Image Reconstruction and Comparison for Forensic Footwear Analysis

Thesis

Submitted in Partial Fulfillment of the requirements of: BITS F422T Thesis
By

Gautham Venkatasubramanian
ID No. 2014B4A70637P

Under the Supervision of:
Dr. Martin Herman

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
PILANI CAMPUS
May 2019

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Martin Herman, for allowing me to pursue this project under his supervision, and for his guidance and direction in all facets of my work in this project.

I would like to thank Dr. Sangita Yadav for consenting to be my co-supervisor for this project, and the Department of Mathematics, BITS Pilani, for their assistance and support.

I am grateful to Dr. Hari Iyer and Dr. Steve Lund, who provided many valuable suggestions that helped me gain new insights whenever I was stuck with some problem. Their feedback on my work helped me understand certain nuances of the concepts I was dealing with.

I would like to thank the other members of the NIST Forensic Footwear Team for their reflections on my work during the weekly meetings, which provided me with many ideas to refine my work. I am also grateful to the past members of the team: their work was instrumental in shaping my own.

CERTIFICATE

This is to certify that the thesis entitled *"Image Reconstruction and Comparison for Forensic Footwear Analysis"* and submitted by Gautham Venkatasubramanian ID No. 2014B4A70637P in partial fulfillment of the requirements of BITS F422T thesis embodies the work done by him under my supervision.

_____

Dr. Martin Herman
Senior Advisor for Forensics and IT
National Institute of Standards and Technology
Gaithersburg, Maryland USA

Date :

ABSTRACT

We will review current approaches to analysis of footwear impression evidence in the context of one-to-one comparison and unique identification.

Following this, we will study the accuracy of approximation techniques for image reconstruction, from traditional techniques like image momentsto modern methods such as convolutional autoencdoers, and consider their feasibility for producing scores for one-to-one shoeprint comparison.

Contents

Introduction

Usage of Shoeprints in Forensics

Footwear impressions are readily found in crime scenes, and developments in forensic evidence have allowed for the recording of such impressions as images. These impressions are used by investigators in apprehending suspects, and also as evidence in a court of law. As there are a wide variety of footwear models, it is necessary to have reliable methods of finding the type (brand, model etc.) of shoe which can produce such an impression.

Once the type of shoe has been identified and suspect(s) have been discovered, it is necessary to analyze further details to uniquely identify the exact shoe which could have produced a given impression. This unique identification is performed manually by an expert footwear examiner. The aim of the NIST Forensic Footwear team is to discover how automated methods can aid an investigator in the identification process.

Finding the model of shoe

Finding the model of shoe for a given impression can be described as a database retrieval problem: Given a large database of shoeprint impressions, one has to find efficient ways of retrieving a matching impression from the database for a given query impression(as in Figure 1). We summarize some of the developments in this field.

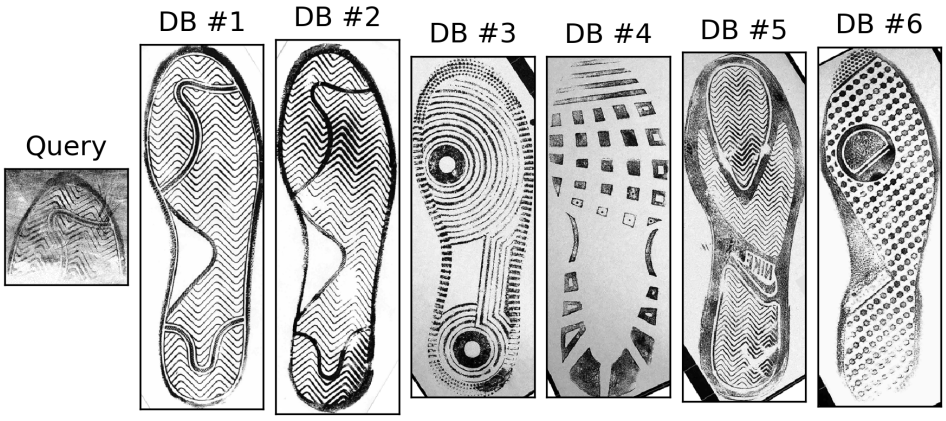

Figure 1: Database Retrieval

1995-2002: Early Forays

- The need for database retrieval of shoeprint images is identified; databases like REBEZO (Geradts & Keijzer, 1996) are developed for a systematic method of recording and retrieving shoeprint information.
- Various features such as points, bars, squiggles, and shapes are manually identified and entered into the database in the form of codes. These codes are used to classify the images, and retrieval is done via regular database queries.
- Attempts are made to automate the retrieval: various methods (Fourier transforms, Fractal decompositions (Bouridane, Alexander, Nibouche, & Crookes, 2000) etc.) are tried, but computational aspects hamper the feasibility.

2003-2010: Signal Processing methods

- As various methods for comparing images are discovered, some of these are attempted on shoeprints. A simple description of the retrieval procedure would be as follows:
  1. Transform the queried impression into a feature space via a suitable set of functions (i.e. a filterbank)
  2. Compare it to database images in the feature space and assign a similarity score
  3. Retrieve from the database the image that has the best score when compared to the query impression.
- For assigning the feature space, many different functions were used: Radon transforms (Patil, Deshmukh, & Kulkarni, 2012), Gabor wavelets (Patil & Kulkarni, 2009), Fourier-Mellin (Gueham, Bouridane, Crookes, & Nibouche, 2008) transforms et cetera. For the comparison, the transformed images were either compared directly (i.e. using normalized cross correlation) or via their image moments.

2010-present: Machine Learning and Computer Vision

- Advances in machine learning enabled more flexible methods of feature extraction from shoeprints. This in turn resulted in retrieval methods of high accuracy for a given database, as one could now train an algorithm to accurately retrieve shoeprint images.

- Object recognition models and other discoveries in deep learning are being attempted on shoeprints: most recently, parts of ResNet-50 (Kong, Ramanan, & Fowlkes, 2017) were used to extract features and compare images.

Obtaining one-to-one similarity scores

While the above discoveries are extremely accurate at retrieving a given shoe from a database, it is not known if the exact method(s) used to identify a model from a database can be used to distinguish between two shoes of the same model.

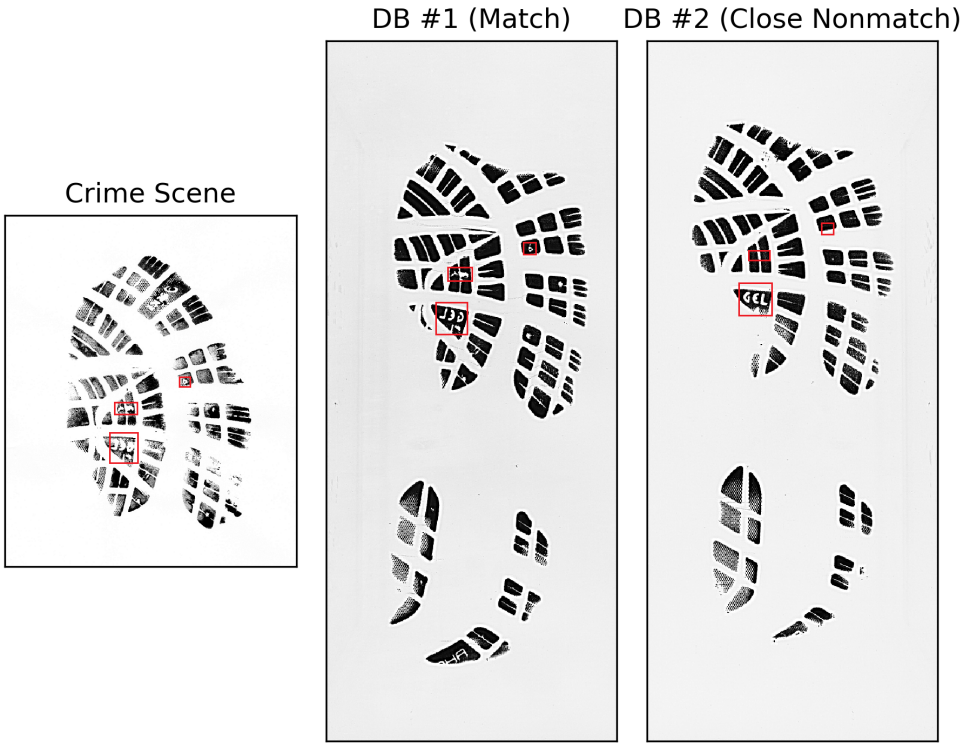

Figure 2: Notice differences in the rectangular boxes

Consider the example in Figure 2: We can see that the third image is not as good a match to the crime scene as compared to the second, but it is not known how such a distinction is expressed quantitatively during database retrieval.

However, if we consider the crime-scene image to be a poor reconstruction of the other two, we will immediately notice that the features in the boxes marked do not correspond for the wrong pair. My aim is to study possible techniques to perform image reconstruction and test if they can be used for analysis.

Reconstruction and Analysis using Image Moments

We study the basic concepts of constructing moments from images. (law Pawlak, 2006) was used as a primary reference.

Computational Aspects

From (Bruna & Mallat, 2013), we see that image moments are vulnerable to global nonrigid transformations. We attempt to mitigate this factor in two ways:

- Performing global alignment and computing moments on the overlapping areas (middle image) as in Figure 3

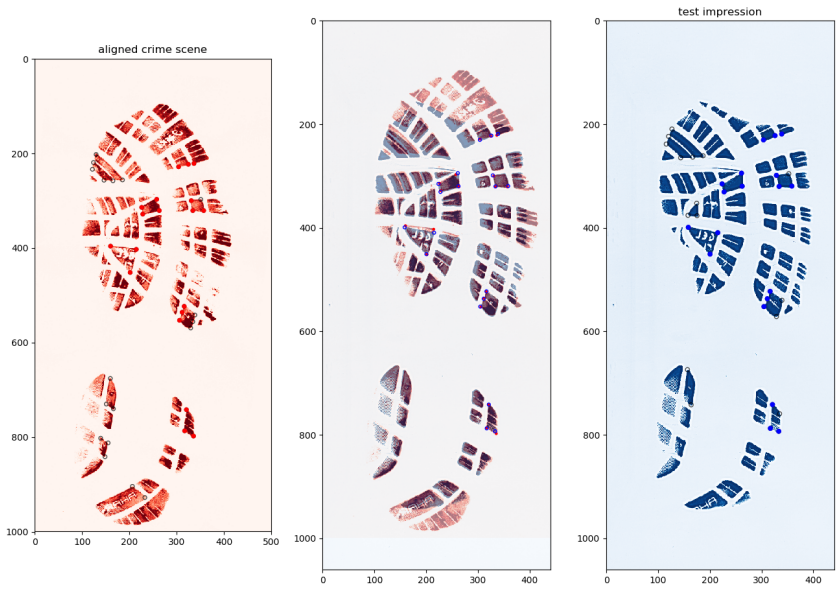

Figure 3: Only the overlapping regions are considered for similarity

- Splitting up the images into 50x50 blocks and performing the relevant calculations for each block: let Φ be a mapping from image space to

moment space, and let I_1 be an. Thus, we reconstruct the image by reconstructing each block individually.

Let $\Omega = [-1, 1] \subset \mathbb{R}^2$ denote the image plane, and let the image be a function $f(x, y) : \Omega \to \mathbb{R}$. f is considered to a function of *bounded variation*, i.e.

$$TV(f) = \iint_\Omega \left|\frac{\partial f(x,y)}{\partial x}\right| + \left|\frac{\partial f(x,y)}{\partial y}\right| dx\, dy < \infty$$

Now we can define the moments of the function f.

Geometric Moments

The $(p, q)^{th}$ geometric moment of f is defined as:

$$m_{p,q} = \int \int_\Omega x^p y^q f(x, y) dx dy$$

The set of geometric moments upto order N are :

$$\{m_{p,q} : 0 \leq p + q \leq N\}$$

We note that for geometric moments, it is not possible to reconstruct the function f given the geometric moments upto order N; from (Talenti, 1987), we see that the mapping from moment space to (just 1-D) function space is an ill-posed problem, replacing $m_{p,q}$ with $m_{p,q} + \epsilon$ may map to a function f_ϵ that is arbitrarily far from f .

Hu Moments

Central moments are geometric moments taken with a point $(\bar{x}, \bar{y})$ as the center of the image:

$$\mu_{p,q} = \int \int_\Omega (x - \bar{x})^p (y - \bar{y})^q f(x, y) dx dy$$

where $\bar{x} = \frac{m_{1,0}}{m_{0,0}}$ and $\bar{y} = \frac{m_{0,1}}{m_{0,0}}$. Central moments are invariant to translation.

Normalized central moments can be obtained from central moments as follows:

$$\nu_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{1+\frac{(p+q)}{2}}}$$

The seven **Hu moments** can be constructed from the normalized central moments as follows:

$$\Psi_1 = \nu_{2,0} + \nu_{0,2}$$
$$\Psi_2 = (\nu_{2,0} - \nu_{0,2})^2 + 4\nu_{1,1}^2$$
$$\Psi_3 = (\nu_{3,0} - 3\nu_{1,2})^2 + (3\nu_{2,1} - \nu_{0,3})^2$$
$$\Psi_4 = (\nu_{3,0} + \nu_{1,2})^2 + (\nu_{2,1} + \nu_{0,3})^2$$

$$\Psi_5 = (\nu_{3,0} - 3\nu_{1,2})(\nu_{3,0} + \nu_{1,2})[(\nu_{3,0} + \nu_{1,2})^2 - 3(\nu_{2,1} + \nu_{0,3})^2] - (\nu_{0,3} - 3\nu_{2,1})(\nu_{0,3} + \nu_{2,1})[3(\nu_{3,0} + \nu_{1,2})^2 - (\nu_{2,1} + \nu_{0,3})^2] \quad (1)$$

$$\Psi_6 = (\nu_{2,0} - \nu_{0,2})[(\nu_{3,0} + \nu_{1,2})^2 - (\nu_{2,1} + \nu_{0,3})^2] + 4\nu_{1,1}(\nu_{3,0} + \nu_{1,2})(\nu_{2,1} + \nu_{0,3}) \quad (2)$$

$$\Psi_7 = (3\nu_{2,1} - \nu_{0,3})(\nu_{3,0} + \nu_{1,2})[(\nu_{3,0} + \nu_{1,2})^2 - 3(\nu_{2,1} + \nu_{0,3})^2] - (\nu_{3,0} - 3\nu_{1,2})(\nu_{0,3} + \nu_{2,1})[3(\nu_{3,0} + \nu_{1,2})^2 - (\nu_{2,1} + \nu_{0,3})^2] \quad (3)$$

Orthogonal Polynomial Moments

Instead of using geometric moments, we can use a system of 2D orthogonal polynomials to construct moments.

The $(p, q)^{th}$ moment of f is defined as:

$$\lambda_{p,q} = \int \int_{\Omega} V_{p,q}(x, y) f(x, y) w(x, y) dx dy$$

Here $V_{p,q}$ belongs to a set of 2D orthogonal polynomials, and w is the corresponding weight function for that set. We construct a set of 2D orthogonal polynomials $V_{p,q}$ from sets of 1D orthogonal polynomials P_p, Q_q:

$$V_{p,q}(x, y) = P_p(x) Q_q(y)$$

One of the classical orthogonal polynomials in one dimension are **Legendre Polynomials**, which are the set of polynomials defined by the following recurrence relation:

$$P_{n+1}(x) = \frac{2n+1}{n+1} x P_n(x) - \frac{n}{n+1} P_{n-1}(x); \quad P_0(x) = 1, P_1(x) = x$$

With weight function $w(x) = 1$, and $\Omega = [-1, 1]$.

Generalized 1D orthogonal polynomials: we have the class of **Gegenbauer** or **ultraspherical polynomials** with a parameter γ that allows for scaling. The recurrence relation is as follows:

$$G_{n+1}(x, \gamma) = 2\frac{(n+\gamma)}{(n+1)} x G_n(x, \gamma) - \frac{(n+2\gamma-1)}{(n+1)} G_{n-1}(x, \gamma)$$

With weight function $w(x, \gamma) = (1-x^2)^{\gamma-\frac{1}{2}}$, $x \in [-1, 1]$, and $\gamma > -\frac{1}{2}$. High values of γ lead to capture of more localized features in the image.

Legendre Polynomials are a subset of Gegenbauer polynomials: when $\gamma = \frac{1}{2}$, the recurrence relation reduces to that of Legendre Polynomials.

Unlike geometric moments, the image $f(x, y)$ can be approximated in a straightforward manner using the orthogonal moments $\lambda_{p,q}$ as below:

$$f(x, y) \approxeq f_N(x, y) = \sum_{p=0}^{N} \sum_{q=0}^{p} \lambda_{p-q,q} \tau p - q, q V_{p-q,q}(x, y)$$

Where $\tau_{p,q}$ is a normalizing factor for $V_{p,q}(x, y)$.

We approximate $\lambda_{p,q}$ as follows:

$$\hat{\lambda_{p,q}} = \sum_i \sum_j h_{p,q}(x_i, y_j) f(x_i, y_j)$$

Where f is assumed to be constant over the pixel (x_i, y_j) and

$$h_{p,q}(x_i, y_j) = \int_{x_i - \frac{\Delta}{2}}^{x_i + \frac{\Delta}{2}} \int_{y_j - \frac{\Delta}{2}}^{y_j + \frac{\Delta}{2}} V_{p,q}(x, y)\, dx\, dy$$

is the integral over the pixel. We approximate the integral by Simpson's $1/3^{rd}$ rule, using 23 points between $x_i - \frac{\Delta}{2}$ and $x_i + \frac{\Delta}{2}$.

Due to issues of floating point overflow, we currently only compute moments upto order 25.

Sample Reconstruction

We view the reconstructions of the sample image with Gegenbauer polynomials for various values of γ below. Though the reconstruction does improve as the order increases, we see that moments upto order $N = 25$ are not enough to distinguish detailed features on the image. We also see that the weight function $w(x, y)$ tends to skew the values as the polynomial reaches the boundary.

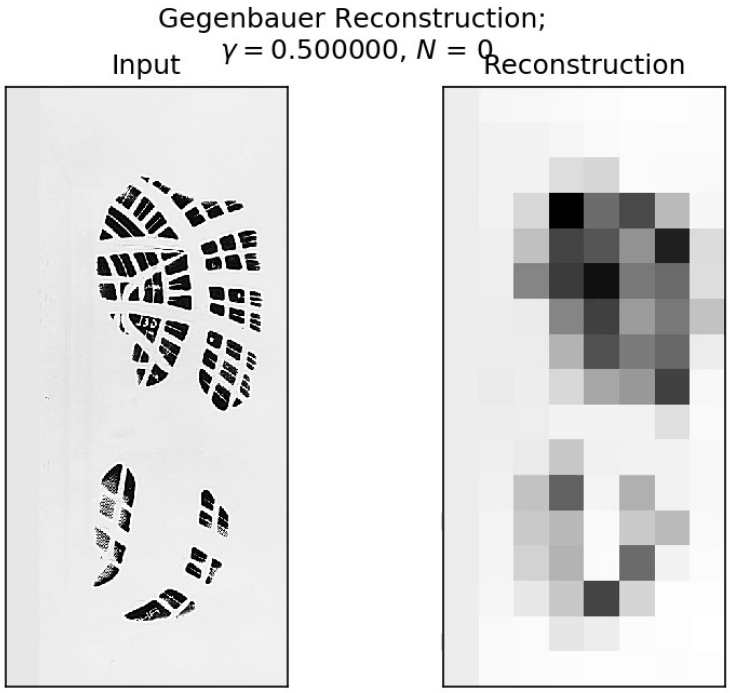

Figure 4: Reconstruction $\gamma = 0.5$, upto moments of order 25

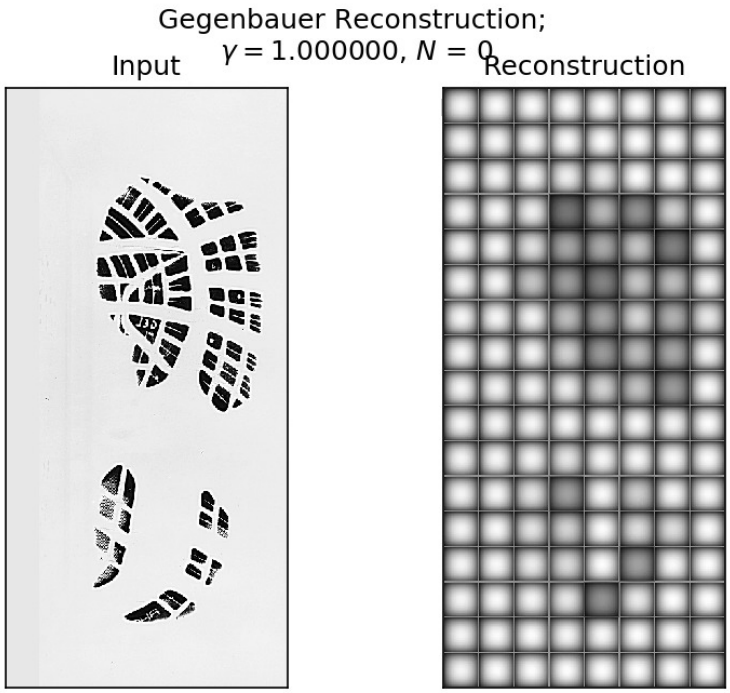

Figure 5: Reconstruction $\gamma = 1.0$, upto moments of order 25

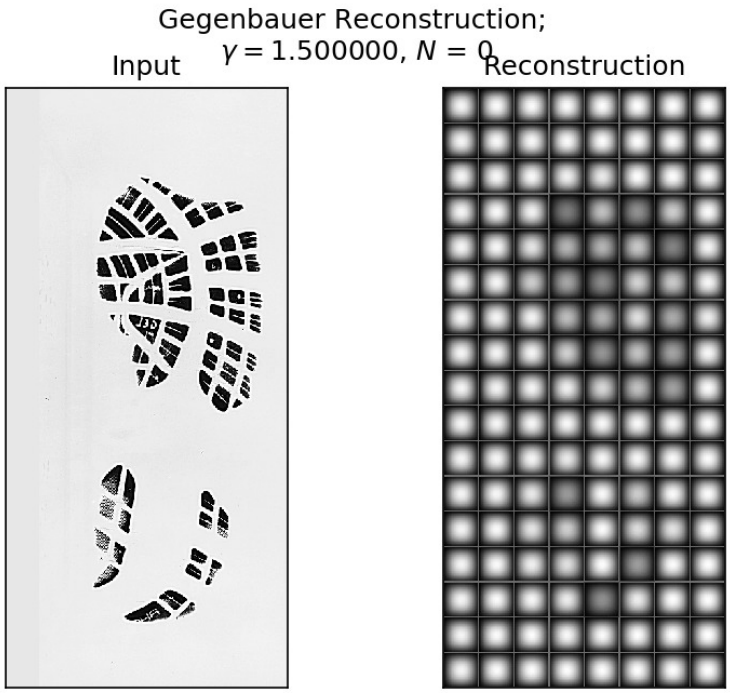

Figure 6: Reconstruction $\gamma = 1.5$, upto moments of order 25

Reconstruction and Analysis with Kolmogorov Superpositions

The following theorem (henceforth described as *Kolmogorov's Superposition Theorem* or *KST*) was proved by Andrei Kolmogorov and his student Vladimir Arnold in 1957 ((Kolmogorov, 1957)), solving one of David Hilbert's 23 problems formulated in 1900.

The 13th of Hilbert's problems contained the question of whether continuous functions of three variables could be represented as superpositions of continuous functions of two variables. KST solves this problem for the general case, whereby any continuous function of n variables can be represented by superpositions of continuous functions of one variable. We note here the statement of KST as described in (Lorentz, 1966).

Statement of the Theorem

Theorem 1
There exist n constants $0 < \lambda_p \leq 1$, $p = 1..n$, and $2n + 1$ functions defined on $I = [0, 1]$ with values in I, such that:

- ϕ_q are all continuous, strictly increasing, belonging to the class $Lip\ \alpha, \alpha > 0$
- For each continuous function f on $I^n = [0, 1]^n$, one can find a continuous function $g(u)$, $u \in [0, n]$ such that

$$f(x_1, x_2, ...x_n) = \sum_{q=0}^{2n} g(\sum_{p=1}^{n} \lambda_p \phi_q(x_p)) \tag{4}$$

- The functions ϕ_q depend only on n and are *independent of the choice of* f.

While the proof in (Kolmogorov, 1957) used the below equality,

$$f(x_1, x_2...x_n) = \sum_{q=0}^{2n} g_q(\sum_{p=1}^{n} \phi_{pq}(x_p)) \tag{5}$$

with $2n^2 + n$ inner functions ϕ_{pq} and $2n + 1$ outer functions g_q, (Sprecher, 1965) showed that just one outer function g, and expressing $\phi_{pq} = \lambda_p \phi_q$ would

be enough to prove the above result. Later, in (Sprecher, 1996), we see that the $2n+1$ inner functions ϕ_q can be replaced with just one function Ψ, albeit with slight modifications to get the same result.

Computing the Functions

The result proved by KST is powerful: it shows that we need to compute the $2n+1$ inner functions ϕ_q only once, and we can then use them to represent any n-dimensional continuous function f, as long as we find an appropriate function g every time. Therefore, if we consider an image to be a two-dimensional function $f(x,y)0 \leq x,y \leq 1$, we can compute an *approximation* with a suitable function g, and the pre-computed inner functions $\phi_0 ... \phi_4$. Hence, we would get:

$$f(x,y) \approxeq \sum_{q=0}^{4} g(\lambda_1 \phi_q(x) + \lambda_2 \phi_q(y)) \tag{6}$$

(Note that we can only compute an approximation of g upto r terms, as the actual proof involves an infinite sum to compute g, which converges to f.)

The Inner Functions ϕ_q

For the construction of the inner functions ϕ_q, we follow the procedure in (Sprecher, 1996), defining a function Ψ on $[0,1] \cap Q$ as follows:

1. Select an integer $\gamma \geq 2n+2$ as the base.
2. We compute $\Psi(x) \forall x$ defined as rational numbers in base γ. Since the rational numbers can have infinite digits, we truncate them at k digits, and consider the set $D = \{d_k\}$, the set of rational numbers in $[0,1]$ with upto k digits in base γ
3. $\forall d_k \in D$, we have $d_k = 0.i_1 i_2 i_3 .... i_k$, where i_r is a digit in base γ $(0 \leq i_r < \gamma,\ 1 \leq r \leq k)$
4. We define a function

   $$\beta(r) = 1 + n + n^2 ... n^{r-1} = \frac{n^r - 1}{n-1}$$

   where n is the target dimension, and $1 \leq r \leq k$

5. We define $\Psi(d_k)$ recursively over k as below :

$$\Psi(d_k = 0.i_1 i_2 ....i_k) = \begin{cases} i_k \gamma^{-1} & k = 1 \\ \sum_{r=1}^{k} i_r \gamma^{-\beta(r)} & k > 1,\ 0 \leq i_r < \gamma - 1 \forall r \\ 0.5(\Psi(d_k + \frac{1}{\gamma_k}) + \Psi(d_k - \frac{1}{\gamma^k})) & i_k = \gamma - 1, k > 1 \end{cases} \tag{7}$$

and $\Psi(x+1) = \Psi(x) + 1$.

6. Now, we compute the functions ϕ_q as

$$\phi_q(d_k) = \Psi(d_k + qa)$$

, where $a = \frac{1}{\gamma(\gamma-1)}$ and $0 \leq q \leq 4$

7. Since a cannot be expressed in the d_k form, we approximate it as follows:

$$\Psi(d_k + qa) = \Psi(d_k + \frac{q}{\gamma}(1 + \gamma^{-1} + \gamma^{-2}.....))$$

For an n dimensional function, we have $0 \leq q \leq 2n$, and as per our definition of γ, $q < \gamma\ \forall q$, hence, in base γ

$\Psi(d_k{+}qa) = \Psi(0.i_1 i_2 ....i_k{+}0.0qqqqqqqqqq.....) \cong \Psi(d_k{+}0.0qqqq..(\text{k digits})){+}q\epsilon_k$

This produces an error term $\epsilon_k = \sum_{k+1}^{\infty} \gamma^{-\beta(r)}$ in the computation of ϕ_q, but it diminishes as $k \to \infty$.

8. The coefficients λ_p are defined as follows:

$$\lambda_p = \begin{cases} 1 & p = 1 \\ \sum_{r=1}^{\infty} \gamma^{(1-p)\beta(r)} & p > 1 \end{cases} \tag{8}$$

9. Hence we have the inner part of the superposition to approximate f.

Given in Figure 7 is a plot of the function Ψ on $[0, 1]$

We see that Ψ is mostly flat, and also has a fractal-like property.

The Outer Function g

Computing the outer function g is slightly more complex, as it requires iterating towards an accurate approximation and some form of a lookup every

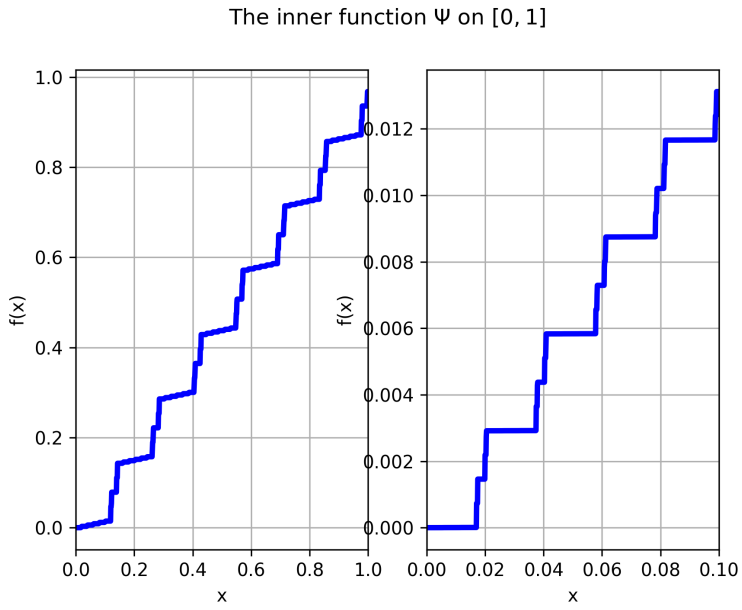

Figure 7: The function Ψ on [0,1] and [0, 0.1]. Notice the similarity of the function across scales

iteration; indeed, there have been methods for approximating g, such as (Sprecher, 1997), (Köppen, 2002) and (Igelnik & Parikh, 2003), and analyzing the properties of ϕ_q and g together is an active field research today. It is worth noting that the structure of neural networks have been drawn upon to describe the process of approximating the functions required for KST.

We shall be using the technique from (Sprecher, 1997) and (Köppen, 2002):

1. We have
$$f(x_1, x_2, ...x_n) = \sum_{q=0}^{2n} g(\sum_{p=1}^{n} \lambda_p \phi_q(x_p))$$

2. From Step 7 above, we see that

   $\Psi(d_k{+}qa) = \Psi(0.i_1i_2....i_k{+}0.0qqqqqqqqq.....) \approxeq \Psi(d_k{+}0.0qqqq..(\text{k digits})){+}q\epsilon_k$

   where $\epsilon_k \to 0$ as $k \to \infty$,

3. We define $d_k^q = d_k + 0.0qqqq..(\text{k digits})$ and a cumulative error term

$b_k = \sum_{p=1}^{n} \lambda_p \epsilon_k$:

$$\sum_{p=1}^{n} \lambda_p \phi_q(d_k) = \sum_{p=1}^{n} \lambda_p \Psi(d_k + qa)$$

$$\approxeq \sum_{p=1}^{n} \lambda_p (\Psi(d_k^q) + q\epsilon_k)$$

$$= \sum_{p=1}^{n} \lambda_p \Psi(d_k^q) + qb_k \tag{9}$$

4. Therefore for $\mathbf{d}_k \in D^n = D \times D \times D ... n$ times , we have:

$$f(\mathbf{d}_k) = \sum_{q=0}^{2n} g(\sum_{p=1}^{n} \lambda_p \Psi(\mathbf{d}_k^q)) \text{ as } k \to \infty$$

$$\approxeq \sum_{q=0}^{2n} g(\sum_{p=1}^{n} \lambda_p \Psi(\mathbf{d}_k^q)) \tag{10}$$

5. We define the transfer function $\xi_q(\mathbf{x})$ as

$$\xi_q(\mathbf{d}_k) = \sum_{p=1}^{n} \lambda_p \Psi(\mathbf{d}_k^q)$$

6. We define functions θ around each $\mathbf{d}_k^q$ to cover the domain of g

$$\theta(\mathbf{d}_k^q; y_q) = \sigma(\gamma^{\beta(k+1)}(y_q - \xi_q(\mathbf{d}_k)) + 1)$$
$$-\sigma(\gamma^{\beta(k+1)}(y_q - \xi_q(\mathbf{d}_k) - (\gamma - 2)b_k)) \tag{11}$$

where:

$$\sigma(x) = \begin{cases} 0 & x \leq 0 \\ \text{an } \textit{arbitrary continuous function on } [0,1] & x \in [0,1] \\ 1 & x \geq 1 \end{cases} \tag{12}$$

7. We now define $g = \sum g_r$, where

$$f_0 = f(x_1 ... x_n)$$

$$g_r = \frac{1}{2n+1} \sum_{\mathbf{d}_k^q} f_{r-1}(\mathbf{d}_k)\theta(\mathbf{d}_k^q; \xi_q(\mathbf{x}))$$

$$f_r = f_{r-1} - g_r$$

Given below is a plot of the functions ξ_q:

The transfer functions ξ_q on $[0, 1]^2$

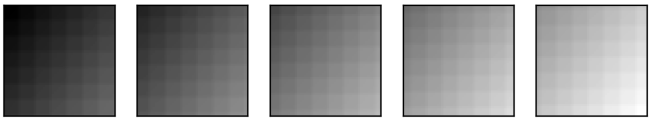

Figure 8: The functions Φ_q built from Ψ. They preserve the blocky structure of Ψ in 2D

Sample Reconstruction

We view the reconstruction of the sample image using the inner and outer functions defined earlier. We consider rational numbers in base 7 ($2n + 2 = 6$) and use $k = 1$,2 for our sample image. We iterate the function g for 10 iterations with k=1 and 10 iterations with k=2; the result is shown in Figure 9.

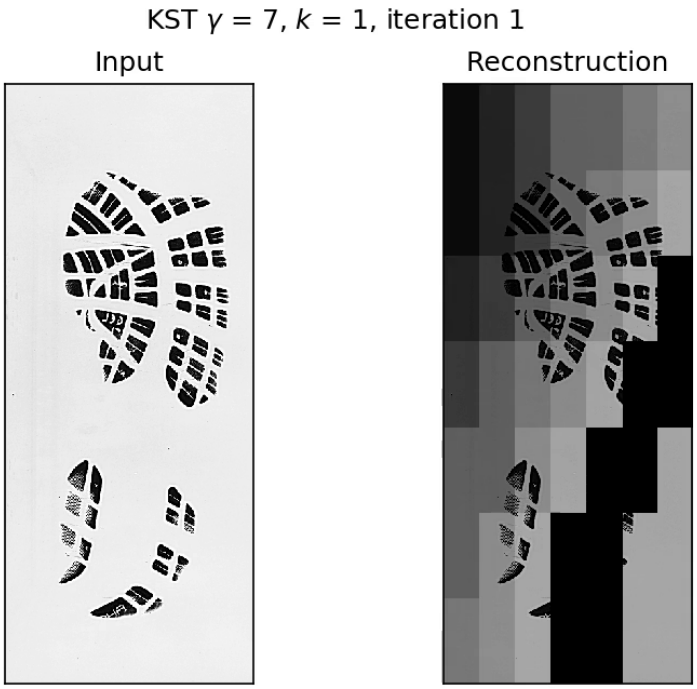

Figure 9: Reconstruction using Kolmogorov superpositions.

We see that the reconstruction steadily becomes better across the iterations with a noticable jump when $k = 2$; this is because the number of points considered in Ψ have been increased, the domain of the approximation function is denser, and hence we are able to cover more regions. It should be noted that, since every iteration of the function g is a lookup, we have to store the all the values computed for g in order to approximate the image. This turns out to be a large amount when compared to the other methods.

Reconstruction and Analysis with Neural Networks

In (Sprecher, 1996) we saw that the procedure for approximating via KSTs took a structure similar to a small neural network. Indeed, we observe that neural networks, due to their composable nature and computational ease of use, have gained widespread acclaim for enabling computers to perform human-level tasks such as image classification with great ease (see (Raghu, Poole, Kleinberg, Ganguli, & Dickstein, 2017)). All the progress from neural networks would not have been possible without the theoretical foundation provided by (Cybenko, 1989), who proved to the following theorem:
Theorem 2 (Univeral Approximation Theorem)
Let σ be any continuous sigmoidal discriminatory function on $[0, 1]$ i.e.

$$\sigma(x) \to \begin{cases} 1 & x \to \infty \\ 0 & x \to -\infty \end{cases}$$

and, if for a measure μ on $[0, 1]^n$

$$\int_{I_n} \sigma(y^T x + \theta) d\mu x = 0 \forall y \in \mathbb{R}^n, \forall \theta \in \mathbb{R} \implies \mu = 0 \tag{13}$$

Then, finite sums of the form

$$G(x) = \sum_{j=1} N\sigma(y_j^T x + \theta_j)$$

are dense in $[0, 1]^n$.

The above result was extended in (Hornik, 1991) to networks with Rectified Linear Unit (ReLU) activation functions as well.

Convolutional Neural Networks (CNNs) and Autoencoders

The networks used in the above results took linear inputs, which, when applied on high-dimensional data such as images, resulted in highly dense networks that were difficult to train, store, and use. Convolutional neural networks reduce the number of weights required for these kinds of data, by *convolving* the same kernel of weights over the input. This allows for faster training of the network, and also provides some robustness against translation. (Krizhevsky,

Sutskever, & Hinton, 2012) used CNNs to great success in image classification, and now CNNs have become ubiquitous in the field of computer vision.

Autoencoders extend the concept of principal component analysis by the adding further layers to the network; the aim is to obtain a robust lower-dimensional representation of a class of data (an *encoding*) that can later be decoded to accurately reproduce the input provided (see (Bengio, Lamblin, Popovici, & Larochelle, 2007)). Convolutional Autoencoders use convolutions to downsize the data, thereby ensuring that only the important components are encoded. Like all other networks, autoencoders must be trained: the loss function usually involves a difference computed between the original and the reconstructed image.

Sample Reconstruction

Now, we view the reconstructions for some basic autoencoder networks. We use an 80-20 split of the data from (Kortylewski, Albrecht, & Vetter, 2015) for training our autoencoders (80 percent for training, 20 percent for testing). We train the networks in batches for 50 images, use cross-correlation as the optimization function, optimize with the Adam method and view the reconstructions of test samples below. Note that the autoencoder contains only convolutional layers, and not linear layers due to high dimensionality of weights. Each network was trained for 500 epochs on the training set. A small amount of noise was added to test accuracy.

2-Layer Encoder-Decoder

In Figure 10 and Figure 11, we see some reconstructions from an autoencoder with 2 encoding and 2 decoding layers. The reconstructions for these images are sharp, but some of the finer details are still missing (as seen in the fourth picture).

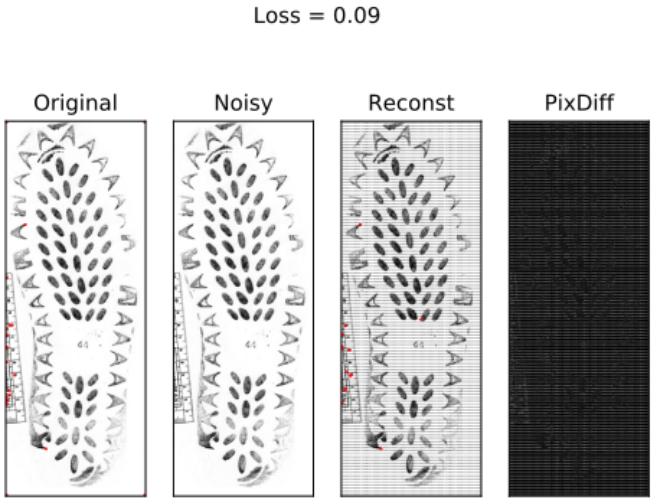

Figure 10: Pixel-level differences are visible

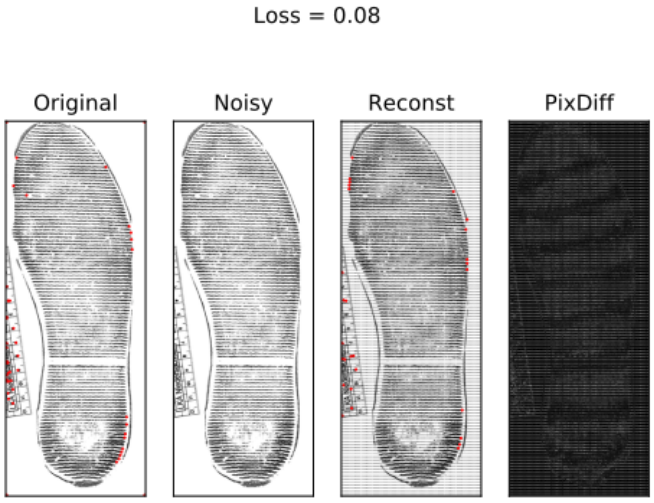

Figure 11:

4-Layer Encoder-Decoder

In Figure 12 and Figure 13, we see some reconstructions from an autoencoder with 4 encoding and 4 decoding layers. The reconstructions for these images are not as good, because the network was trained for a shorter time, despite being deeper than the previous one. This means that some of the weights have not been trained optimally, and so the network is unable to reconstruct accurately.

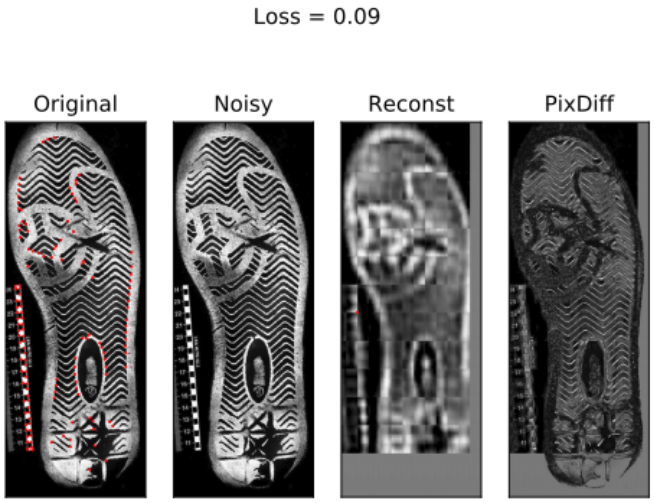

Figure 12: Lack of optimum training leads to blurry images and lost data

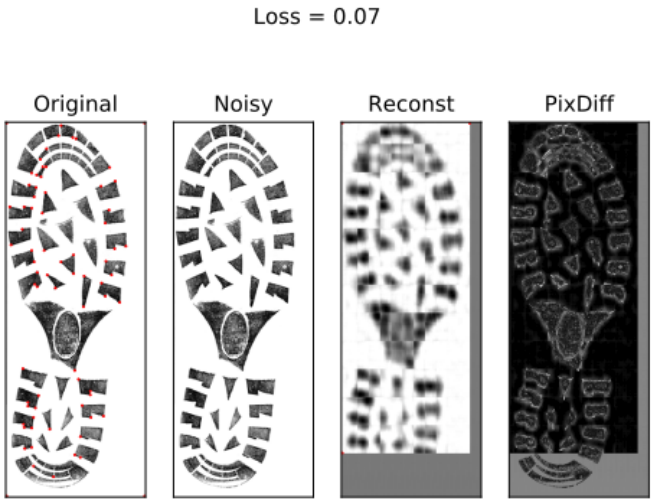

Figure 13:

Conclusion

We attempted the reconstruction of shoeprint images via the following techniques; each of which differed in terms of conceptual structure and implementation details.

- Orthogonal Image Moments:
  - Approximate the image function using moments from classical orthogonal polynomials
  - Uses concepts of moments of a function
  - Reconstruction accuracy $\to 1$ as k, the order of moments used $\to \infty$
  - Does not require large amounts of input data for one-to-one use
  - Requires high precision, especially closer to the boundary of the image, due to infinitesimal weights
  - Discretization error is present due to using continuous polynomials

on a discrete image

  - Brittleness of global moments means more outputs need to stored per image. For an image of dimensions $H \times W$, we might need to store $k \times \left\lceil \frac{H}{50} \right\rceil \times \left\lceil \frac{W}{50} \right\rceil$ values, where k is the order of moments used

- Kolmogorov's Superposition Theorem:
  - Approximate the image function using superpositions arbitrary continuous one-dimensional inner function Ψ, and outer function g
  - Requires very high precision, but does not generate large values
  - Reconstruction accuracy is iterative; converges as number of iterations and accuracy $\to \infty$
  - Ψ needs to be computed only once per accuracy level (can be reused and stored)
  - Does not require large amounts of input data for one-to-one use
  - Discretization error is not as heavy since the image is approximated using a large number of rational points
  - Requires lot of storage space; the function g needs to computed as a lookup table for every image: for an image of dimensions $H \times W$ we might need to store $r \times H \times W$ values, one for each pixel over every iteration.
  - Reconstruction is blocky but not as blurry as others
- Convolutional Autoencoders
  - Approximate *a class of image functions* by reconstructing images with abstract representations (edges, blobs, corners etc.) learned a given dataset
  - Use concepts from principal component analysis and optimization theory
  - Convergence is stoachastic, lends itself to computationally friendly implementations
  - Requires a large representative training set to be useful in comparison

- Flexible in terms of storage and construction
- Tradeoff between accuracy and speed is available (shallow networks are easier to trains

However, we observe a common limitation from the sample reconstructions:

1. A large amount of data needs to be stored for reconstructions to be accurate (moments for large images computed patch-by-patch; KST Outer Function for KST requires lookup from a large table; neural networks require a representative training dataset), and
2. Intensive and high-precision computation is required to approximate fine details (higher degree moments; Inner Function for KST needs to be computed for many points in $[0, 1] \cap \mathbf{Q}$, Outer Function from KST has an iterative convergence; training time is longer for deeper networks)

This leads to the question of finding computationally feasible methods that can extend to higher approximations: while deep learning looks to be the method of choice for such problems, it remains to be seen how the limitations of data and precision affect their capabilities.

References

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems* (pp. 153–160).

Bouridane, A., Alexander, A., Nibouche, M., & Crookes, D. (2000). Application of fractals to the detection and classification of shoeprints. In *Image processing, 2000. proceedings. 2000 international conference on* (Vol. 1, pp. 474–477).

Bruna, J., & Mallat, S. (2013). Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1872–1886.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303–314.

Geradts, Z., & Keijzer, J. (1996). The image-database rebezo for shoeprints with developments on automatic classification of shoe outsole designs. *Forensic Science International*, *82*(1), 21–31.

Gueham, M., Bouridane, A., Crookes, D., & Nibouche, O. (2008). Automatic recognition of shoeprints using fourier-mellin transform. In *Adaptive hardware and systems, 2008. ahs'08. nasa/esa conference on* (pp. 487–491).

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, *4*(2), 251–257.

Igelnik, B., & Parikh, N. (2003). Kolmogorov's spline network. *IEEE transactions on neural networks*, *14*(4), 725–733.

Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Doklady akademii nauk* (Vol. 114, pp. 953–956).

Kong, B., Ramanan, D., & Fowlkes, C. (2017). Cross-domain forensic shoeprint matching. In *British machine vision conference (bmvc).*

Köppen, M. (2002). On the training of a kolmogorov network. In *International conference on artificial neural networks* (pp. 474–479).

Kortylewski, A., Albrecht, T., & Vetter, T. (2015). Unsupervised footwear impression analysis and retrieval from crime scene data. In *Computer vision - accv 2014 workshops* (pp. 644–658). Springer International Publishing.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural*

information processing systems (pp. 1097–1105).

law Pawlak, M. (2006). Image analysis by moments: reconstruction and computational aspects. *Oficyna Wydawnicza Politechniki Wrocławskiej*.

Lorentz, G. (1966). *Approximation of functions, athena series*. Holt, Rinehart and Winston, New York.

Patil, P. M., Deshmukh, M. P., & Kulkarni, J. V. (2012). Investigation of shoeprints using radon transform with reduced computational complexity. *Journal of Pattern Recognition Research*, *7*, 80–89.

Patil, P. M., & Kulkarni, J. V. (2009). Rotation and intensity invariant shoeprint matching using gabor transform with application to forensic science. *Pattern Recognition*, *42*(7), 1308–1317.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., & Dickstein, J. S. (2017). On the expressive power of deep neural networks. In *Proceedings of the 34th international conference on machine learning-volume 70* (pp. 2847–2854).

Sprecher, D. A. (1965). On the structure of continuous functions of several variables. *Transactions of the American Mathematical Society*, *115*, 340–355.

Sprecher, D. A. (1996). A numerical implementation of kolmogorov's superpositions. *Neural Networks*, *9*(5), 765–772.

Sprecher, D. A. (1997). A numerical implementation of kolmogorov's superpositions ii. *Neural Networks*, *10*(3), 447–457.

Talenti, G. (1987, aug). Recovering a function from a finite number of moments. *Inverse Problems*, *3*(3), 501–517. Retrieved from `https://doi.org/10.1088%2F0266-5611%2F3%2F3%2F016` doi: 10.1088/0266-5611/3/3/016