This is the title of a thesis submitted to Iowa State University

Note that only the first letter of the first word and proper names are capitalized

by

Miranda Tilton

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Statistics

Program of Study Committee:
Susan Vanderplas, Co-major Professor
Danica Ommen, Co-major Professor
Kris De Brabanter

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation/thesis. The Graduate College will ensure this dissertation/thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2019

Copyright © Miranda Tilton, 2019. All rights reserved.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This is the text of my abstract that is part of the thesis itself. The abstract describes the work in general and the heading and style match the rest of the document.

CHAPTER 1. INTRODUCTION

1.1 Motivation

In forensic science, shoe prints and outsole characteristics fall into the category of pattern evidence. When a shoe print or impression is found at a crime scene, the investigator may ask a series of questions. Initially, it may be important to determine the make and model of the shoe, which may help detectives locate comparison shoes from suspects. Later in the investigation, the forensic examiner may consider individualizing characteristics found in the print; that is, small defects that make it possible to tie a specific shoe to the print left at the scene. In cases where such individualizing characteristics are not considered (estimated at 95% of cases in the United States according to some experts[1]), it is important to be able to assess the probability that the specific model of shoe which made the print would be found in the suspect's posessions. This question is much more difficult than identifying the make and model of the shoe, because it requires that the forensic examiner have access to a database containing information about the frequency of shoes in the local population, where the local population itself may be difficult to define. Any tractable solution to the problem of assessing the random match probability of a shoeprint based only on class characteristics (**?**) (make, model, and other characteristics determined during the manufacturing process) requires a way to assemble this database: an automated solution to efficiently classify many types of shoes within a common system. This project is designed to address the computational and statistical process of assembling statistical features which can be used to assess similarity beteween two or more images.

1.2 Background

1.2.1 Outsole Class Characteristics

[1] Leslie Hammer, presentation to CSAFE on March 5, 2018

This paragraph is going to be a project for me on its own. I'll overload quickly if I try to go through all the sources at once, so for now I'm going to collect important bits from the textbook and continue from there another day.

Class characteristics are defined as the set of features which allow an object to be placed into a group with other physically-similar objects. In the context of footwear, the term refers to the design and physical dimension of the shoe, particularly with regard to the shoe's outsole. While class characteristics are not sufficient for identification, or matching an imprint from a crime scene to a specific shoe, they in many cases enable the exclusion of footwear (**?**). I forget APA guidelines for multiple ideas coming from the same source. If multiple sentences come from the same source, is it cite at the end of them?

?

- Size: MANY metrics for size, not consistent across manufacturers/countries, many conversion charts differ widely, so not an easy or consistent metric. Also different shoe styles with same size inside may be differently sized outside
- Size difficult to determine from crime scene if poor images/impression records
- Model: If a shoe is popular, can have multiple molds in same size that may all have slight differences
- Make/model: (Ch 12) Counterfeit designs and look-alikes, shoe models constantly changing, old ones coming back -¿ make and model are difficult

1.2.2 Image Analysis

Question: I looked into these, but I'm still not quite sure how feature detection isn't just convolutional filters.

Methods that are worth briefly describing: (links to wiki, but you can use Computer Vision textbook to get a citeable reference)

- Low level feature detectors - edge, corner, ridge, blob

- Hough transforms

Gist: these methods 1) work at a very low level, 2) produce features that aren't "global" - corners are 3x3 pixel region corners, not quadrilateral corners, 3) are computationally intensive, and 4) are very fragile - the parameters used break with lighting or color changes. Even if we did use them, we'd need many different random forest models to handle aggregating low-level features into things like quadrilaterals, lines, etc., and those models would be as fragile as the methods producing the input data.

Additional paragraph: Why CNNs are a good option - they've taken over image recognition, they're fast and work at scale on new images, they produce results that are interpretable, and they match human visual structure architecture, so the features they pick out should match human-labeled features (unlike low-level CV methods).

1.2.3 CNN Theory

Classification

- Visual classification (i.e., assinging a label to an object based on features that can be seen) is a complex task that humans do very well.
- Sight is our dominant sense and a significant part of our brain is dedicated to vision –¿ our visual process is refined and skilled (cite somehow?). You are 100% encouraged to use the xkcd comic here to talk about how hard it is to emulate this with a computer
- (Describe the human visual/classification process: feature detection, routing to the brain, and label application) Use Sensation & Perception (Goldstein) heavily here. You might also use some of the computer vision books to compare/contrast.
- Want to make a point about how subtle some differences can be. Can use the caterpillar/carrot example with "fuzzy texture" difference, or use corgi/fox and say that the differences are in the presence of the tongue, curve of the body, size of the mouth.

Convolutional Neural Networks (CNNs)

While our brains are adept at parsing images and classifying the objects within them, the task has proved much more difficult for computers. Computer vision was thought to be easy in 1966 when a researcher at MIT believed that teaching a computer to separate picture regions into objects and background regions could be completed as a summer project (**?**). The task proved much more difficult than expected, and remained difficult for decades. Now, CNNs are a widely implemented method for automated image recognition and perform comparably to humans on certain tasks. The ImageNet Large Scale Visual Recognition Competition (ILSVRC)is a widely followed contest to produce the best algorithm for image classification; since 2014, it has been dominated by convolutional neural networks (**?**).

Convolutional neural networks (CNNs) are a tool for supervised deep-learning that have become standard in recent years for automatic image classification. CNNs are a form of artificial neural network, which were inspired by biological processes in the brain (**?**). CNNs primarily use combinations of convolutional and pooling layers to filter raw information into features. These features are then fed into densely connected layers which are trained to associate given sets of features with their desired labels. This translation-invariant automated classification closely mimics the human eye-to-brain classification process.

Filters and Convolution Convolutional Neural Networks are named to highlight their use of convolution to extract information from an image. To a computer, an image is stored as a 3-dimensional array with a length and width corresponding to its number of pixels and a depth of 3 to represent the typical RBG color channels.

Let x be an image represented as a numerical matrix, indexed by i, j, and β be a filter of dimension $(2a+1) \times (2b+1)$

The convolution of image x and filter β is

$$(\beta * x)(i,j) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} \beta(s,t) x(i-s, j-t)$$

What's the balance between describing with math and using plain English? I like the plainspeak, but maybe it's not so necessary? Use math, describe in plain English, then add pictures to be sure. The goal is to communicate. A single convolutional filter is a small array (say 5x5x3) of real valued weights that represents some feature of the image. When a filter is applied to a portion of the imagea single value is returned that is associated with the presence of the feature for a given subsection of the input image. When applied over an entire image, the resulting matrix of values maps the strength of the feature across the entire image. A convolutional layer of a CNN takes a large number of these filters and passes them over the image to return one feature map per filter.

Max-Pooling Max-pooling is a technique to reduce the size, and therefore computational load, of feature maps through structured down-sampling. Max-pooling layers apply a maximum function over adjacent regions of a feature map (like using a sliding window) to encode the important information of how strongly a feature was activated in a given region of the image while simultaneously reducing redundant or unneccessary information about smaller activations. For example, taking 2x2 pieces of a feature map and keeping only the largest of the four values reduces the size of the feature map by a factor of 4! Max-pooling is also beneficial in that it allows CNN "vision" to be translation invariant, because it emphasizes the relative position of a feature rather than its absolute position. VGG16 follows groups of 2 or 3 convolutional layers with a max-pooling layer, which ultimately takes in initial feature maps of size 224x224 and ends with maps of size 7x7.

Densely Connected Layers Densely connected layers are typically the final layers in a CNN. These layers form the meaningful connection between the features of an image (detected by convolutional and max-pooling layers) and the corresponding labels associated with the image. These layers act like the human brain: just as we learned which combinations of features should be associated with a given label, densely connected layers use real-valued weights to represent these associations. For example, if we see an item that is orange, small, and fuzzy, we are taught to call it "caterpillar". Fuzzzy is not a feature we meaningfully associate with a baby carrot, so there is little connection between the feature "fuzzy" and the label "carrot". Similarly, in CNNs, each final feature is connected to each label through a weight (hence the name "densely connected"), and

those weights are learned through the training process (using an algorithm called back-propogation) to minimize loss and thus improve classification accuracy. Go ahead and add the pictures in here - you have them, may as well use them...

Using a Pre-Trained CNN for a New Task "Transfer learning" is the technical term. You may also want to talk about "modularity" in that the Convolutional part is one module, the head is another, and they're separately trainable and useable.

I went off on a tangent and am purposely not restructuring this paragraph yet. Sorry. As we have just seen, convolutional layers and max-pooling layers in a CNN are analogous to the human visual perception process, and densely connected layers behave like the human brain. In short, the approach to classifying an image is to detect the features in the image (like our eyes) and then assign labels to combinations of those features (brain). This analogy is also appropriate because it reflects the difficulty of the task: it takes many years and a significant amount of effort for humans to learn how to distinguish a large variety of features and also to connect those features to labels that are often complex, hierarchical, and subtle. Similarly, training a CNN is no small task. VGG16, in particular, has over 14.7 million trainable parameters in its "eyes" alone. Luckily, CNNs offer one benefit that humans do not: you can utilize the eyes and replace the brain for new tasks. In terms of CNNs, it is possible to build a CNN that uses the weights already trained on over 1.2 million images in the convolutional layers, and then only retrain a new classifier for any new classification task. This reduced task brings the number of required training images down from millions to only thousands. Furthermore, this kind of approach is quite reasonable when considering what the CNN was originally trained to classsify. Since the 1,000 categories from the ILSVRC span a huge variety of natural and unnatural objects, we can likely trust that the features detected by the pre-trained CNN to be diverse enough to be applied to a new task.

Pre-trained CNNs

I don't love this paragraph. Not sure how much to include, awkward flow. Pre-trained CNNs are CNNs that have been trained on a standard data set. The standard data set comes from ImageNet, a database containing over 14 million images in about 22,000 categories (called "synsets",

short for "synonym sets"). The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) was established in 2010 as a contest for CNN accuracy on a specific subset of ImageNet. Various CNN structures are tested on about 1.2 million images spanning 1,00 categories. These categories range from natural and man-made objects (e.g., daisy, chainsaw) to living creatures (e.g., ring-tailed lemur, sea lion, and dingo). There are also many categories which require subtle distinctions, such as differentiating between a grass snake and a vine snake. *Something about the best CNNs for this task are the ones that are famous.* Some of the most well-known pre-trained CNNs include AlexNet, GoogLeNet/Inception, VGG, and ResNet. *Can use just structure and train weights yourself, use fully trained model to reproduce ILSVRC results, or just use pre-trained weights for feature detection*

VGG16 Architecture The main difference between different CNNs is their structure, meaning the number of layers they contain and the pattern those layers are in. In our research, we have tested a few pre-trained CNNs, and we are currently using VGG16. Developed by Oxford's Visual Graphics Group, VGG16 has 16 "functional" (i.e., convolutional and densely connected) layers and 5 max-pooling layers, which function more to alter the structure of the information at each step.

Show VGG16 architecture image (use gimp + XCF file from my presentation to get the "right" image - let me know if you need help tweaking it...) and talk through the image dimension changes.

Contrast VGG16 structure with ResNet - one source, explain VGG16 is simpler, so we can more easily explain it and produce diagnostic images.

1.2.4 Model Evaluation Metrics

Multi-class classification refers to the number of categories to which an item may be identified. Multi-label classification is the special case of multi-class classification where categories are not mutually exclusive and, thus, an item may fall into a combination of categories simultaneously. Evaluating the accuracy of multi-label predictions requires attention to both false-positive and false-negative predictions. When predictions are probablistic and labels are binary, like in the case of our CNN, something about cut-offs for 0-1 metrics

Will include graphic from: https://towardsdatascience.com/precision-vs-recall-386cf9f89488

Confusion matrix shows

Accuracy, Precision/Recall, Confusion Matrix, and any other brilliant ways of examining multi-label multi-class model effectiveness. TBD

CHAPTER 2. DATA AND METHODS

2.1 Data

2.1.1 Choosing Class Characteristics

Include something that explains what class characteristics are? *Transition from CNN to shoes* In the beginning of this project, the literature (source IDK) indicated that geometric shapes are unique and well-defined enough to provide most of the necessary information to classify a shoe outsole. We have since settled on nine geometric categories, modified from (source): bowtie, chevron, circle, line, polygon, quadrilateral, star, text, and triangle. Many of these categories are self-explanatory, such as circle and triangle. Polygon is a catch-all for pentagons, hexagons, and octagons. Star is any concave shape, including X- or plus-shapes. Line is a difficult distinction, as most shapes are simply a combination of lines, but in this case it is reserved for cases where the other categies do not readily apply, such as repeating patterns of lines that are not distinctly quadrilaterals.

Defining categories this way does not remove all ambiguities. The best example lies in considering text. The letter "v" can easily be considered a chevron, and the letter "o" is clearly a circle. However, text is also an important category to encompass the variety of ways text appears on footwear outsoles, and it is not necessarily helpful (or possible) to try to categorize every shape in text into another category. Many of the ambiguities that arise can be solved by applying multiple labels to an image, but some shapes also do not fit into any categories. Applying comprehensive and consistent labels to difficult or ambiguous shapes is the most difficult part of this process.

2.1.2 Data Collection

Thousands of outsole images were web-scraped from Zappos.com, a large online shoe retailer. These images were then uploaded for use in a tool called LabelMe need to cite it, a labeling/annotating

Table 2.1 Geometric Elements. Categories modified from **?**

interface which allows users to easily select and label regions of an image. To date, about [2,200] shoes have been labeled, yielding about [24,000] multi-label images. *Image processing?* R code, using the imager package and spatial packages - check code to get a good list *Number of images per category?* To train the CNN, data was split such that 70% went to training, and 15% each to validation and test data. Actually using 60/20/20 right now, but that's easily changed in the model or in here...

2.1.3 Data Characteristics

-Quantities, examples, etc Show histogram of class distribution Talk about "other" - it's necessary to train the model on null data as well

256x256 images, aspect ratio not maintained (but efforts to label with relatively square labels).

One thing I need to examine is what happens if we use histogram equalization to modify the images before augmentation. I did a bit of that for presentation images and it changed the output probabilities a lot... which suggests it may help with images that don't have even color balance.

2.1.4 Augmentation

Talk about image augmentation

CHAPTER 3. RESULTS

3.1 Model Specs

Model Training

Model training was conducted using the *keras* package in R, which provides an interface to the neural network API of the same name which is written in Python, with a TensorFlow computational backend.

I want to include the following ideas, but just don't want to spend the time writing them out when I'm unsure about the format.

-Use VGG16 Convolutional base and train new classifier, done by getting features from conv base and training only dense layers -Model training parameters (e.g., augmentation parameters, drop-out rate) -Model predictions are multi-label binary, probabilities don't add to one

3.2 Evaluating the model

Model Performance

I feel like graphs will speak more here Yes, but you have to interpret them :) -Accuracy and loss during training -Examples of prediction -Ways to measure accuracy (TPR, FPR, ROC/AUC) This is in the introduction, you then use those measures and interpret them -Interesting case studies

3.2.1 Model Accuracy

3.2.2 Model Consistency

Look at how model predictions for the same feature of different color options for a shoe change. Should be a fun case study - does CoNNOR actually have the robustness we claim it should?

3.2.3 Heatmaps - Model Diagnostics

Add the fun stuff in here!

CHAPTER 4. CONCLUSION

I don't heckin' know Write this part last... :) But, we can conclude that CoNNOR works well for identifying features within outsole images, that the geometric feature set is able to classify many common tread elements (and we can compute statistics for how many shoes have recognizable features...)

4.1 Future Work

Integrate features for whole shoes - add spatial information; explore color histogram normalization to increase discrimination power

4.2 Philosophical Conclusions

APPENDIX A. ADDITIONAL MATERIAL

This is now the same as any other chapter except that all sectioning levels below the chapter level must begin with the *-form of a sectioning command.

More stuff

Supplemental material.

APPENDIX B. STATISTICAL RESULTS

This is now the same as any other chapter except that all sectioning levels below the chapter level must begin with the *-form of a sectioning command.

Supplemental Statistics

More stuff.