvement was key to resolving OOM exceptions during the stride reduction and the Multi-Level architecture execution.

7.4 Proposed Architectures

We integrated these layers into the FoodSeg103 `mmsegmentation` pipeline, placing them before the ResNet50 backbone. We explored three progressive architectural designs.

7.4.1 Direct Spectral Injection

In the simplest configuration, as illustrated in Fig. 7.12, the RGB image is passed through the spectral layer extracting (HW, ZM or ZW). The resulting coefficients are rearranged as previously mentioned and projected via a 1×1 convolution to match the input channels expected by ResNet50, namely 3. This setup forces the network to rely solely on spectral features.

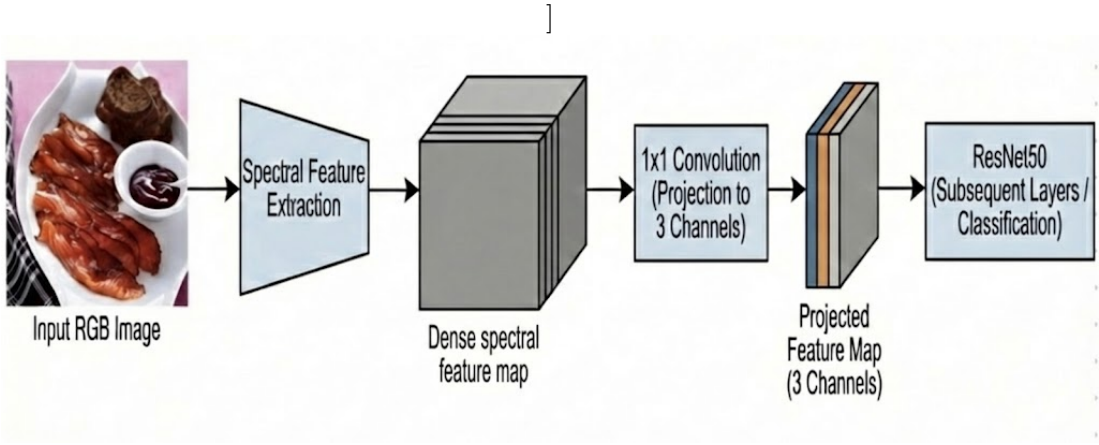


Figure 7.12: Schematic of the Direct Spectral Injection Architecture. The pipeline first calculates the spectral feature extraction and then reduces their channel dimensionality to 3.

7.4.2 RGB Fusion and SE-Block Refinement

Experiments showed that relying solely on spectral coefficients occasionally led to lower accuracies. To mitigate this, we designed a *Fusion Module*, illustrated in Fig. 7.13:

1. **Spectral Path:** Extraction of the high-dimensional feature-map via the proposed spectral layer.

2. **Concatenation:** The spectral coefficients are concatenated with the (downsampled if needed) RGB image along the channel dimension.

3. **Squeeze-and-Excitation (SE-Block):** Based on [23], we integrated an SE-Block post-concatenation. As detailed in the diagram, this mechanism explicitly models channel interdependencies by "squeezing" global spatial information into a channel descriptor and then "exciting" it to generate importance weights. This allows the network to dynamically emphasize informative spectral textures over generic RGB features (or vice versa) before the final 1×1 channel reduction.

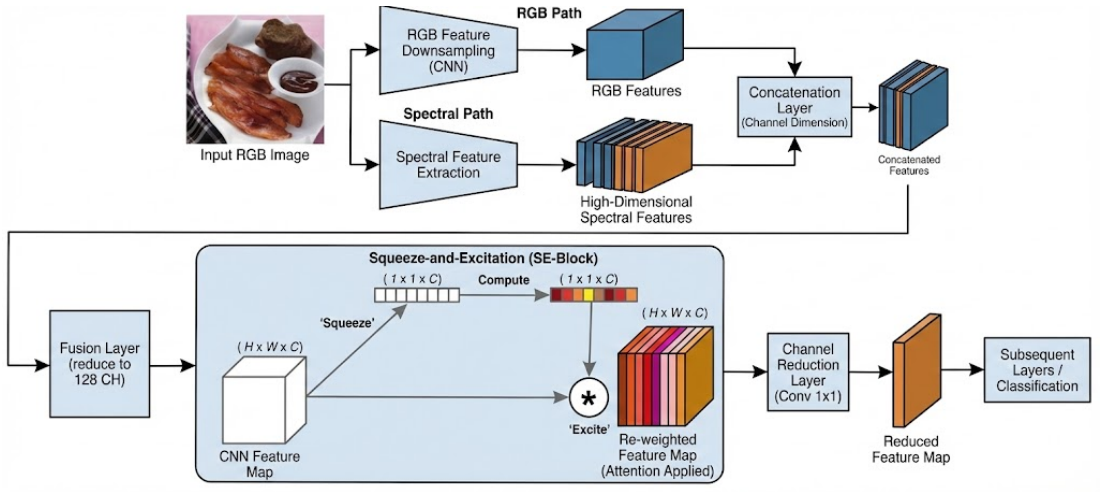


Figure 7.13: Schematic of the RGB Fusion and SE-Refinement Module. The pipeline fuses parallel RGB and Spectral feature branches via concatenation. The subsequent Squeeze-and-Excitation block dynamically recalibrates channel weights to prioritize relevant modalities before the final dimensionality reduction.

7.4.3 Multi-Level Spectral Pyramid

Given the sensitivity of the spectral layers to the chosen patch size, in the final stage of the project, we introduced a multi-scale enhancement technique for the best-performing spectral layers (ZM and ZW). We denote this configuration as the *Multi-Level Spectral Architecture* (see Figure 7.14).

Instead of relying on a single patch size, we extract features at multiple scales (e.g., patch sizes of 32, 64, 128, etc.) and concatenate them. While this approach implies a significant computational and memory overhead, the transition from explicit matrix multiplication to the convolutional implementation effectively mitigated the memory constraints.

Ideally, maximizing both the patch size and the radial degree N would allow for the simultaneous extraction of fine details and global structures. However, combined high values for both parameters present prohibitive computational costs. Thus, the Multi-Level approach allows us to specialize the Zernike order N based on the theoretical role of each scale:

- **Large Patches (Scale 128 – Context):** We employ lower N values. As these patches cover large portions of the image, their primary goal is to capture global context, illumination, and general shape (low frequencies), which require fewer coefficients.
- **Small Patches (Scale 32 – Detail):** We employ higher N values. These patches focus on local regions where high-frequency details (textures, sharp boundaries) are located, necessitating a higher radial degree to be accurately resolved.

Remark 7.4. Theoretically, extremely high N values are desired for the smallest patches to maximize detail retention. However, care must be taken: first, the set of sampling points (pixels) must always exceed the number of coefficients $J = (N + 1)(N + 2)/2$ to ensure the system is overdetermined. Second, as observed in our experiments, high N values drastically increase VRAM consumption, forcing a compromise between the theoretical ideal and hardware limitations. ◊

To address the memory constraints, avoiding severe CUDA out-of-memory (OOM) exceptions caused by the direct concatenation of high-dimensional feature maps, we introduce a crucial architectural component. Each branch incorporates a dedicated *Trainable Adapter Tower*, a stack of convolutional blocks, that drastically compresses the channel dimensions *before* the final concatenation stage. Specifically, we employ a series of 1×1 convolutional layers to iteratively reduce the channel dimension by a factor of 4 at each step, until reaching a compact representation (e.g., 4 channels per scale).

These compressed feature maps are then concatenated, and a final convolutional projection layer maps the aggregate tensor to the desired target dimension. In this Multi-Level configuration, experiments demonstrated that projecting to a channel count higher than 3 yielded superior segmentation performance compared to forcing a standard RGB-like output. Although this modification necessitated discarding the pre-trained weights for the first layer of the ResNet backbone (initializing it from scratch), the performance gain from the richer spectral representation justified this trade-off.

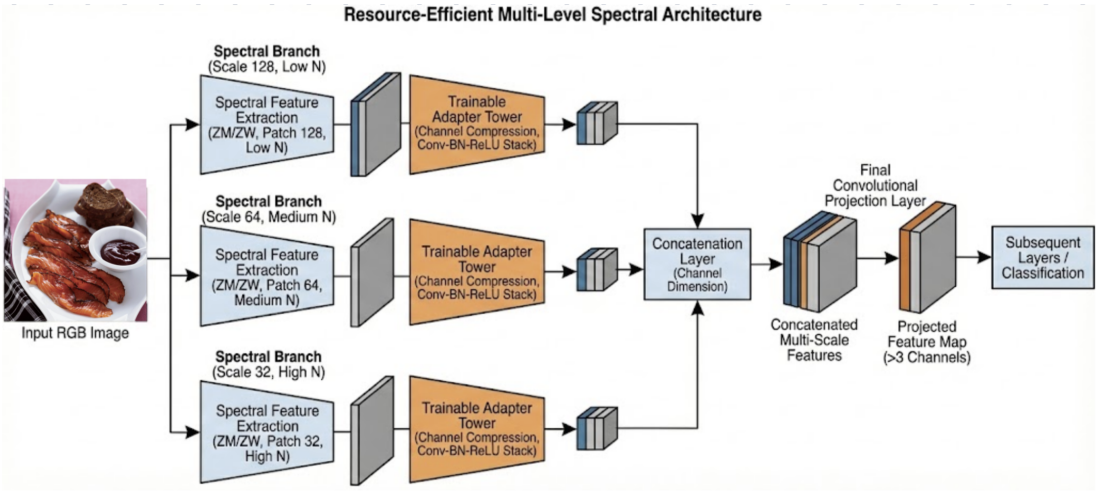


Figure 7.14: Schematic of the Multi-Level Spectral Architecture. The input image is processed in parallel by three spectral branches with decreasing patch sizes (128, 64, 32) and increasing Zernike orders (N) to capture both global context and fine texture. To make high-resolution processing feasible, each branch feeds into a Trainable Adapter Tower that compresses the feature depth before the concatenation layer. A final projection layer adapts the multi-scale features for the subsequent backbone.

Chapter 8

Experiments and results

In this chapter, we explain in detail the different experiments we conducted on each of the three spectral layers, with their respective results. We have conducted numerous experiments, each of them motivated by a hypothesis drawn from previous experimental results.

Table 8.1 summarizes all the experiments and their results. To facilitate cross-referencing with Table 8.1, metric names are italicized in their descriptions.

Remark 8.1. Some configurations were tested and resulted in *OOM* exceptions or were cancelled, as the estimated time was untenable. ◊

8.1 Experimental Setup

To ensure the reproducibility of our results and provide a rigorous comparison with the baseline, we first detail the benchmark dataset, the training configuration, and the evaluation metrics employed.

8.1.1 Benchmarks in Food Segmentation: The FoodSeg103 Dataset

Data availability constitutes a crucial bottleneck in food segmentation. Early datasets were often limited in size or focused on whole-image classification rather than pixel-level segmentation (e.g., the Food-101 dataset [7]). The release of *FoodSeg103* [50] marked a significant milestone. Containing 7,118 images annotated with 103 food categories, it is currently one of the most comprehensive benchmarks for this task. Although larger datasets exist (e.g., Food Recognition 2022 with over 40k images [33]), the FoodSeg103 dataset adapts perfectly to the scope and hardware availability of this project.

The FoodSeg103 dataset is characterized by a high diversity of food appearance and complex background configurations with their respective hand-made segmentation masks, as shown in Fig. 8.1. While the mask may appear to the human eye as a uniform grey blob against a black background, it actually encodes semantic information: the black pixels represent the background class (index 0), while the grey regions consist of specific integer values, each corresponding to a unique food category index. The challenge lies not only in the number of classes but in the semantic similarity between them (e.g., distinguishing between different types of meat or sauces). For a complete breakdown of the 103 food categories and the background class utilized in this benchmark, refer to the full listing in Table B.1.

(a) RGB input Image **(b) Semantic Segmentation Mask**

Figure 8.1: Visualization of a sample from the FoodSeg103 dataset. (a) The original RGB food image. (b) The corresponding ground truth segmentation mask. Note that the mask visualizes class indices as pixel intensities; the background is represented by black pixels (value 0), while the specific food items are represented by distinct gray values corresponding to their unique class IDs.

8.1.2 Development Environment

The official GitHub repository associated with the FoodSeg103 paper provides implementations for various baselines, including our selected architecture. However, a significant challenge arose due to software obsolescence and incompatibilities with modern hardware. To ensure reproducibility and effectively manage these complex dependencies of computer vision libraries, the entire development was conducted within a containerized environment using Docker.

Hardware and Remote Setup

All experiments were conducted on a high-performance workstation provided by the University of Barcelona (UB), equipped with an NVIDIA RTX 3090 (24GB VRAM). This high memory capacity was critical for training the memory-intensive baselines and our proposed architectures.

The system was operated remotely via Secure Shell (SSH), replicating a standard cloud-computing workflow. This setup allowed for long-running training sessions (up to several days) to be managed efficiently independent of the local machine.

Software Stack and Docker Workflow

The host software stack was built upon Ubuntu 20.04 LTS. Due to specific version incompatibilities between CUDA, PyTorch, and the `mmcv` library required by the FoodSeg103 repository, a custom Docker image was created targeting the `sm_86` architecture of the GPU. This environment standardizes the execution of:

- **PyTorch 1.10** with CUDA 11.3 support.
- **MMSegmentation 0.20.0**: An open-source semantic segmentation toolbox based on PyTorch.
- **Custom Modules**: Our implementations of Zernike and Haar transforms.

To facilitate an agile development cycle, Docker Volumes were utilized to map the local repository and dataset into the container. This configuration enabled seamless code editing and persistence on the host machine while executing the training scripts within the isolated container environment, avoiding the need to rebuild the image for every code modification.

8.1.3 Training Configuration and Data Augmentation

All the models were trained from scratch on the FoodSeg103 dataset with a fixed batch size of 4 and 160k iterations.

The train and test datasets used follow the default repository configuration. Namely, the test images are resized to 2048×1024 pixels and randomly flipped, and the train images go through the following pipeline, visualized in Fig. 8.2:

(a) Resize to 2048×1024.

(b) Random rescale from 1024×512 to 4096×2048.

(c) Extract a random window of 1024×512.

(d) Random flip.

(e) Apply random photometric distortion.

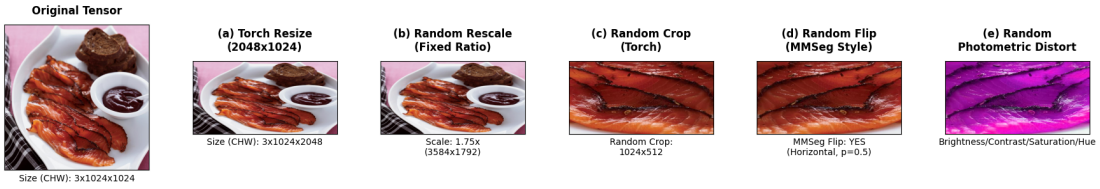


Figure 8.2: Visualization of the data augmentation pipeline for training images applied on the benchmark image Fig. 7.2. The process consists of five sequential steps: (a) the input image is resized to a fixed size of 2048×1024; (b) it undergoes a random rescale to a size between 1024×512 and 4096×2048; (c) a random 1024×512 window is extracted; (d) a random flip is applied; and (e) random photometric distortion is performed. This resulting image is then used for network training.

Unless stated otherwise, in the first two sections, the spectral layer configuration consisted of a patch size of 64×64 pixels, a stride of 4 pixels and 3 output channels.

8.1.4 Quality Metrics and Computational Complexity

To rigorously assess the proposed WaveFood architectures against the baseline, we employ a comprehensive set of metrics that evaluate both the semantic precision of the segmentation and the computational efficiency of the models.

Quality Metrics

The following metrics are used to quantify the model's ability to correctly classify pixels across the 103 food categories. We adopt the standard evaluation protocols for semantic segmentation as detailed in [44]:

- **mIoU (Mean Intersection over Union):** Widely regarded as the standard metric for semantic segmentation. It calculates the ratio of the intersection to the union of the predicted and ground truth masks, averaged across all classes. This metric is particularly critical for food segmentation as it rigorously penalizes false positives and assesses how well the model delineates amorphous boundaries. Mathematically, let $p_{ij}^{(k)} \in \{0,1\}$ denote the predicted value and $g_{ij}^{(k)} \in \{0,1\}$ the ground truth value at pixel (i,j) of a given class k. The $\text{IoU}^{(k)}$ of a given class k is calculated as follows:

$$IoU^{(k)} = \frac{\sum_{i,j}(p_{ij}^{(k)} \cdot g_{ij}^{(k)})}{\sum_{i,j}(p_{ij}^{(k)} + g_{ij}^{(k)} - p_{ij}^{(k)} \cdot g_{ij}^{(k)})}. \tag{8.1}$$

  In this formulation, the numerator represents the *intersection* (evaluating to 1 only when both pixels are active), while the denominator represents the *union* according to the inclusion-exclusion principle. Finally, the total mIoU is obtained by averaging these scores over all M classes ($mIoU = \frac{1}{M}\sum_{k=1}^{M} IoU^{(k)}$).

- **mAcc (Mean Class Accuracy):** The average pixel accuracy calculated separately for each class. This metric helps evaluate performance in the presence of class imbalance, ensuring that smaller or less frequent food items are not overshadowed by dominant background classes. Mathematically, it represents the average recall across all M classes. For a specific class k, the accuracy is the ratio of correctly predicted pixels to the total number of ground truth pixels for that class:

$$Acc^{(k)} = \frac{\sum_{i,j}(p_{ij}^{(k)} \cdot g_{ij}^{(k)})}{\sum_{i,j} g_{ij}^{(k)}}. \tag{8.2}$$

  Here, the denominator $\sum g_{ij}^{(k)}$ represents the total area of class k in the ground truth. The final mAcc is computed as $\frac{1}{M}\sum_{k=1}^{M} Acc^{(k)}$.

- **aAcc (Average Accuracy):** The ratio of correctly classified pixels to the total number of pixels in the dataset. While it provides a global view of performance, it is generally less informative than mIoU for detailed segmentation tasks as it can be biased by the background class. Using the previous notation, where the sum of intersections represents all correctly classified pixels across all classes, the formula is:

$$aAcc = \frac{\sum_{k=1}^{M}\sum_{i,j}(p_{ij}^{(k)} \cdot g_{ij}^{(k)})}{\sum_{k=1}^{M}\sum_{i,j} g_{ij}^{(k)}}.$$

  Here the denominator $\sum_{k=1}^{M}\sum_{i,j} g_{ij}^{(k)}$ represents the total classified pixels.

Remark 8.2. The mathematical formulations provided above are presented for a single image instance for clarity. In our experimental evaluation, however, these metrics are aggregated over the entire test dataset to ensure a robust assessment of the model's performance. Consequently, the pixel-wise summation $\sum_{i,j}$ is generalized to $\sum_{n=1}^{N}\sum_{i,j}$, where N denotes the test dataset size and n represents the index of each image. $\Diamond$

Computational Complexity

Given our focus on creating efficient alternatives to heavy CNN baselines, the following resource metrics are reported:

- **Params (Parameters):** The total number of parameters in the model (in millions). This indicates the model's complexity and storage requirements.
- **Size:** The physical storage size of the saved model parameters and buffers (in Megabytes).
- **Speed:** The average training time per image (in seconds). This metric is crucial for validating our hypothesis that spectral layers can accelerate convergence and reduce training costs.
- **VRAM (Video Random Access Memory):** The peak GPU memory consumption recorded during the forward and backward passes. This metric validates the feasibility of deploying the Multi-Level architectures on hardware-constrained environments.

To extract the Params, Size and Speed metrics, we have implemented two Python scripts, added to the `tools` folder in the repository.

8.2 Direct Injection Architecture

First, we evaluated the direct injection architecture on the three spectral layers.

8.2.1 Haar Wavelets

For the HW layer, we chose to start with a 3-scale decomposition, performed with a 32 pixel stride, i.e., a half patch-size overlap. The number of scales used could have theoretically reached $\log_2(512) = 5$, as each scale divides the image size by 2 on each axis and the smallest axis of training images has 512 pixels. However, the extremely low accuracy achieved combined with a noticeable reduction in training speed compared to the baseline forced us to discard more computationally-expensive settings. A similar argument prevented us from investigating higher pixel overlap.

8.2.2 Zernike Moments

The use of $N = 20$ on the ZM was motivated by the previous notebook implementation, where we saw that it achieved great SSIM values on the test image, without substantially increasing the computation overhead. Remarkably, this model improved upon the baseline on the mAcc metric, with a significant speed improvement.

We also investigated the difference with a *stride* of 2 instead of 4, as we wanted to corroborate that our selection of the 4 pixel stride on the Zernike approaches was valid; hypothetically, more overlap meant denser feature maps and better chances of higher mIoU. Curiously, while the mIoU worsened, the aAcc surpassed the baseline's. However, the drastic increase in execution time made us discard this approach.

8.2.3 Zernike Wavelets

Regarding ZW, as the reconstruction SSIM results were similar to the ZM, we decided to use $N = 32$, as we were forced to use powers of 2 and we wanted to exceed the $N = 20$ used on the ZM. The results were far better than the HW ones, but considerably worse than both ZM and the baseline.

8.2.4 Architecture Conclusions and Spectral Comparisons

We immediately observed the learning capacity of the different coefficients applied to the CNN, corroborating our primary hypothesis: the Zernike coefficients describe the food better than the Haar ones. However, contrary to the hypothesis, the Zernike Moments performed far better than the Zernike Wavelets, surpassing the baseline in some accuracy metrics, with a $\sim 30\%$ speed improvement.

We also confirmed the superiority of the 4 pixel stride, as testing a 2 pixel stride resulted in lower mIoU and mAcc alongside a huge increase in training time, which far exceeded the baseline's training time (Speed metric).

8.3 RGB Fusion and SE-Block Architecture

Given the previous results, we wanted to check if a richer architecture could increase the accuracy, without being detrimental to the Speed.

8.3.1 Haar Wavelets

First of all, we checked if this improved architecture could benefit the HW. As the Speed was critically close to the baseline's, we chose to test the new architecture *without overlap*, trying to lower the Speed metric. In the Haar spectral layer, no overlap is viable, as unlike in the disk spectral layers, it involves no data loss. Due to the lack of improvement in the accuracy, we definitively discarded the Haar Wavelets.

8.3.2 Zernike Moments

Given that Zernike Moments proved to be our best performing layer, we conducted extensive experiments in this category.

The first hypothesis was the lack of need for *data augmentation*. We modified the previously described training data pipeline by removing items 2, 4 and 5. The accuracy results were marginally lower than those of the simpler architecture and we observed clear overfitting, noting better accuracies at 144k iterations than at the end of training, as shown in Fig. B.1.

Consequently, we trained the model utilizing the full data augmentation pipeline with $N = 20$. This yielded our best performing model regarding mIoU and mAcc, surpassing the baseline. As shown in Table 8.1, both the previous model and the current one substantially decreased the VRAM metric and were almost as fast as the Direct Injection model.

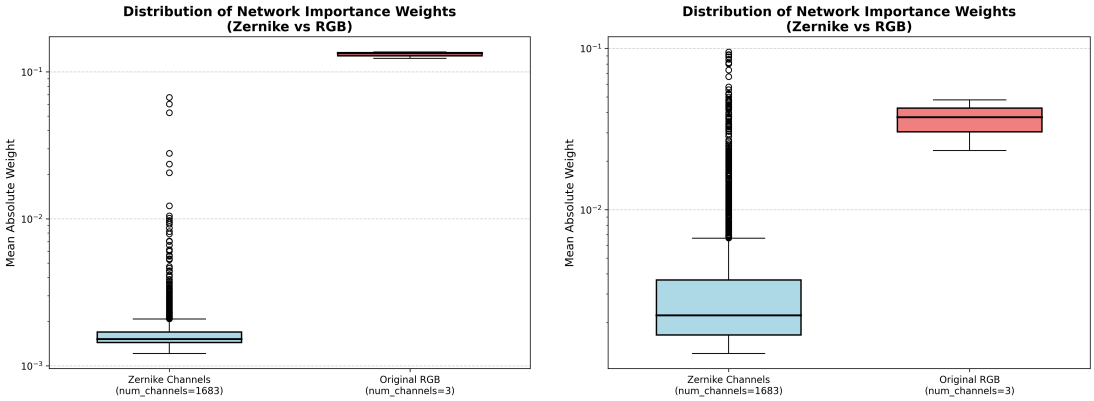
Later, we tested $N = 32$ to check if the increase in N yielded a marked increase in accuracy without significantly impacting the Speed metric. This model surpassed the baseline on all three accuracy metrics. However, this incurred a notable penalty on VRAM usage.

Using the settings of the previous best performing model, we also investigated if the number of channels passed to the backbone could be *expanded* from the default

3, to determine if the significant channel reduction had a detrimental effect. We tried increasing the final channels to 64. All metrics were worse than those of the previous model, so we discarded the hypothesis.

We implemented a Python script to generate a boxplot of the CNN weights corresponding to the partial channel reduction to 128 channels (Fig. 8.3), in order to assess how much the model was relying on the Zernike Moments versus the RGB. Unfortunately, we observed that the model was ignoring almost all the Zernike Moments.

To determine if we could improve the Zernike Moments utilization, we hypothesized that the random rescaling of the training images was forcing the model to rely only on the RGB channels. To test that hypothesis, we tried *removing the random rescale* from the training data pipeline. That resulted in a substantial drop in accuracy and a slight overfit, but the plot showed that the model relied on the ZM much more, even more than on the RGB channels.



(a) RGB Dominance: The model assigns disproportionately high weights to the 3 RGB channels (upper red box) while suppressing the Zernike moments (lower blue box, $< 10^{-2}$). This disparity suggests the network is ignoring geometric shape information in favour of colour cues.

(b) Balanced Feature Integration: In this configuration, the distribution of Zernike weights shifts upwards significantly, becoming comparable in magnitude to the RGB weights. This indicates a successful incorporation of spectral geometric features into the learning process.

Figure 8.3: Log-scale Analysis of Feature Importance ($N = 32$ Fusion). Comparative visualization of mean absolute weights for Zernike (1683 channels) vs. RGB (3 channels). (a) With Random Resizing: The model relies almost exclusively on RGB cues, suppressing geometric information. (b) Without Random Resizing: The model demonstrates robust reliance on both shape (Zernike) and colour. This reveals that resizing augmentation hinders the effective integration of geometric features.

As the accuracy of the previous experiment was not satisfactory, we tested removing the RGB information. Relying solely on the *SE-Block*, the mIoU and mAcc were not satisfactory, even when compared to the previous architecture. This finding led us to discard the SE-Block Attention layer as a key component of the given architecture.

8.3.3 Zernike Wavelets

Using this architecture, we aimed to assess if the ZW performance could increase to match that of the moments or at least the baseline.

We tested configurations with $N = 16$ and $N = 32$, as lowering to $N = 8$ would likely lead to underfitting, and increasing to $N = 64$ resulted in an OOM exception.

Notably, we observed a significant increase with respect to the simple architecture, falling just below the baseline's performance. As suspected, the accuracies with $N = 32$ were slightly better, surpassing the baseline's mAcc.

We investigated if the wavelet frequency focus was adversely affected by the photometric distortion, as varying the image colours affects each RGB channel differently, creating varying degrees of intra-channel discrepancies. Thus, our hypothesis was that the detail focus of the wavelets on each channel would respond inconsistently to each random photometric change, making the learning process difficult. Consequently, we *removed the photometric distortion* from the training data pipeline. To mitigate the overfitting observed when removing data augmentation, we introduced *Dropout* after the concatenation step and before the SE-Block attention layer. The results were the best ones for ZW so far on mIoU and mAcc, without presenting overfit. However, we ultimately discarded the use of Dropout for future experiments. While effective at reducing overfitting, randomly zeroing out neuronal activations indiscriminately suppresses potentially critical spectral features, effectively capping the representational capacity of the network. We concluded that the model should ideally learn to be robust through better feature extraction mechanisms (like the Multi-Level approach) rather than through aggressive regularization that sacrifices information.

8.3.4 Architecture Conclusions and Spectral Comparisons

With this architecture, we definitively discarded the Haar Wavelets due to their poor performance. We improved the Zernike Wavelets accuracies to approximately match the baseline's, with a much better Speed metric. We also corroborated the superiority of the Zernike Moments, as they surpass the accuracy of the baseline by **0.87**%, **1.74**% and **0.14**% in mIoU, mAcc and aAcc correspondingly, while improving the Speed by $\sim$ **22.35**%.

Unfortunately, as previously noted, these models tended to ignore almost all the spectral coefficients. Even the SE-Block did not yield any improvement, as the gains were primarily driven by the RGB fusion. This highlighted the necessity of a new improved architecture relying only on the spectral data, aiming to improve upon these results, or at least maintain them.

8.4 Multi-Level Architecture

As described in Ch. 7, we designed this architecture to mitigate the dependency on the patch size used in the convolution of the spectral extraction layers. This approach was only implemented to focus solely on spectral data using our best performing layers, namely the Zernike Moments and Wavelets.

As explained in Sec. 7.4.3, we were forced to use a minimum patch size of 32×32. This was due to the need for an overdetermined Zernike Basis and the interest in decreasing N-values to prevent OOM exceptions.

Unless stated otherwise in this section, each level was progressively reduced to 4 channels before concatenation and the spectral layer had 8 output channels.

8.4.1 Zernike Moments

We first chose to use the Multi-Level approach with the following configuration: 32×32 patches with $N = 20$, 64×64 with $N = 12$ and 128×128 with $N = 8$. We successfully surpassed the aAcc metric of our best performing model. However, the

mIoU and the mAcc did not improve. Furthermore, the mAcc were worse than that of both the RGB Fusion and the Direct Injection approaches. The Speed metric was also slightly worse, but that was expected given the notable computation overhead of this per-level repetitive strategy.

We tried to improve the mIoU by using higher N-values. Therefore, we used $N = 32, 20, 12$ on the corresponding levels. This resulted in a significant increase in the mIoU and the mAcc, but also a marked decrease in the aAcc. Nevertheless, the speed worsened, exactly matching that of the baseline.

Given that levels with higher N-values generated more channels, we hypothesized that we should not equally reduce all the levels. Thus, we conducted an experiment reducing each level to $\lfloor N/2 \rfloor$, thereby maintaining more final channels for levels with higher N-values, denoted as *high-focus*. To check the validity of this approach, we determined that reducing the final output channels to 8 would negatively affect this experiment, as now we ended with $16 + 10 + 6 = 32$ instead of $4 + 4 + 4 = 12$ channels before the final channel reduction. Therefore, we decided to increase the final channel number to 16. Recalling the experiment of the previous architecture where we augmented the output channels, we hypothesized that 64 had been excessive, whereas 16 could be ideal. Unfortunately, this experiment was unsuccessful.

We then hypothesized that focusing the final channel representation on the smallest patch (and therefore highest N-value) could introduce significant noise. Thus, we changed the levels channel reduction to 4, 8 and 4 respectively, creating a *mid-focus* approach. Consequently, as we wanted to reduce the hypothesized noise and improve the speed, we slightly lowered the N-values to $N = 26, 18, 10$. This was by far the worst performing model, so the mid-focus strategy was clearly discarded.

As the high- and mid-focus approaches had negative results and given that low-focus was theoretically unsound (as lower N-values generate fewer channels), we concluded that the default *equal-focus* was the best approach. Since it was already tested, we changed two factors. First, we increased the reduced channels per-level from 4 to 6, since 8 produced an OOM exception. Then, we tried to combine the general pixel accuracy from the first experiment (with the best aAcc) and the detail level of the second experiment (with the best mIoU result). Given that smaller patches with higher N-values are responsible for fine detail (mIoU), and the bigger patches with lower N-values are responsible for general pixel accuracy (aAcc), we chose the following N-values: $N = 32$, 16 and 8. Unfortunately, although this model theoretically seemed superior, it performed worse than either the first or second models.

Finally, as we had been experimenting with the number of channels, we tried to *compress* the final channels to only 3. This was meant to take advantage of the pre-trained first layer of the backbone, as it expected the original 3 RGB channels. The results were unsatisfactory.

We would like to mention that the three last models experienced a very notable drop in the VRAM of $\sim$ **23**%. We couldn't determine its cause, as the models are similar to the previous Multi-Level models with the expected VRAM increase due to the increased model complexity. We could only attribute this property to the chosen N-values, which seemingly significantly impacted the GPU memory management.

8.4.2 Zernike Wavelets

We conducted the initial experiment using the default settings. Therefore, the N-values, restricted to powers of 2, were selected as $N = 32$, 16 and 8. Although the accuracy was slightly worse than in the previous architecture, it represented a suc-

cessful finding, as we approached the baseline's accuracy relying solely on the spectral coefficients, with a notable drop in VRAM usage of $\sim$ **23**%.

To test the Multi-Level accuracy potential, we set aside the baseline's speed limitation. We tried adding levels of 256×256 and 512×512 patch sizes. Bigger patches would exceed the training image size, so they were automatically discarded. Even with $N = 1$ on the 512×512 level, the estimated time reached the 3 day mark, more than doubling the baseline's total time, forcing us to discard it. We then only added the 256×256 patch size level with $N = 4$. Although the Speed metric was worse than the baseline's, as expected, we achieved peak Zernike Wavelets performance on all accuracy metrics, almost matching the baseline's mIoU and aAcc while surpassing its mAcc. Furthermore, we demonstrated that adding levels did not significantly increase VRAM usage.

The final experiment was motivated to experimentally test the validity of decreasing the N-values as the patch sizes grow. Thus, we chose the following N-values: 32, 32 and 8. The results were almost identical to the first experiment (changing the mid-level frequency from $N = 16$ to 32), while substantially worsening all other metrics.

8.4.3 Architecture Observations and Spectral Comparisons

The presented Zernike Wavelets experiments demonstrate that we can approximately match the baseline's accuracy solely on spectral coefficients, confirming the hypothesis of the strong scale dependency of the chosen patch sizes.

We have also further demonstrated the superiority of the Zernike Moments, which by relying solely on the spectral coefficients, achieve top performance, surpassing the baseline across key accuracy metrics.

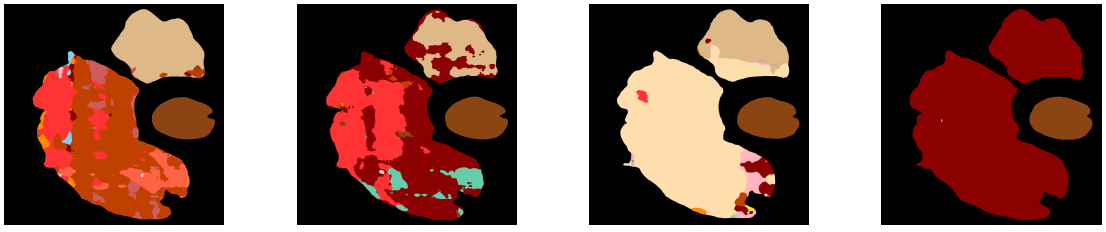
As stated previously, the implementation of this architecture tends to significantly decrease VRAM usage by $\sim$ **23**%, constituting a highly desirable upgrade, while matching the baseline's accuracy and speed.

8.5 Results Plot and Table

Tab. 8.1 shows all the experimental results, while Fig. 8.4 shows the segmentation result of the best-performing models of each spectral layer on the benchmark image Fig. 7.2. Further details of the best-performing model can be found in Tab. B.1

Table 8.1: Comparative results of the proposed WaveFood architectures against the baseline on the FoodSeg103 dataset. Each spectral strategy is separated and ordered following the experimental order. Metrics marked with ↑ indicate higher is better, while ↓ indicates lower is better. The top three results for each metric are highlighted in Best, Second and Third. Abbreviations: **M**: Method (HW: Haar, ZM: Zernike Moments, ZW: Zernike Wavelets), **RGB**: RGB Fusion, **SE**: SE-Attention Block.

M	N / Config	RGB	SE	Details	↑mIoU	↑mAcc	↑aAcc	↓Params (M)	↓Size (MB)	↓Speed (s/img)	↓VRAM (GB)
CCNet		✓			0.3269	0.4244	0.7926	49.892	190.54	0.1754	16.33
HW	Scales=3			Overlap	0.1851	0.2692	0.6783	49.892	190.54	0.1835	16.81
HW	Scales=3	✓	✓	NoOverlap	0.1752	0.2310	0.6668	49.892	190.54	0.1820	16.98
ZM	N=20			-	0.3233	0.4379	0.7875	49.894	196.26	0.1236	18.57
ZM	N=20			Stride=2	0.3224	0.4204	0.7936	49.981	194.49	0.2233	16.70
ZM	N=20	✓	✓	No Aug	0.3218	0.4310	0.7824	49.983	196.58	0.1248	12.59
ZM	N=20	✓	✓	-	0.3303	0.4430	0.7873	49.983	196.58	0.1250	12.59
ZM	N=32	✓	✓	-	0.3356	0.4418	0.7940	50.110	205.19	0.1362	19.27
ZM	N=32	✓	✓	Expanded	0.3275	0.4343	0.7930	50.136	205.29	0.1368	19.30
ZM	N=32	✓	✓	No Rescale	0.3173	0.4160	0.7872	50.110	205.19	0.1373	19.27
ZM	N=32		✓	-	0.3189	0.4257	0.7911	50.110	205.19	0.1354	19.27
ZM	N=20,12,8			-	0.3246	0.4299	0.7944	50.046	196.27	0.1460	19.14
ZM	N=32,20,12			-	0.3313	0.4369	0.7890	50.796	205.49	0.1754	20.56
ZM	N=32,20,12			HighFocus+Exp	0.3110	0.4224	0.7869	50.799	205.50	0.1760	20.57
ZM	N=26,18,10			MidFocus	0.3043	0.4038	0.7858	50.333	200.80	0.1662	13.97
ZM	N=32,16,8			EqualFocus	0.3213	0.4290	0.7869	50.709	201.06	0.1620	12.53
ZM	N=32,16,8			Compressed	0.3122	0.4189	0.7801	50.707	201.05	0.1634	12.52
ZW	N=32			-	0.2419	0.3339	0.7524	50.108	198.28	0.1345	19.20
ZW	N=16	✓	✓	-	0.3176	0.4240	0.7897	49.953	194.55	0.1254	18.64
ZW	N=32	✓	✓	-	0.3217	0.4263	0.7899	50.110	205.19	0.1397	19.27
ZW	N=32	✓	✓	Drop+NoPhoto	0.3257	0.4304	0.7868	50.110	205.19	0.1367	19.35
ZW	N=32,16,8			-	0.3122	0.4195	0.7895	50.700	201.08	0.1624	12.52
ZW	N=32,16,8,4			-	0.3263	0.4346	0.7910	50.710	204.89	0.2056	12.56
ZW	N=32,32,8			-	0.3123	0.4182	0.7897	51.408	210.12	0.1970	15.61



(a) Baseline **(b) Haar Wavelets (RGB)** **(c) Zernike Moments (N=32 & RGB)** **(d) Zernike Wavelets (4 levels Multi-Level)**

Figure 8.4: Visual comparison of segmentation results across the different spectral layers, with their best performing architecture and configuration. The Baseline and Haar models show coarser segmentation, while Zernike-based methods (c, d) achieve better definition of object boundaries and smaller details. A different colour has been applied to each class for our visualization.

Chapter 9

Limitations, Conclusions and Future Directions

This final chapter highlights the limitations encountered during the development and experimentation phases, summarizes the main findings of our research, and suggests potential avenues for future investigation to further improve spectral-based semantic segmentation.

9.1 Limitations

Despite the promising results, our approach faced several limitations that hindered the models from surpassing the baseline by a larger margin:

- **Haar Wavelets Incompatibility:** We found that Haar Wavelets are largely unsuitable for food segmentation. Their blocky, pixel-aligned nature didn't suit the dataset's needs, as the two tested architectures showed clear underfitting (see Tab. 8.1).

- **RGB Dominance and Shortcut Learning:** The most critical limitation observed was the "shortcut learning" phenomenon in the RGB Fusion architecture (see Fig. 8.3. When given access to both raw RGB data and spectral coefficients, the CNN overwhelmingly learnt to prioritize the RGB channels, treating the spectral data as noise. We could only force the model to look at the Zernike coefficients by degrading the RGB pipeline (e.g., removing random resizing), which were not an ideal solution.

- **Patch Size Dependency:** The spectral extraction is inherently tied to the chosen patch size. While our Multi-Level architecture attempted to mitigate this by combining features from different sized windows, the fixed nature of these windows still lacks the flexibility of standard dilated convolutions or improvements used in modern segmentation backbones (more information in [49]).

- **Hardware Constraints:** Due to memory limitations (VRAM), we were unable to test higher orders of N (e.g., $N = 64$ or higher), which might have yielded better fine-grained details; larger batch sizes, which could have resolved the gradient optimization problem smoother; and larger patch-sizes, which could have hold better general image context for better semantic classification.

- **Time Constraints:** Although we improved the training speed by 22% (see Tab. 8.1), each model's training lasted a minimum of a day, imposing a restrictive time limit on experimentation, due to the semester length and the previously needed theoretical background study.

- **Data Augmentation Variance:** The proven mandatory use of data augmentation created performance fluctuations in each model due to the randomness of the sampled training data. To mitigate this, common practice dictates repeating each training session several times to extract mean metrics. However, due to time constraints, we were limited to training each model only once, potentially introducing statistical noise into our comparative conclusions.

9.2 Conclusions

The primary objective of this thesis was to determine if the novel WaveFood spectral framework, specifically using Haar Wavelets, Zernike Moments and Zernike Wavelets, could serve as a viable, efficient alternative or complement to standard RGB features in semantic segmentation tasks using the FoodSeg103 dataset.

Based on the extensive experimental results presented in the previous chapter, we can draw the following conclusions:

- **Superiority of the Zernike approaches:** We have definitively demonstrated that Zernike Moments and Wavelets are significantly more effective than Haar Wavelets for feature extraction in this context for any of the presented architectures. The rotational invariance and orthogonal properties of Zernike polynomials capture the organic structures of food much better than the rigid, axis-aligned nature of Haar Wavelets, as we had hypothesized.

- **Zernike Wavelets Viability:** We proved that a "Direct Injection" approach was inefficient for the Zernike Wavelets. However, more complex architectures, like the RGB Fusion with SE-Block and the Multi-Level improved their performance, achieving performance comparable to the baseline. Although the first one tended to bypass the spectral information, the second one provided the stability needed to achieve good performance even relying solely on spectral information.

- **Zernike Moments vs Wavelets:** Theoretically, the spatial-frequency localization of the ZW should outperform the basic orthonormal projection of the ZM. However, experimentally we have demonstrated that the robust stability of using orthonormal bases yields a performance improvement in every architecture.

- **Surpassing the Baseline:** Beyond our primary objective of establishing spectral layers as a viable and efficient alternative, our experiments revealed that Zernike-based architectures can outperform the CCNet baseline in both accuracy and speed. Specifically, the ZM (N=32) with RGB fusion model exceeded the baseline's accuracy by **0.87**%, **1.74**% and **0.14**% in mIoU, mAcc and aAcc correspondingly, while improving the training speed by $\sim$ **22.35**%.

- **Resource Efficiency in Multi-Level Architectures:** While the Multi-Level Zernike Moments approach maintained accuracy and training speed comparable to the baseline, it demonstrated a substantial and highly desirable reduction in VRAM usage of $\sim$ **23**%. This validates the architecture as a superior alternative

for hardware-constrained environments, offering the same performance at a lower memory cost.

- **Dependence on Convolution and Data Augmentation:** Theoretically, the rotation and scale invariance of Zernike approaches suggested a reduced need for data augmentation. However, experiments demonstrated that removing augmentation caused overfitting. We conclude that the sliding-window convolution mechanism used for feature extraction re-introduces a strong axis and rotation dependence, limiting the global invariance of the spectral basis. Notably, however, the Multi-Level architecture successfully mitigated the scale dependence, confirming that architectural adjustments can compensate for these convolutional constraints.

In summary, we have successfully validated that spectral layers can replace standard convolutional blocks to achieve superior results with a fraction of the computational and memory costs, provided the correct layer (Zernike Moments) is chosen.

9.3 Future Work

To address the previously mentioned limitations and unlock the full potential of spectral layers in deep learning, we propose the following directions for future research:

- **Improved Fusion Mechanisms:** To mitigate the RGB dominance issue, future work should explore more sophisticated fusion techniques beyond simple concatenation or the specific SE-Block [23] configuration used in this study. Mechanisms such as *Cross-Attention* [45] could enable the model to explicitly learn dependencies between the spectral and spatial modalities. Alternatively, advanced *Gated Fusion* strategies, for instance, applying SE-Blocks directly after concatenation rather than after the initial channel reduction, could dynamically weigh the importance of spectral versus spatial features, encouraging the network to leverage the texture information provided by Zernike Moments.

- **Network Pruning and The Lottery Ticket Hypothesis:** Our work demonstrates that spectral layers can significantly reduce model complexity while maintaining performance. To push efficiency further, future research could explore *Network Pruning* techniques inspired by the Lottery Ticket Hypothesis [18]. Investigating whether our manual selection of Zernike orders corresponds to "winning tickets", sparse subnetworks capable of matching dense performance, could lead to automated methods for selecting the optimal spectral coefficients, eliminating the need for manual hyperparameter tuning of N-values.

- **Optimized Coefficient Ordering and Patch Traversal:** Future work will explore Zig-Zag coefficient ordering [46] to prioritize high-energy components for more efficient feature representation. Additionally, replacing raster scan with space-filling curves such as the Hilbert curve [34] could better preserve spatial locality and improve memory efficiency when processing large high-resolution samples.

Acknowledgments

This work was partially funded by the EU project MUSAE (No. 01070421), 2021-SGR-01094 (AGAUR), ICREA Academia'2022 (Generalitat de Catalunya), Robo STEAM (2022-1-BG01-KA220-VET000089434, Erasmus+ EU), Deep-Sense (ACE053/22/000029, ACCIÓ), and Grants PID2022141566NB-I00 (IDEATE), PDC2022-133642-I00 (Deep-FoodVol), and CNS2022-135480 (A-BMC) funded by MICIU/AEI/10.13039/501100011033, by FEDER (UE), and by European Union NextGenerationEU/ PRTR. A. AlMughrabi acknowledges the support of FPI Becas, MICINN, Spain. The authors thankfully acknowledge the RES resources provided by the Barcelona Supercomputing Center on MareNostrum5 for IM-2025-3-0008.

Part III

Appendices

Appendix A

Mathematical background

In this appendix, we share some general definitions and propositions that are core to the results shown in this thesis. All the missing proofs can be found at [8] and [41].

A.1 Pre-Hilbert and Hilbert Spaces

Definition A.1. *A vector space over the complex field is called a* pre-Hilbert space *if there exists a map* $\langle \cdot, \cdot \rangle : H \to \mathbb{C}$ *named as* inner product*, so that the following rules hold:*

- $\langle x, y \rangle = \overline{\langle y, x \rangle}$ *for every* $x, y \in H$ *(the overline denotes complex conjugation).*
- $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle$ *for every* $x, y, z \in H$*.*
- $\langle \alpha x, z \rangle = \alpha \langle x, z \rangle$ *for every* $x, y \in H$ *and* $\alpha \in \mathbb{C}$*.*
- $\langle x, x \rangle \geq 0$ *for every* $x \in H$*.*
- $\langle x, x \rangle = 0$ *if and only if* $x = 0$*.*

$\Diamond$

Proposition A.2. *Given a pre-Hilbert space, we can define its norm from its inner product as such:*

$$\begin{aligned} \|\cdot\| : H &\to [0, +\infty] \\ x &\mapsto \langle x, x \rangle^{1/2}. \end{aligned}$$

$\Diamond$

If this space is also *complete*, i.e., every Cauchy sequence converges in H, then we call it a *Hilbert space*. Thus, each Hilbert space is also a pre-Hilbert space.

Thus, a Hilbert space is also a Banach space (normed and complete). It is also a metric space, as we can also create a metric for the given space as follows:

$$\begin{aligned} d : H \times H &\to [0, +\infty] \\ x, y &\mapsto \|x - y\|. \end{aligned}$$

Proposition A.3 (*The Cauchy-Schwarz inequality*)**.** *For any two elements x, y in a pre-Hilbert space H, the absolute value of their inner product is bounded by the product of their norms:*

$$|\langle x, y\rangle| \leq \|x\|\|y\|. \tag{A.1}$$

Equality holds if and only if x and y are linearly dependent ($x = \alpha y$ for a $\alpha \in \mathbb{C}$).$\Diamond$

Proposition A.4. *Given a fixed $y \in H$, where H is a pre-Hilbert space, the mapping*

$$\begin{aligned} \langle \cdot, y\rangle : H &\to [0, +\infty] \\ x &\mapsto \langle x, y\rangle \end{aligned}$$

is a continuous function on H.

Proof. Follows immediately from the definition of the inner product and the Cauchy-Schwarz inequality. □

Thus, the norm is also a continuous function on H.

Proposition A.5. *Given a Hilbert space H. Let F be a closed subspace of H. Then, F is also a Hilbert space.*

Proof. We know that if F is closed it is a vector space. Its inner product immediately comes from the restriction of the inner product of H onto F. The completeness comes from the completeness of H and given that F is closed, thus, having all its limit points. □

We can even add another condition to this spaces.

Definition A.6. A Hilbert space H is *separable* if it has a countable orthonormal basis. $\Diamond$

Corollary A.7. *Given a separable Hilbert space H. Let F be a closed subspace of H. Then, F is also a separable Hilbert space.* $\Diamond$

Theorem A.8 (*Finite Dimensional Subspaces are Closed*)**.** *Let H be a Hilbert space and let V be a subspace of H. If V has finite dimension, then V is a closed subspace of H.*

Proof. Since V has finite dimension $d < \infty$, it is isomorphic (algebraically and topologically) to the Euclidean space $\mathbb{K}^d$ (where $\mathbb{K}$ is $\mathbb{R}$ or $\mathbb{C}$). Since Euclidean spaces are complete (every Cauchy sequence converges within the space), it follows that V is complete with respect to the norm induced by H.

Let $\{y_n\}_{n\in\mathbb{N}} \subset V$ be a sequence such that $y_n \to y$ in H. Since the sequence converges, it is a Cauchy sequence. Because V is complete, the limit y must exist within V. Thus, V contains all its limit points and is closed. □

A.2 The Orthogonal Projection Theorem

Definition A.9. Let H be a pre-Hilbert space. We say that $x, y \in H$ are orthogonal if $\langle x, y\rangle = 0$. We abbreviate this condition as

$$x \perp y.$$

$\Diamond$

Definition A.10. Let H be a pre-Hilbert space. Given a subset $A \subseteq H$, we define the *orthogonal complement* of A as follows:

$$A^{\perp} = \{x \in H : \langle x, y\rangle = 0 \quad \forall y \in A\}.$$

$\Diamond$

Lemma A.11. *For any subset $A \subseteq H$, the set $A^{\perp}$ is a closed subspace of H.*

Proof. First, to show that $A^{\perp}$ is a subspace, let $y_1, y_2 \in A^{\perp}$ and $\alpha, \beta \in \mathbb{K}$. By the linearity of the inner product, for any $x \in A$:

$$\langle x, \alpha y_1 + \beta y_2\rangle = \bar{\alpha}\langle x, y_1\rangle + \bar{\beta}\langle x, y_2\rangle = 0.$$

Thus, $\alpha y_1 + \beta y_2 \in A^{\perp}$.

To prove that $A^{\perp}$ is closed, consider a sequence $\{y_n\}_{n\in\mathbb{N}} \subset A^{\perp}$ such that $y_n \to y$. We must show that $y \in A^{\perp}$, i.e., $\langle y, x\rangle = 0$ for all $x \in A$. Using the continuity of the inner product:

$$\langle y, x\rangle = \Big\langle \lim_{n\to\infty} y_n, x \Big\rangle = \lim_{n\to\infty} \langle y_n, x\rangle.$$

Since $y_n \in A^{\perp}$, we have $\langle y_n, x\rangle = 0$ for all n, implying that the limit is 0. Therefore, $y \in A^{\perp}$, concluding the proof. □

Definition A.12. Given a subset $A \subseteq H$ and a fixed $x \in H$. We say that a point $y \in A$ is a best approximation of x in A if

$$\|x - y\| = \inf_{a\in A} \|x - a\| = d(x, A).$$

$\Diamond$

Theorem A.13. *Given a closed subspace F of a Hilbert space H, $x \in H$ and $y \in F$. The following statements are equivalent:*

1. $\|x - y\| = d(x, F)$
2. $x - y \in F^{\perp}$.

$\Diamond$

Theorem A.14. *Let H be a Hilbert space and $F \subseteq H$ a closed subspace. Then for each $x \in H$, there exists exactly one best approximation of x in F.* $\Diamond$

Definition A.15. Given a closed subspace F of a Hilbert space H, we define the orthogonal projection map:

$$\begin{aligned} P_F : H &\to H \\ x &\mapsto P_F(x), \end{aligned}$$

which sends every element of H to its unique best approximation in F. ◊

Since $F^\perp$ is also a closed subspace (by Lem. A.11), the projection $P_{F^\perp}$ onto the orthogonal complement is also well defined. This leads to the main result:

Theorem A.16 (*The Orthogonal Projection Theorem*)**.** *Let H be a Hilbert space and let $F \subseteq H$ be a closed subspace. Then, for every $x \in H$:*

1. ***Existence and Uniqueness:*** *There exists a unique element $P_F(x) \in F$ such that*
$$\|x - P_F(x)\| = \inf_{z \in F} \|x - z\|.$$

2. ***Characterization:*** *This element is uniquely characterized by the orthogonality condition:*
$$x - P_F(x) \in F^\perp.$$

3. ***Decomposition:*** *The space H can be decomposed as the direct sum $H = F \oplus F^\perp$. That is, every $x \in H$ can be uniquely written as:*
$$x = P_F(x) + (x - P_F(x)), \quad \text{where } P_F(x) \in F \text{ and } (x - P_F(x)) \in F^\perp.$$

A.3 The L^p-spaces

In this section, we will work with measurable spaces and thus we first introduce some definitions about them.

Definition A.17. Given a measure space $(X, \mathcal{M}, \mu)$, a condition or equality is said to *hold almost everywhere* or *a.e.* if it exists a set $\mathcal{Y} \subset \mathcal{X}$ of measure 0 ($\mu(\mathcal{Y}) = 0$) which makes the condition or equality hold for every $x \in \mathcal{X} \setminus \mathcal{Y}$. ◊

Definition A.18. Given two measure spaces $(X, \mathcal{M}, \mu)$ and $(Y, \mathcal{N}, \nu)$, a function $f : X \to Y$ is *measurable* if for every $B \in \mathcal{N}$, $f^{-1}(B) \in \mathcal{M}$. ◊

Let X be an arbitrary measure space with positive measure μ.

Definition A.19. Let $0 < p < +\infty$ and a measurable function $f : X \to \mathbb{C}$, define

$$\|f\|_p = \left(\int_X |f|^p d\mu \right)^{1/p},$$

and let $\mathcal{L}^p(\mu)$ consist of all the functions f which hold the following inequality:

$$\|f\|_p < +\infty.$$

◊

We can now define an equivalence relation in the space $\mathcal{L}^p(\mu)$. Given $f, g \in \mathcal{L}^p(\mu)$ we define the equivalence relation $\sim$ as follows:

$$f \sim g \iff f = g \quad \text{a.e.}$$

Theorem A.20. *The quotient set $L^p(\mu) := \mathcal{L}^p(\mu)/\sim$ is a Banach space (complete normed vector space) with the operations*

$$\begin{aligned} + : L^p(\mu) \times L^p(\mu) &\to L^p(\mu) \\ ([f],[g]) &\mapsto [f+g] \end{aligned}$$

,

$$\begin{aligned} \cdot : \mathbb{C} \times L^p(\mu) &\to L^p(\mu) \\ (\alpha,[f]) &\mapsto [\alpha f] \end{aligned}$$

and

$$\begin{aligned} \|\cdot\|_p : L^p(\mu) &\to \mathbb{C} \\ [f] &\mapsto \left(\int_X |f|^p d\mu\right)^{1/p}. \end{aligned}$$

◊

The special case where $p = 2$ gives us the following result.

Theorem A.21. *$L^2(\mu)$ is a Hilbert space with inner product*

$$\begin{aligned} \langle\cdot,\cdot\rangle : L^2(\mu) \times L^2(\mu) &\to \mathbb{C} \\ ([f],[g]) &\mapsto \int_X f\overline{g} d\mu. \end{aligned}$$

◊

Usually, the equivalence brackets are dropped for convenience and so, we tend to use $f, g \in L^p$ when we really mean $f, g \in \mathcal{L}^p$. This is why we tend to clarify equalities with the *a.e.* and saying that the equality holds in the L^p-sense.

In this thesis particular context, X is a measurable set of $\mathbb{R}^n$, $\mathcal{M}$ is the Borel sigma-algebra (normally denoted $\mathcal{B}$) and μ is the Lebesgue measure in $\mathbb{R}^n$. In this particular configuration, we call the associated L^2-space the $L^2(X)$ space. Here are some particular results of said spaces that we will use.

Theorem A.22. *The $L^2(\mathbb{R}^n)$ space is a separable Hilbert space i.e., it has a countable orthonormal basis.* ◊

Proof. The construction of the Haar wavelets family in $\mathbb{R}^n$ using the tensor product creates a countable orthonormal basis of $L^2(\mathbb{R}^n)$. □

Proposition A.23. *Given a function $f \in L^2(\mathbb{R})$. The tails of the function tend to* 0*, i.e.,*

$$\left\| f\chi_{(-\infty,-k)\cup(k,+\infty)} \right\|_2 \to 0 \quad \text{as } k \to +\infty.$$

Proof. Let $A_k = (-\infty, -k) \cup (k, +\infty)$ and define the sequence of functions $g_k(x) = |f(x)|^2 \, \chi_{A_k}(x)$. For any fixed $x \in \mathbb{R}$, there exists $K \in \mathbb{Z}$ so that for all $k > K$, $x \notin A_k$. Thus, $g_k(x) \to 0$ as $k \to \infty$.

Since $g_k(x) \leq |f(x)|^2$ for all k, and $|f|^2$ is integrable (as $f \in L^2(\mathbb{R})$), we can apply the *Dominated Convergence Theorem* (interchange of limit and integral):

$$\lim_{k\to\infty} \|f\chi_{A_k}\|_2^2 = \lim_{k\to\infty} \int_{\mathbb{R}} |f(x)|^2 \chi_{A_k}(x)\, dx = \int_{\mathbb{R}} \lim_{k\to\infty} \left(|f(x)|^2 \chi_{A_k}(x) \right) dx = 0.$$

□

Theorem A.24 (Density of Continuous Functions)**.** *Let $K \subset \mathbb{R}^n$ be a compact set. The set of continuous functions $C(K)$, is dense in $L^2(K)$.*

Proof. The proof is presented for the one-dimensional case $K = I = [a, b]$ for clarity. The generalization to $\mathbb{R}^n$ follows analogous steps using n-dimensional rectangles and constructing continuous approximations via Urysohn's Lemma or distance functions (see [41] chapters 2 and 3).

Let $f \in L^2(I)$ and let $\epsilon > 0$. The proof follows a standard three-step approximation argument:

1. **Approximation by Simple Functions.** By definition of the Lebesgue integral, there exists a simple function $s(x) = \sum_{k=1}^{N} c_k \chi_{E_k}(x)$ (where E_k are disjoint measurable sets and $c_k \in \mathbb{R}$) such that:

$$\|f - s\|_2 < \frac{\epsilon}{3}.$$

2. **Approximation of Measurable Sets by Intervals.** Since the Lebesgue measure is *outer regular*, for each measurable set E_k and any $\eta > 0$, there exists an open set U_k containing E_k such that $\mu(U_k \setminus E_k) < \eta/2$.

   Every open set in $\mathbb{R}$ is a countable union of disjoint open intervals, so we can write $U_k = \bigcup_{j=1}^{\infty} I_j$. Since E_k has finite measure (as $f \in L^2$), the sum of the lengths of these intervals is finite ($\sum \mu(I_j) = \mu(U_k) < \infty$).

   This convergence implies that the "tail" of the series tends to zero. Thus, we can choose a large enough integer M_k to truncate the union, defining a finite set of intervals $J_k = \bigcup_{j=1}^{M_k} I_j$, such that the measure of the discarded tail is negligible ($\mu(U_k \setminus J_k) < \eta/2$).

   The approximation error between the original set E_k and our finite intervals J_k is measured by the *symmetric difference* $E_k \Delta J_k = (E_k \setminus J_k) \cup (J_k \setminus E_k)$. Since both sets are contained in U_k, their difference lies within $(U_k \setminus J_k) \cup (U_k \setminus E_k)$. By the triangle inequality of measures:

$$\mu(E_k \Delta J_k) \leq \mu(U_k \setminus E_k) + \mu(U_k \setminus J_k) < \frac{\eta}{2} + \frac{\eta}{2} = \eta.$$

Crucially, the L^2 distance between characteristic functions corresponds exactly to the square root of the measure of their symmetric difference:

$$\|\chi_{E_k} - \chi_{J_k}\|_2 = \left(\int_{\mathbb{R}} |\chi_{E_k} - \chi_{J_k}|^2\right)^{1/2} = \left(\int_{\mathbb{R}} \chi_{E_k \Delta J_k}\right)^{1/2} = \sqrt{\mu(E_k \Delta J_k)} < \sqrt{\eta}.$$

Therefore, let $C = \max_{1 \le k \le N} |c_k|$. By choosing η sufficiently small (specifically $\eta < \frac{\epsilon^2}{9N^2C^2}$), we ensure the approximation holds. Defining the step function $\psi = \sum_{k=1}^{N} c_k \chi_{J_k}$, we apply the Triangle Inequality to conclude:

$$\begin{aligned}\|s - \psi\|_2 &= \left\| \sum_{k=1}^{N} c_k (\chi_{E_k} - \chi_{J_k}) \right\|_2 \le \sum_{k=1}^{N} |c_k| \, \|\chi_{E_k} - \chi_{J_k}\|_2 < \sum_{k=1}^{N} C\sqrt{\eta} \\ &= NC \frac{\epsilon}{3NC} = \frac{\epsilon}{3}.\end{aligned}$$

3. **Approximation of Characteristic Functions by Continuous Functions.** It suffices to show that the characteristic function of a single interval $J = (u, v) \subset I$ can be approximated by a continuous function. We define a trapezoidal function g_δ that is 1 on $[u + \delta, v - \delta]$, 0 outside (u, v), and linear in between. For δ sufficiently small,

$$\|\chi_J - g_\delta\|_2^2 = \int_I |\chi_J - g_\delta|^2 dx \le \int_u^{u+\delta} 1 dx + \int_{v-\delta}^{v} 1 dx = 2\delta.$$

By choosing δ small enough, we can make this error negligible. Extending this to the finite sum in ψ, we construct a continuous function $g \in C(I)$ such that $\|\psi - g\|_2 < \epsilon/3$.

By the Triangle Inequality:

$$\|f - g\|_2 \le \|f - s\|_2 + \|s - \psi\|_2 + \|\psi - g\|_2 < \frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} = \epsilon.$$

Thus, $C(I)$ is dense in $L^2(I)$. □

Theorem A.25. *The space of continuous functions with compact support is dense in* $L^2(\mathbb{R})$. *That is*

$$\forall f \in L^2(\mathbb{R}), \forall \epsilon > 0, \exists g \in C_c(\mathbb{R}) \text{ and } \exists h \in L^2(\mathbb{R}) \text{ with } \|h\|_2 < \epsilon, \text{ so that } f = g + h.$$

Proof. Given $f \in L^2(\mathbb{R})$, since f is square-integrable, the tail of the integral must vanish (by definition of its non-infinite integral). Thus, given $\epsilon > 0$, there exists a sufficiently large $K > 0$ such that

$$\left\| f \cdot \chi_{\{|x| > K\}} \right\|_2 < \epsilon/3.$$

By Thm. A.24, the set of continuous functions $C([-K, K])$ is dense in $L^2([-K, K])$. Therefore, we can find a function $g_1 \in C([-K, K])$ such that

$$\left\| (f - g_1) \cdot \chi_{[-K,K]} \right\|_2 < \epsilon/3.$$

However, simply extending g_1 to be zero outside $[-K, K]$ might introduce discontinuities at the boundaries if $g_1(\pm K) \neq 0$. To ensure continuity on all of $\mathbb{R}$, we define a new function $g \in C_c(\mathbb{R})$ that matches g_1 on the interval $[-K + \delta, K - \delta]$ and decays linearly to zero at $\pm K$. By choosing $\delta > 0$ sufficiently small, the L^2 difference between g_1 and this modified g is negligible:

$$\left\|(g_1 - g) \cdot \chi_{[-K,K]}\right\|_2 < \epsilon/3.$$

Finally, we define the error term $h = f - g$. By decomposing the domain and applying the Triangle Inequality, we verify that the norm is within the desired bound:

$$\begin{aligned}\|h\|_2 &= \left\|f \cdot \chi_{\{|x|>K\}} + (f - g_1) \cdot \chi_{[-K,K]} + (g_1 - g) \cdot \chi_{[-K,K]}\right\|_2 \\ &\leq \left\|f \cdot \chi_{\{|x|>K\}}\right\|_2 + \left\|(f - g_1) \cdot \chi_{[-K,K]}\right\|_2 + \left\|(g_1 - g) \cdot \chi_{[-K,K]}\right\|_2 \\ &< \frac{\epsilon}{3} + \frac{\epsilon}{3} + \frac{\epsilon}{3} = \epsilon.\end{aligned}$$

Thus, $f = g + h$ with $g \in C_c(\mathbb{R})$ and $\|h\|_2 < \epsilon$. □

A.4 The ℓ^p-spaces

Definition A.26. Let $0 < p < +\infty$ and an index set I, the set of sequences $s : I \to \mathbb{C}$, define

$$\ell^p(I) = \left\{ \{a\}_{n\in I} : \left(\sum_{n\in I} |a_n|^p\right)^{1/p} < +\infty \right\}.$$

◊

Theorem A.27. *The set $\ell^p(I)$ is a Banach space (complete normed vector space) with the operations*

$$\begin{aligned} + : \ell^p(I) \times \ell^p(I) &\to \ell^p(I) \\ (\{a_n\}_{n\in I}, \{b_n\}_{n\in I}) &\mapsto \{a_n + b_n\}_{n\in I} \end{aligned}$$

,

$$\begin{aligned} \cdot : \mathbb{C} \times \ell^p(I) &\to \ell^p(I) \\ (\alpha, \{a_n\}_{n\in I}) &\mapsto \{\alpha a_n\}_{n\in I} \end{aligned}$$

and

$$\begin{aligned} \|\cdot\|_p : \ell^p(I) &\to \mathbb{C} \\ \{a_n\}_{n\in I} &\mapsto \left(\sum_{n\in I} |a_n|^p\right)^{1/p}. \end{aligned}$$

◊

The special case where $p = 2$ gives us the following result.

Theorem A.28. *$\ell^2(I)$ is a Hilbert space with inner product*

$$\begin{aligned}\langle\cdot,\cdot\rangle : \ell^p(I)\times\ell^p(I) &\to \mathbb{C}\\ (\{a_n\}_{n\in I},\{b_n\}_{n\in I}) &\mapsto \sum_{n\in I} a_n\overline{b_n}.\end{aligned}$$

◊

In this thesis, we will use $I=\mathbb{Z}$ and thus, the Hilbert space $\ell^2(\mathbb{Z})$.

A.5 Tensor Product of Orthonormal Bases

In this section we will present some results that are key to the definition of the *tensor product* of closed subspaces of the $L^2(\mathbb{R}^n)$ and $L^2(\mathbb{R}^p)$ spaces.

Lemma A.29. *Let $f_0, f_1 \in L^2(\mathbb{R}^n)$ and $g_0, g_1 \in L^2(\mathbb{R}^p)$. The following functions*

$$\begin{aligned}h_0(x,y) &= f_0(x)g_0(y) \quad \textit{and}\\ h_1(x,y) &= f_1(x)g_1(y) \quad \textit{for all } x\in\mathbb{R}^n, y\in\mathbb{R}^p\end{aligned}$$

live in $L^2(\mathbb{R}^{n+p})$ and hold the following equality:

$$\langle h_0, h_1\rangle = \langle f_0, f_1\rangle\langle g_0, g_1\rangle.$$

◊

Proof. First of all, we need to check that $h_0, h_1 \in L^2(\mathbb{R}^{n+p})$. Since the proof is identical for both, let $h(x,y)=f(x)g(y)$ be the generalization. Using Tonelli's Theorem for non-negative functions, we operate as follows:

$$\begin{aligned}\|h\|^2_{L^2(\mathbb{R}^{n+p})} &= \int_{\mathbb{R}^{n+p}} |f(x)g(y)|^2\,dx\,dy = \left(\int_{\mathbb{R}^n}|f(x)|^2\,dx\right)\left(\int_{\mathbb{R}^p}|g(y)|^2\,dy\right)\\ &= \|f\|^2_{L^2(\mathbb{R}^n)}\|g\|^2_{L^2(\mathbb{R}^p)}.\end{aligned}$$

Since $f\in L^2(\mathbb{R}^n)$ and $g\in L^2(\mathbb{R}^p)$, both norms on the right-hand side are finite, which implies that $\|h\|_{L^2(\mathbb{R}^{n+p})} < +\infty$.

Now, using Fubini, we can see that

$$\begin{aligned}\langle h_0, h_1\rangle &= \int_{\mathbb{R}^{n+p}} f_0(x)g_0(y)\overline{f_1(x)g_1(y)}\,d(x,y)\\ &= \left(\int_{\mathbb{R}^n} f_0(x)\overline{f_1(x)}\,dx\right)\left(\int_{\mathbb{R}^p} g_0(y)\overline{g_1(y)}\,dy\right)\\ &= \langle f_0, f_1\rangle\langle g_0, g_1\rangle.\end{aligned}$$

□

From now on, the norm will be the L^2-norm of the space on which is applied.

Proposition A.30. *Let $\{f_k\}_{k\in\mathbb{N}} \subset L^2(\mathbb{R}^n)$ be an orthonormal set and $\{g_\ell\}_{\ell\in\mathbb{N}} \subset L^2(\mathbb{R}^p)$ be another orthonormal set. Then, the set*

$$\{h_{k,\ell}(x,y) := f_k(x)g_\ell(y)\,|\,\forall x\in\mathbb{R}^n, \forall y\in\mathbb{R}^p\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}}$$

is an orthonormal set of $L^2(\mathbb{R}^{n+p})$ ◊

Proof. Given $(k_1, \ell_1), (k_2, \ell_2) \in \mathbb{N} \times \mathbb{N}$, by Lemma A.29 we see that

$$\langle h_{k_1,\ell_1}, h_{k_2,\ell_2} \rangle = \langle f_{k_1}, f_{k_2} \rangle \langle g_{\ell_1}, g_{\ell_2} \rangle = \delta_{k_1,k_2} \delta_{l_1,l_2}.$$

□

Corollary A.31. *Given $f \in L^2(\mathbb{R}^n)$, $g \in L^2(\mathbb{R}^p)$ and $h(x,y) := f(x)g(y)$ for every $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^p$, then*

$$\|h\| = \|f\| \, \|g\| .$$

◊

Proposition A.32. *Given a closed space V of $L^2(\mathbb{R}^n)$, then, V has a countable orthonormal basis.* ◊

Proof. Thm. A.22 states that $L^2(\mathbb{R}^n)$ is a separable Hilbert space, therefore by Cor. A.7, V is also a separable Hilbert space and thus, accepts a countable orthonormal basis. □

The previous results allow us to give the following definition:

Definition A.33. *Let V be a closed space of $L^2(\mathbb{R}^n)$ with orthonormal basis $\{v_k\}_{k\in\mathbb{N}}$ and let U be a closed space of $L^2(\mathbb{R}^p)$ with orthonormal basis $\{u_\ell\}_{\ell\in\mathbb{N}}$. Let also the set*

$$\{w_{k,\ell}(x,y) := v_k(x)u_\ell(y) \,|\, \forall x \in \mathbb{R}^n, \forall y \in \mathbb{R}^p\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}}.$$

Then, we define the tensor product of V and U, denoted as $V \otimes U$, as the closure of its span $\overline{\operatorname{span}(\{w_{k,\ell}\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}})}$, i.e., the family $\{w_{k,\ell}\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}}$ is an orthonormal basis of $V \otimes U$. ◊

Proposition A.34. *Let V be a closed subspace of $L^2(\mathbb{R}^n)$ and U be a closed subspace of $L^2(\mathbb{R}^p)$. The definition of the tensor product space $V \otimes U$ is independent of the chosen bases.*

Specifically, if $\{f_k\}$ and $\{v_k\}$ are orthonormal bases for V, and $\{g_\ell\}$ and $\{u_\ell\}$ are orthonormal bases for U, then:

$$\overline{\operatorname{span}\left(\{f_k(x)g_\ell(y)\}_{k,\ell}\right)} = \overline{\operatorname{span}\left(\{v_k(x)u_\ell(y)\}_{k,\ell}\right)}. \tag{A.2}$$

Consequently, both sets form orthonormal bases for the tensor space $V \otimes U$.

Proof. Let $H = \overline{\operatorname{span}(\{w_{k,\ell}\})}$ be the space generated by the basis $\{v_k\}$ and $\{u_\ell\}$, where $w_{k,\ell}(x,y) = v_k(x)u_\ell(y)$. We want to show that for any basis element of the other set, $h_{k,\ell}(x,y) = f_k(x)g_\ell(y)$, it holds that $h_{k,\ell} \in H$.

Let $(k,\ell) \in \mathbb{N} \times \mathbb{N}$. Since $\{v_i\}_{i\in\mathbb{N}}$ is a basis for V and $\{u_j\}_{j\in\mathbb{N}}$ is a basis for U, for any $\epsilon > 0$, there exists an $N \in \mathbb{N}$ sufficiently large such that the finite partial sums approximate the functions:

$$\left\| f_k - \sum_{i=1}^{N} \langle f_k, v_i \rangle v_i \right\| < \frac{\epsilon}{2} \quad \text{and} \quad \left\| g_\ell - \sum_{j=1}^{N} \langle g_\ell, u_j \rangle u_j \right\| < \frac{\epsilon}{2}.$$

Let us define this approximations for further manipulation:

$$v_\epsilon = \sum_{i=1}^{N} \langle f_k, v_i \rangle v_i \quad \text{and} \quad u_\epsilon = \sum_{j=1}^{N} \langle g_\ell, u_j \rangle u_j.$$

Then, applying the triangle inequality and Cor. A.31 (norm of the product is product of norms):

$$\begin{aligned} \|f_k g_\ell - v_\epsilon u_\epsilon\| &\leq \|f_k g_\ell - v_\epsilon g_\ell\| + \|v_\epsilon g_\ell - v_\epsilon u_\epsilon\| \\ &= \|(f_k - v_\epsilon) g_\ell\| + \|v_\epsilon (g_\ell - u_\epsilon)\| \\ &= \|f_k - v_\epsilon\| \|g_\ell\| + \|v_\epsilon\| \|g_\ell - u_\epsilon\|. \end{aligned}$$

Recall that $\|g_\ell\| = 1$ given by the fact that it is an orthonormal basis. Also, by Bessel's inequality, $\|v_\epsilon\| \leq \|f_k\| = 1$. Thus:

$$\|f_k g_\ell - v_\epsilon u_\epsilon\| < \frac{\epsilon}{2} \cdot 1 + 1 \cdot \frac{\epsilon}{2} = \epsilon.$$

Notice that the product $v_\epsilon(x) u_\epsilon(y)$ is a finite linear combination of terms $v_i(x) u_j(y)$, which means $v_\epsilon u_\epsilon \in \operatorname{span}(\{w_{i,j}\}) \subset H$.

Since for every $\epsilon > 0$ we can find an element in $\operatorname{span}(\{w_{i,j}\})$ arbitrarily close to $h_{k,\ell}$, it follows that $h_{k,\ell}$ belongs to the closure, i.e., $h_{k,\ell} \in H$.

By symmetry, the reverse inclusion is immediately given. Therefore, the closed spaces generated by both product bases are identical. □

Now, the following lemma gives us how to find elements for the tensor product space $V \otimes U$.

Lemma A.35. *Let V be a closed space of $L^2(\mathbb{R}^n)$ and let U be a closed space of $L^2(\mathbb{R}^p)$. If $f \in V$ and $g \in U$, then $f(x)g(y) \in V \otimes U$.* ◊

Proof. Let $\{v_k\}_{k \in \mathbb{N}}$ and $\{u_\ell\}_{\ell \in \mathbb{N}}$ be orthonormal bases of V and U respectively. Since the functions $\{v_k(x) u_\ell(y)\}_{(k,\ell) \in \mathbb{N} \times \mathbb{N}}$ define an orthonormal basis of the space $V \otimes U$, we need to prove that for all $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^p$,

$$\lim_{N \to +\infty} \lim_{M \to +\infty} \left\| f(x)g(y) - \sum_{i=1}^{N} \sum_{j=1}^{M} \langle f(x)g(y), v_i(x) u_j(y) \rangle v_i(x) u_j(y) \right\| = 0.$$

For all $N, M \in \mathbb{N}$, applying Lem. A.29 we get the following:

$$\begin{aligned} &\left\| f(x)g(y) - \sum_{i=1}^{N} \sum_{j=1}^{M} \langle f(x)g(y), v_i(x) u_j(y) \rangle v_i(x) u_j(y) \right\| \\ &\qquad = \left\| f(x)g(y) - \sum_{i=1}^{N} \langle f, v_i \rangle v_i(x) \left(\sum_{j=1}^{M} \langle g, u_j \rangle u_j(y) \right) \right\| \\ &\qquad \leq \left\| f(x)g(y) - f(x) \left(\sum_{j=1}^{M} \langle g, u_j \rangle u_j(y) \right) \right\| \\ &\qquad\quad + \left\| f(x) \left(\sum_{j=1}^{M} \langle g, u_j \rangle u_j(y) \right) - \sum_{i=1}^{N} \langle f, v_i \rangle v_i(x) \left(\sum_{j=1}^{M} \langle g, u_j \rangle u_j(y) \right) \right\|. \end{aligned}$$

To finish the proof, we need to see that the two summands on the right-hand side tend to 0. For the first term, after distributing, given the Cor. A.31 we see that

$$\left\| f(x)g(y) - f(x)\left(\sum_{j=1}^{M}\langle g, u_j\rangle u_j(y)\right)\right\|^2 = \|f\|^2 \left\| g - \sum_{j=1}^{M}\langle g, u_j\rangle u_j \right\|^2 \xrightarrow[M\to+\infty]{} 0.$$

On the other hand, operating similarly

$$\begin{aligned}\left\| f(x)\left(\sum_{j=1}^{M}\langle g, u_j\rangle u_j(y)\right) - \sum_{i=1}^{N}\langle f, v_i\rangle v_i(x)\left(\sum_{j=1}^{M}\langle g, u_j\rangle u_j(y)\right)\right\|^2 \\ = \left\| f - \sum_{i=1}^{N}\langle f, v_i\rangle v_i \right\|^2 \left\| \sum_{j=1}^{M}\langle g, u_j\rangle u_j \right\|^2 \\ \xrightarrow[N,M\to+\infty]{} 0 \cdot \|g\|^2 = 0.\end{aligned}$$

□

The tensor product and the direct sum hold the mixed distributive property as given in the next result.

Proposition A.36. *Let V and W be closed subspaces of $L^2(\mathbb{R}^n)$ such that $V \perp W$ and let U be a closed space of $L^2(\mathbb{R}^p)$. Then, the following equalities hold:*

1. $U \otimes (V \oplus W) = (U \otimes V) \oplus (U \otimes W)$.
2. $(V \oplus W) \otimes U = (V \otimes U) \oplus (W \otimes U)$.

◊

Proof. 1. Let $\{u\}_{k\in\mathbb{N}}$, $\{v_\ell\}_{\ell\in\mathbb{N}}$ and $\{w_\ell\}_{\ell\in\mathbb{N}}$ be orthonormal bases of U, V and W, respectively. We will show that the left-hand side and right-hand side have the same basis, as then its closure has to be the same.

- **Right-hand side:** First, we show $(U \otimes V) \perp (U \otimes W)$. Following the Def. A.33, the set $\{u_k(x)\ v_\ell(y)\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}}$ is a basis for $U \otimes V$ and $\{u_k(x)\ w_\ell(y)\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}}$ is a basis for $U \otimes W$. Using Lemma A.29, their inner product is:

  $$\langle u_k v_\ell, u_m w_n\rangle = \langle u_k, u_m\rangle\langle v_\ell, w_n\rangle. \quad \text{for each pairs } (k,\ell),(n,m) \in \mathbb{N}\times\mathbb{N}.$$

  Since $V \perp W$, the term $\langle v_\ell, w_n\rangle$ is always 0. Thus, the subspaces are orthogonal. Consequently, the union of their bases, denoted by $\mathcal{B}$, forms an orthonormal basis for the direct sum $(U \otimes V) \oplus (U \otimes W)$:

  $$\mathcal{B} = \{u_k(x)v_\ell(y)\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}} \cup \{u_k(x)w_\ell(y)\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}}.$$

- **Left-hand side:** Since $V \perp W$, the union $\{v_\ell\}_\ell \cup \{w_\ell\}_\ell$ is an orthonormal basis for $V \oplus W$. By the definition of tensor product (Def. A.33), the basis for $U \otimes (V \oplus W)$ is formed by taking the products of $\{u_k\}$ with this union. This results in exactly the same set $\mathcal{B}$.

  Since both spaces are generated by the same orthonormal basis $\mathcal{B}$, they are equal.

2. Item (2) follows by a strictly symmetrical argument, swapping the roles of the first and second spaces.

□

To end the section we give a result that shows that $L^2(\mathbb{R}^n) \otimes L^2(\mathbb{R}^p) = L^2(\mathbb{R}^{n+p})$.

Theorem A.37. *Let $\{f_k\}_{k\in\mathbb{N}}$ be an orthonormal basis of $L^2(\mathbb{R}^n)$ and $\{g_\ell\}_{\ell\in\mathbb{N}}$ be an orthonormal basis of $L^2(\mathbb{R}^p)$. Then, the system $\{f_k(x)g_\ell(y)\}_{(k,\ell)\in\mathbb{N}\times\mathbb{N}}$ defines an orthonormal basis of the space $L^2(\mathbb{R}^{n+p})$.* ◊

Appendix B

Complementary Figures and Tables

Table B.1: Detailed Per-Class Segmentation Metrics (IoU and Acc). This table presents the semantic segmentation performance for each of the 104 classes in the FoodSeg103 dataset. The IoU and Acc are calculated according to Equations 8.1 and 8.2, respectively. The results shown correspond to the best-performing model configuration: ZM with $N = 32$, utilizing the RGB Fusion + SE-Block Attention refinement architecture.

Class	**IoU (%)**	**Acc (%)**
background	92.17	96.30
candy	9.59	12.66
egg tart	0.00	0.00
french fries	60.78	73.98
chocolate	19.39	35.64
biscuit	39.13	55.43
popcorn	50.05	71.00
pudding	0.00	0.00
ice cream	42.38	65.19
cheese butter	7.40	11.21
cake	36.14	51.74
wine	18.13	24.51
milkshake	28.13	43.91
coffee	26.76	32.62
juice	36.75	51.55
milk	17.16	19.75
tea	22.06	42.92
almond	25.03	29.23
red beans	11.48	16.26
cashew	0.00	0.00
dried cranberries	1.02	1.02
soy	31.60	56.53
walnut	37.51	44.94
peanut	0.00	0.00

Continued on next page...

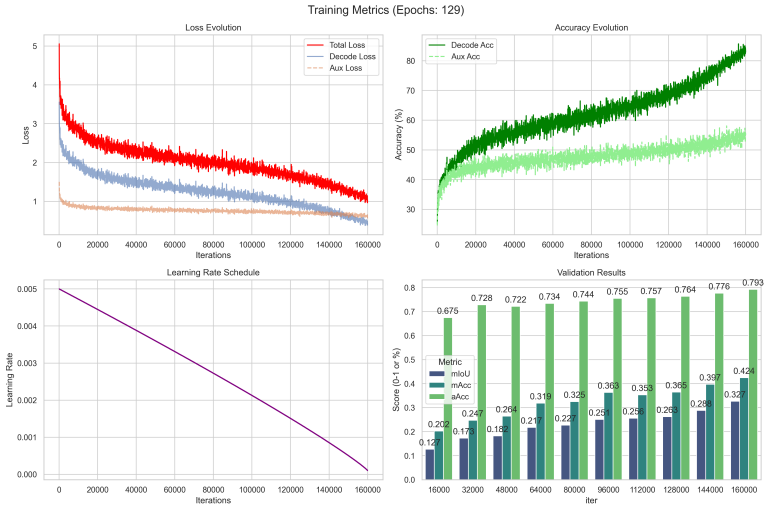
Table B.1 – Continued from previous page

Class	**IoU (%)**	**Acc (%)**
egg	53.87	71.73
apple	40.56	57.75
date	0.00	0.00
apricot	1.83	3.50
avocado	38.06	40.76
banana	69.35	75.16
strawberry	67.52	81.31
cherry	39.57	53.50
blueberry	64.21	81.78
raspberry	34.83	52.56
mango	17.20	22.46
olives	35.59	38.81
peach	23.90	32.19
lemon	70.20	88.92
pear	4.95	10.65
fig	49.42	76.89
pineapple	18.88	34.33
grape	58.47	65.10
kiwi	73.63	95.52
melon	2.75	2.93
orange	44.01	54.79
watermelon	37.52	68.82
steak	35.86	48.63
pork	16.02	32.20
chicken duck	39.73	60.90
sausage	31.11	48.20
fried meat	23.05	31.23
lamb	11.62	14.54
sauce	36.87	56.99
crab	9.74	9.91
fish	28.18	42.63
shellfish	55.18	65.70
shrimp	51.50	67.22
soup	41.07	57.26
bread	51.74	67.06
corn	83.08	89.54
hamburg	0.00	0.00
pizza	33.05	42.90
hanamaki baozi	1.85	6.20
wonton dumplings	8.47	12.43
pasta	69.29	89.84
noodles	54.15	60.95
rice	68.84	88.03
pie	36.13	46.47
tofu	20.58	24.50
eggplant	1.15	1.38

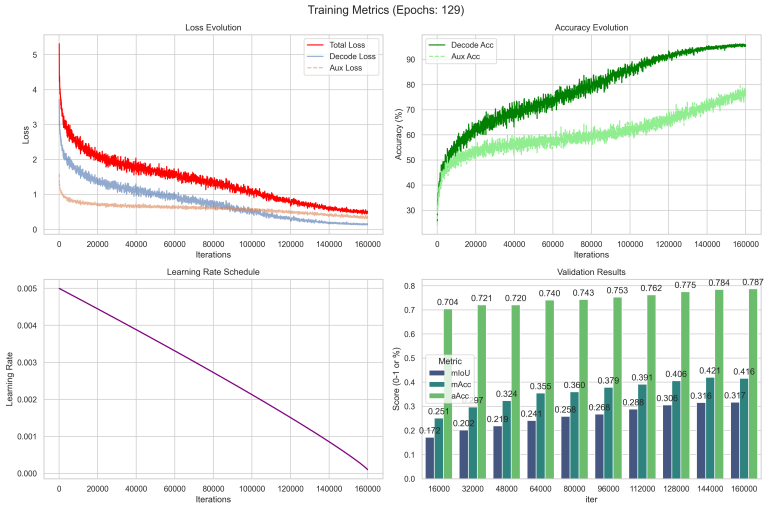
Continued on next page...

Table B.1 – Continued from previous page

Class	**IoU (%)**	**Acc (%)**
potato	57.25	82.45
garlic	28.08	44.02
cauliflower	61.20	82.30
tomato	71.57	84.57
kelp	0.00	0.00
seaweed	60.99	62.63
spring onion	14.18	26.38
rape	23.64	46.61
ginger	27.98	31.32
okra	6.27	7.02
lettuce	47.67	61.29
pumpkin	36.05	64.52
cucumber	62.67	74.46
white radish	18.71	24.96
carrot	76.08	89.57
asparagus	65.57	76.73
bamboo shoots	0.00	0.00
broccoli	85.57	93.81
celery stick	65.96	76.36
cilantro mint	48.21	73.16
snow peas	44.81	51.27
cabbage	38.32	61.65
bean sprouts	25.33	32.35
onion	28.63	49.13
pepper	45.01	55.02
green beans	70.39	75.06
French beans	68.13	89.49
king oyster mushroom	0.00	0.00
shiitake	15.94	22.16
enoki mushroom	0.00	0.00
oyster mushroom	0.00	0.00
white button mushroom	23.58	36.85
salad	4.78	8.19
other ingredients	0.61	0.81



(a) Reference: Baseline training dynamics with standard Data Augmentation.



(b) Ablation Study: ZM configuration without Data Augmentation (showing overfitting).

Figure B.1: Comparative analysis of training dynamics with and without data augmentation. The figure juxtaposes the training logs for the baseline (a) against the experimental ZM configuration without geometric data augmentation (b). Each panel displays four key metrics: **(Top-Left) Loss Evolution:** Tracks the convergence of the *Total Loss* (red), which is the weighted sum of the primary *Decode Loss* (blue) and the auxiliary head *Aux Loss* (orange) by the *Deep Supervision* implementation (Sec. 6.1.2). **(Top-Right) Training Accuracy:** Illustrates the aAcc metric for the main decoder (dark green) and the auxiliary head (light green). **(Bottom-Left) Learning Rate:** Confirms the execution of the linear decay schedule over the 160k iterations. **(Bottom-Right) Validation Results:** A bar chart monitoring *mIoU*, *mAcc* and *aAcc* metrics on the validation set every 16k steps. Crucially, the "No Augmentation" experiment (b) exhibits clear signs of overfitting: while the training accuracy continues to rise, the validation performance saturates and degrades in the final stages, with mIoU dropping from a peak of 0.441 at 144k iterations to 0.431 at the end of training. This degradation is not exhibited in the baseline (a), thereby validating the necessity of the data augmentation pipeline.

Appendix C

List of Symbols

Symbol	Description
Sets, Operators and Functional Spaces	
$\mathbb{R}$	The set of real numbers
$\mathbb{Z}$	The set of integers
$\mathbb{N}_0$	The set of natural numbers including zero ($\{0\} \cup \mathbb{N}$)
$\mathbb{C}$	The set of complex numbers
$L^2(\mathbb{R})$	Space of square-integrable functions defined on $\mathbb{R}$
$L^2(B(0,1))$	Space of square-integrable functions defined on the unit disk
$l^2(\mathbb{Z})$	Space of square-summable sequences indexed by integers
$C_c(\mathbb{R})$	Continuous functions with compact support defined on $\mathbb{R}$
$B(0,1)$	The unit disk $\{(x,y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$
$\langle f, g \rangle$	Inner product between functions f and g
$\|\|f\|\|_2$	L^2-norm (Euclidean norm) of the function f
$\chi_A(x)$	Characteristic function of the set A (1 if $x \in A$, 0 otherwise)
δ_{ij}	Kronecker delta (1 if $i = j$, 0 otherwise)
$A^\perp$	Orthogonal complement of the set A
$\oplus$	Direct sum of subspaces
$\otimes$	Tensor product of spaces or functions
Multiresolution Analysis (MRA) and Haar Wavelets	
ϕ	Scaling function of an MRA
ψ	Wavelet function
$\phi_{j,k}$	Dilated and translated scaling function
$\psi_{j,k}$	Dilated and translated wavelet function: $2^{j/2}\psi(2^j x - k)$
$h(x)$	The Haar Wavelet function ($h_{[0,1)}$)
$h_I(x)$	Haar function associated to the dyadic interval I
$\{h_k\}$	Low-pass (scaling) filter sequence
$\{g_k\}$	High-pass (wavelet) filter sequence
$a_{j,k}$	Discrete approximation coefficients (Haar Moments)
$d_{j,k}$	Discrete detail coefficients (Haar Wavelet coefficients)
$\mathcal{D}$	The set of all dyadic intervals in $\mathbb{R}$
$\mathcal{D}_j$	The set of all dyadic intervals in $\mathbb{R}$ with length 2^{-j}
$I_{j,k}$	Dyadic interval of generation j and shift k
V_j	Approximation subspace at resolution level j ($V_j \subset V_{j+1}$)

Symbol	Description
W_j	Detail (wavelet) subspace at resolution level j
$\mathcal{V}_j$	Tensor product approximation space $V_j \otimes V_j$ in 2D
$\mathcal{W}_j$	Tensor product detail space in 2D
P_j	Orthogonal projection operator onto the subspace V_j
Q_j	Orthogonal projection operator onto the subspace W_j
ψ^h, ψ^v, ψ^d	Two-dimensional Horizontal, Vertical, and Diagonal wavelets
Zernike Analysis	
$Z_n^m(r, \phi)$	Real Zernike polynomial of radial degree n and azimuthal degree m
$R_n^{\|m\|}(r)$	Radial polynomial component of the Zernike function
γ_n^m	Normalization constant for Zernike polynomials
$Z_j(r, \phi)$	Real Zernike polynomial with single ANSI index j
V_N	Space of polynomials on the unit disk with degree at most N
J	Dimension of the space V_N ($J = (N+1)(N+2)/2$)
D_N	Dimension of the wavelet space W_N
$\mathcal{Z}_N$	Family of Zernike polynomials forming an orthonormal basis for V_N
K_N	Reproducing Kernel Polynomial of degree N
$\phi_{N,j}$	Zernike Scaling Function localized at point P_j
$\psi_{N,j}$	Zernike Wavelet Function (difference of Kernels)
A_N	Collocation matrix of Zernike polynomials
l_k	Fundamental Lagrange interpolating polynomial (dual basis)
$P_j^{(N)}$	Regular Points (Optimal Concentric Sampling) for degree N
$Q_j^{(N)}$	Parameter points (Approximate Fekete Points) for wavelets
$\mathcal{V}_N$	Zernike Vandermonde Matrix
$\mathcal{V}_N^\dagger$	Moore-Penrose pseudoinverse of the Vandermonde matrix
$\mathcal{B}_{MRA}$	Global basis matrix for the Zernike MRA decomposition
Metrics and Learning	
$IoU^{(k)}$	Intersection over Union for class k
$mIoU$	Mean Intersection over Union
$Acc^{(k)}$	Pixel accuracy for class k
$mAcc$	Mean Class Accuracy
$aAcc$	Average Pixel Accuracy
$\mathcal{L}_{total}$	Total Loss function
$\mathcal{L}_{main}$	Main decoder loss
$\mathcal{L}_{aux}$	Auxiliary head loss

Bibliography

[1] A. AlMughrabi *et al.* "BenchSeg: A Large-Scale Dataset and Benchmark for Multi-View Food Video Segmentation", arXiv preprint 2601.07581, 2026.

[2] A. AlMughrabi, R. Marques, and P. Radeva, "MomentsNeRF: Leveraging Orthogonal Moments for Few-Shot Neural Rendering", *arXiv preprint arXiv:2405.08129*, 2024.

[3] A. AlMughrabi, A. Galán, R. Marques, and P. Radeva, "FoodMem: Near Real-time and Precise Food Video Segmentation", *arXiv preprint arXiv:2407.12121*, 2025.

[4] I. Bello, B. Zoph, A. Vaswani, J. Shlens, and Q. V. Le, "Attention Augmented Convolutional Networks", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3286–3295, 2019.

[5] A. Boggess and F. J. Narcowich, *A First Course in Wavelets with Fourier Analysis*, 2nd ed., John Wiley & Sons, New Jersey, 2009.

[6] R. Bommasani et al., "On the Opportunities and Risks of Foundation Models", *arXiv preprint arXiv:2108.07258*, 2021.

[7] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101 – Mining Discriminative Components with Random Forests", *European Conference on Computer Vision*, 2014.

[8] H. Brezis, *Functional Analysis, Sobolev Spaces and Partial Differential Equations*, Springer, New York, 2011.

[9] Carothers, N. L. (2000). *Real Analysis*. Cambridge: Cambridge University Press, p. 293. ISBN 9780521497565.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[11] H. K. Cheng, S. W. Oh, B. Price, A. Schwing, and J.-Y. Lee, "Tracking Anything with Decoupled Video Segmentation", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1316–1326, 2023.

[12] H. K. Cheng and A. G. Schwing, "XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model", in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 640–658, 2022.

[13] G. Ciocca, P. Napoletano, and R. Schettini, "Food Recognition: A New Dataset, Experiments, and Results", *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 585–598, 2017.

[14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding", *arXiv preprint arXiv:1604.01685*, 2016.

[15] S. Datta and K. B. Datta, "Wavelets for $L^2(B(0,1))$ using Zernike polynomials", *arXiv preprint arXiv:2405.08129v2*, 2025.

[16] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", *International Conference on Learning Representations (ICLR)*, 2021.

[17] C. Esteves *et al.*, "Spectral Image Tokenizer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. [Online]. Available at `https://openaccess.thecvf.com/content/ICCV2025/papers/Esteves_Spectral_Image_Tokenizer_ICCV_2025_paper.pdf` [Accessed: Jan. 8, 2026].

[18] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks", in *International Conference on Learning Representations (ICLR)*, 2019.

[19] A. Haar, *Zur Theorie der orthogonalen Funktionensysteme*, Math. Annal. 69 (1910), 331–371.

[20] Y. He *et al.*, "FoodLMM: A Versatile Food Assistant using Large Multimodal Models", *arXiv preprint arXiv:2312.15222*, 2023.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[22] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of Tricks for Image Classification with Convolutional Neural Networks", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 558–567, 2019.

[23] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks", *arXiv preprint arXiv:1709.01507*, 2019.

[24] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. S. Huang, "CCNet: Criss-Cross Attention for Semantic Segmentation", *arXiv preprint arXiv:1811.11721*, 2020.

[25] A. Kirillov *et al.*, "Segment Anything", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4015–4026, 2023.

[26] X. Lan, J. Lyu, H. Jiang, K. Dong, Z. Niu, Y. Zhang, and J. Xue, "FoodSAM: Any Food Segmentation", *IEEE Transactions on Multimedia*, pp. 1–14, 2023.

[27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117–2125, 2017.

[28] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[29] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases for* $L^2(\mathbb{R})$, Trans. Amer. Math. Soc. 315 (1989), 69–87.

[30] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis", in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 405–421, 2020.

[31] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A Survey on Food Computing", *arXiv preprint arXiv:1808.07202*, 2019.

[32] MMSegmentation Contributors, "MMSegmentation Documentation", *OpenMMLab*, 2024. [Online]. Available at `https://mmsegmentation.readthedocs.io/_/downloads/en/0.x/pdf/` [Accessed: Jan. 4, 2026].

[33] S. P. Mohanty et al., "The Food Recognition Benchmark: Using Deep Learning to Recognize Food in Images", *Frontiers in Nutrition*, vol. 9, p. 875143, 2022.

[34] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the Hilbert space-filling curve", *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 1, pp. 124–141, 2001.

[35] B. R. A. Nijboer, *The Diffraction Theory of Aberrations*. Ph.D. thesis, University of Groningen, 1942. Available at `https://nijboerzernike.nl/_downloads/Thesis_Nijboer.pdf`.

[36] K. Niu and C. Tian, "Zernike polynomials and their applications", *Journal of Optics*, vol. 24, no. 12, p. 123001, 2022.

[37] The PyTorch Foundation, "PyTorch Documentation". [Online]. Available at `https://pytorch.org/docs/stable/index.html` [Accessed: Jan. 4, 2026].

[38] The PyTorch Team, "Custom C++ and CUDA Extensions", *PyTorch Tutorials*. [Online]. Available at `https://docs.pytorch.org/tutorials/advanced/custom_ops_landing_page.html` [Accessed: Jan. 8, 2026].

[39] D. Ramos-López, M. A. Sańchez-Granero, M. Fernández-Martínez, and A. Martínez-Finkelshtein, "Optimal sampling patterns for Zernike polynomials",*Applied Mathematics and Computation*, vol. 274, pp. 247–257, 2016.

[40] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, Springer, 2015.

[41] W. Rudin, *Real and complex analysis*. McGraw-Hill, Inc., 1987.

[42] A. Sommariva and M. Vianello, "Computing approximate Fekete points by QR factorizations of Vandermonde matrices", *Computers & Mathematics with Applications*, vol. 57, no. 8, pp. 1324–1336, 2009.

[43] M. R. Teague, "Image analysis via the general theory of moments", *Journal of the Optical Society of America*, vol. 70, no. 8, pp. 920–930, 1980.

[44] I. Ulku and E. Akagündüz, "A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D Images", *Applied Artificial Intelligence*, vol. 36, no. 1, p. 2032924, 2022.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L-LKaiser, and I. Polosukhin, "Attention is all you need", in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[46] G. K. Wallace, "The JPEG still picture compression standard", *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[47] Z. Wang *et al.*, "Diffusion Transformer meets Multi-level Wavelet Spectrum for Single Image Super-Resolution," *arXiv preprint arXiv:2511.01175*, 2025.

[48] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic Networks: Deep Translation and Rotation Invariance", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5028–5037, 2017.

[49] H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, "FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation", *arXiv preprint arXiv:1903.11816*, 2019.

[50] X. Wu, X. Fu, Y. Liu, E.-P. Lim, S. C. H. Hoi, and Q. Sun, "A Large-Scale Benchmark for Food Image Segmentation", *arXiv preprint arXiv:2105.05409*, 2021.

[51] T. Xiao *et al.*, "Unified Perceptual Parsing for Scene Understanding", *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[52] G. Xu, W. Liao, X. Zhang, C. Li, X. He, and X. Wu, "Haar wavelet downsampling: A simple but effective downsampling module for semantic segmentation", *Pattern Recognition*, vol. 121, p. 108306, 2022.

[53] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2881–2890, 2017.

[54] S. Zheng *et al.*, "Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[55] P. Zheng, D. Gao, S. Deng, R. Qian, and H. Fu, "BiRefNet: Bilateral Reference Network for High-Resolution Dichotomous Image Segmentation", *arXiv preprint arXiv:2401.04415*, 2024.

[56] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic Understanding of Scenes through the ADE20K Dataset", *arXiv preprint arXiv:1608.05442*, 2018.