Visual expectations: using machine learning to identify patterns in psychological data

Skyler Place
Colby College

Follow this and additional works at: https://digitalcommons.colby.edu/honorstheses

Part of the Computer Sciences Commons, and the Psychology Commons

Colby College theses are protected by copyright. They may be viewed or downloaded from this site for the purposes of research and scholarship. Reproduction or distribution for commercial purposes is prohibited without written permission of the author.

Recommended Citation
https://digitalcommons.colby.edu/honorstheses/196

This Honors Thesis (Open Access) is brought to you for free and open access by the Student Research at Digital Commons @ Colby. It has been accepted for inclusion in Honors Theses by an authorized administrator of Digital Commons @ Colby. For more information, please contact mkelly@colby.edu.
Visual Expectations:
Using Machine Learning to Identify Patterns in Psychological Data

Skyler Place

Senior Thesis

Advisors:
Clare Bates Congdon
Department of Computer Science

Joseph Atkins
Department of Psychology

Spring, 2004
Acknowledgements

There are several people whose generous help have made this project possible. First, Joe Atkins for his immense help with the psychology experiment. The experiment was his original idea, and his guidance and expertise allowed the creation of an experiment that the both of us could use in our research. A larger part of my research first semester was under his wing working in his lab.

Another important piece of the puzzle is Patrick Rodjito, Joe’s research assistant, who played a major role in the implementation of the dice program. His many revisions to the code based on alterations Joe and I wanted allowed the efficient development of the experimental program.

Marc Attiyeh was a life saver when it came to the overwhelming task of organizing all of the data that I had. His knowledge of the Perl language allowed him to write parsers that turned my mountain of information into neat little piles. His late night parser writing kept me on track with processing the results when it came to crunch time.

Most importantly, my primary advisor, guiding light and dear friend, Clare Bates Congdon. In the midst of her own tenure work, Clare found time to convince me to consider going to graduate school, advise my interdepartmental research and be there at all times of day and night when I had questions for her that she had already answered twice and I had forgotten. It was Clare’s Machine Learning class that first awakened my
interest in Computer Science as something more than code writing, and her guidance and friendship since then has taken me in the direction that leads to this research.

Finally, my three other co-thesis writers, Rachel, Kevin and Danger, for dealing with my constant bantering and distraction in the lab while they were busy trying to write their own mega-documents. Together, we wrote something like three hundred pages in two weeks, and convinced the pizza places to deliver directly to Mudd at ungodly hours of the night. I think we all can safely say we are no longer afraid of seven page English papers.
# Table of Contents

Table of Contents ................................................................. iv  
Table of Figures ................................................................. v  
1 Introduction ........................................................................... 1  
2 Background/Related Work .................................................... 4  
  2.1 Psychology Background ...................................................... 4  
    2.1.1 Sensations ............................................................... 4  
    2.1.2 Perceptions ............................................................. 6  
      2.1.2.1 Similarity ......................................................... 8  
      2.1.2.2 Proximity ......................................................... 9  
      2.1.2.3 Continuity ....................................................... 10  
      2.1.2.4 Closure ......................................................... 11  
      2.1.2.5 Area .............................................................. 13  
      2.1.2.6 Symmetry ....................................................... 15  
      2.1.2.7 Perceptual Knowledge ....................................... 16  
      2.1.2.8 Attention and Perception ................................... 17  
  2.2 Computer Science Background .......................................... 19  
3 Details of the Task .................................................................. 27  
  3.1 Experimental Overview .................................................... 27  
  3.2 Program Design ............................................................... 32  
  3.3 Experimental "Recipe" Files .............................................. 34  
  3.4 Organization of Experiments ........................................... 38  
  3.5 Participant Experience ..................................................... 42  
  3.6 Participant Debriefing Comments ...................................... 44  
  3.7 Experimental Output ....................................................... 46  
4 Data Processing ...................................................................... 50  
  4.1 Cobweb Parser ............................................................... 50  
  4.2 General Statistics Parser ................................................ 54  
  4.3 Totals Parser ................................................................. 55  
5 Datasets and Data Organization ............................................. 57  
6 Results .................................................................................. 60  
  6.1 Analysis of Accuracy ....................................................... 63  
  6.2 Results, By Occurrence ................................................... 64  
  6.3 Results, By Cluster ....................................................... 70  
  6.3 Results, By Graph ....................................................... 76  
7 Future Work .......................................................................... 89  
8 Conclusions ........................................................................... 91  
References: .............................................................................. 93
## Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Gestalt Principle of Similarity, using colors</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>Gestalt Principle of Similarity, using size</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Gestalt Principle of Proximity, using columns</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Gestalt Principle of Continuity, using line segments</td>
<td>11</td>
</tr>
<tr>
<td>5</td>
<td>Gestalt Principle of Closure, using a white triangle</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>Gestalt Principle of Closure, using small circles to form a larger one</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>Gestalt Principle of Area, using within object rectangles</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Gestalt Principle of Symmetry, using pairs</td>
<td>15</td>
</tr>
<tr>
<td>9</td>
<td>Side-by-side dice faces</td>
<td>27</td>
</tr>
<tr>
<td>10</td>
<td>The correct possible faces a subject could see</td>
<td>28</td>
</tr>
<tr>
<td>11</td>
<td>Normal Curve</td>
<td>29</td>
</tr>
<tr>
<td>12</td>
<td>Is it a 5 or a 3?</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>Possible die combinations using faded dots</td>
<td>31</td>
</tr>
<tr>
<td>14</td>
<td>Screenshot of dice presentation program</td>
<td>33</td>
</tr>
<tr>
<td>15</td>
<td>Letters associated with each dot location on the left and right dice</td>
<td>35</td>
</tr>
<tr>
<td>16</td>
<td>Example recipe input file</td>
<td>36</td>
</tr>
<tr>
<td>17</td>
<td>Example global recipe file for experiment parameters</td>
<td>36</td>
</tr>
<tr>
<td>18</td>
<td>Representation of differences between experiments</td>
<td>40</td>
</tr>
<tr>
<td>19</td>
<td>Ordering of experiments</td>
<td>41</td>
</tr>
<tr>
<td>20</td>
<td>Example raw output file</td>
<td>46</td>
</tr>
<tr>
<td>21</td>
<td>Cobweb data formatting</td>
<td>52</td>
</tr>
<tr>
<td>22</td>
<td>Data parsed into format for Cobweb</td>
<td>53</td>
</tr>
<tr>
<td>23</td>
<td>Output from totals parser</td>
<td>55</td>
</tr>
<tr>
<td>24</td>
<td>Participants often mistook the value six for the value four</td>
<td>61</td>
</tr>
<tr>
<td>25</td>
<td>Both perceptual patterns happening concurrently</td>
<td>62</td>
</tr>
<tr>
<td>26</td>
<td>Accuracy percentages of the four datasets</td>
<td>63</td>
</tr>
<tr>
<td>27</td>
<td>Color schemes for varying patterns</td>
<td>65</td>
</tr>
<tr>
<td>28</td>
<td>Occurrence data from the first dataset</td>
<td>66</td>
</tr>
<tr>
<td>29</td>
<td>Occurrence data from the second dataset</td>
<td>67</td>
</tr>
<tr>
<td>30</td>
<td>Occurrence data from the third dataset</td>
<td>68</td>
</tr>
<tr>
<td>31</td>
<td>Occurrence data from the fourth dataset</td>
<td>69</td>
</tr>
<tr>
<td>32</td>
<td>Conceptual clusters for the first dataset</td>
<td>72</td>
</tr>
<tr>
<td>33</td>
<td>Conceptual clusters for the second dataset</td>
<td>73</td>
</tr>
<tr>
<td>34</td>
<td>Conceptual clusters for the third dataset</td>
<td>74</td>
</tr>
<tr>
<td>35</td>
<td>Conceptual clusters for the fourth dataset</td>
<td>75</td>
</tr>
<tr>
<td>36</td>
<td>Graph of occurrences of only incorrect sums in the first dataset</td>
<td>78</td>
</tr>
<tr>
<td>37</td>
<td>Graph of the percentage of incorrect occurrences in the first dataset</td>
<td>79</td>
</tr>
<tr>
<td>38</td>
<td>Graph of occurrences of only incorrect sums in the second dataset</td>
<td>80</td>
</tr>
<tr>
<td>39</td>
<td>Graph of the percentage of incorrect occurrences in the second dataset</td>
<td>81</td>
</tr>
<tr>
<td>40</td>
<td>Graph of occurrences of only incorrect sums in the third dataset</td>
<td>82</td>
</tr>
<tr>
<td>41</td>
<td>Graph of the percentage of incorrect occurrences in the third dataset</td>
<td>83</td>
</tr>
<tr>
<td>42</td>
<td>Graph of occurrences of only incorrect sums in the fourth dataset</td>
<td>85</td>
</tr>
<tr>
<td>43</td>
<td>Graph of the percentage of incorrect occurrences in the fourth dataset</td>
<td>86</td>
</tr>
</tbody>
</table>
1 Introduction

The goal of this project was to utilize the tools of machine learning to evaluate the data obtained through experiments in psychology. Advanced pattern finding algorithms are an effective approach to analyzing large sets of data, from any domain of science. Consequently, we have a psychological question and hypothesis, and a separate machine learning technique to assess these claims.

The realm of psychology that I focused on is visual cognition, and how an individual's knowledge affects how they see the world. This alteration of visual data is a part of perception — when the brain enhances the data coming in from the eyes. We devised an experiment that exploits these knowledge-based changes, and allows trials of a task for visual acuity. Incorrect answers can then be judged to see if the participant's knowledge of the stimuli appeared to have affected their ability to answer correctly. These answers are combined with other traits of the stimulus to create a dataset that was analyzed by machine learning tools.

The instrument chosen from the orchestra of computer science possibilities was the pattern finding algorithms known as machine learning. This approach allows patterns and generalizations to be found from large datasets. Useful across many domains, these programs are very powerful in finding both new information, and validating previously discovered patterns. They were therefore a perfect match for our psychological data. We were searching for patterns that would prove or disprove our psychological hypothesis. Machine learning offered means to not only find the patterns we were looking for if they existed, but also any other patterns that might have been present. Thus an applied set of
computer science tools would allow evaluation of information from a completely
different domain of academia.

This research was interesting for me because it allowed me to combine my major
and minor (Computer Science and Psychology) into a single project. As I have always
been interested in machine learning, it seemed to be a perfect match for both the research
question and the researcher. Furthermore, with the guidance of Joe Atkins, a psychology
experiment was devised that was both in the scope of my knowledge and my interests.
Cognitive psychology is very computation based, and approaches the human mind much
the way computer scientists think of the computer – as an information processing system.
This allowed me to apply many of the ideas from my computer science background
(especially Cognitive Modeling) to the psychology in this research.

As I fully intend to continue on to graduate school in a field that combines these
two interests (Cognitive Science), this project has allowed me to experience
interdisciplinary work, and focus on many different aspects of my Colby education at
once. The flexibility of the Computer Science department, and more specifically of Clare,
to include as much psychology in a coursework that counted for computer science credit
was a gift that allowed me to pursue such a research direction.

The following chapters of this book will take you through a detailed explanation
of all aspects of this research. First a background chapter gives insight into both the world
of visual perception and machine learning. Next the psychological experiment is
explained, followed by the data that was obtained from it. In the fourth chapter, the
results are displayed, with an exhaustive account of how they relate to the psychological
questions posed. Finally a future work and conclusion sections wrap up this research, and give direction for those looking to extend its boundaries.
Due to the interdepartmental nature of this research, the background information can be neatly divided into sections concerning the computer science, and the psychology. Although these two areas come together in the scope of my project, the necessary knowledge needed to understand the project occurs in two different arenas, and will therefore be presented in such a manner.

2.1 Psychology Background

Human vision is a much more complicated procedure than most people realize. The raw information coming in from the eyes is nothing compared to the images that we eventually "see". From color adjustment to feature detection and object recognition, many processes affect the stream of optical data. However, these alterations can be neatly grouped into two sections – bottom-up sensations and top-down perceptions. This research is concerned with perceptions, although a careful explanation of sensations will be provided to clear up any confusion between the two.

2.1.1 Sensations

Sensation is the process by which our sensory organs take in information from our physical world (Goldstein 2002). Stimuli in the environment are converted into electronic impulses which travel through our nervous system to our brain. This process is called
"bottom-up" because it starts with raw information from the source (our environment) and then transfers it up into more complex composite encodings and patterns in the brain. The upward direction represents the flow of information both from external to internal, as well as from simple to complex. For our vision system, sensations take place in the eyes.

The human eye contains photoreceptors, which take in light energy and transfer it into electronic neural transmissions. Two separate types of photoreceptors exist in the retina, rods and cones. Cones number around ten million, and are heavily focused in the center of the eye (the fovea) (Goldstein 2002). They are responsible for color vision, as well as the finite details of our visual world. They need adequate lighting to work effectively, and are useless in dim conditions. Cones are the focus of much of our attention, as we spend more time attending to what is in the center of our visual field.

On the opposite end of the spectrum are rods. Rods are scattered throughout the periphery of the eye, and work well in dim light. They are responsible for our peripheral vision, and are used mostly to detect movement and changes, allowing our eyes to rotate and focus our more powerful fovea cone based vision on stimuli. Together, rods and cones make up the basis for our vision system. They are the lowest-level sensory inputs, and at an individual level take in information without any knowledge of what they are seeing and without any preference.

Sensations, however, do receive some "preprocessing" en route to the brain for perceptual alterations. Feature detectors are groups of neurons that fire (are activated) when they "see" a particular stimuli. These detectors are patterns of rods and cones that work together to identify higher order features in our environment. Such features include straight lines, diagonal lines, and curved lines. Although very basic in their recognition
abilities, these simple patterns are being determined in extreme pre-conscious arenas, long before we are aware of what we are seeing. This low-level bottom-up information processing greatly aids the higher order alternations that occur at the perceptual level, because information is already being grouped and recognized.

All sensations occur very early in the attentive process (Goldstein 2002). Feature detectors fire independently of attention and conscious control. These detectors have developed over thousands of years of evolution, and are not something we have any control over. The information from the millions of photoreceptors is sent through the optical nerve (with different pathways for each eye) to the brain. The brain organizes and groups the sensory information in ways that allow for recognition of movement, object detection, depth perception and facial processing. This wide range of higher level abilities is known as perceptions.

2.1.2 Perceptions

Perceptions are the other side of the story, higher level constructs that convert the sensation information into the world that we see. Perhaps the most obvious perceptual ability is depth perception. There is nothing about an object that innately tells us how far away it is, or where it is relative to other objects. This is obvious when one watches newborns reach for objects. They misjudge the distance when reaching, and have to learn how things appear in their visual field relates to where they are in the real world.

There are many tricks our brains uses to help with this process. We know that things that are further away look smaller, that objects in the foreground are brighter and
more acute, and that partially blocked objects don’t mean that only half the object exists, but that something is in front of it. All of this is learned in the years of our development, every person’s visual system adapts to the natural world in which he or she exist (Sternberg 2003).

A large part of the perceptual picture is painted in the sub-conscious. Before we are even aware of what we are seeing, our brain is piecing together the information from both eyes to create a singular image, complete with relative movement, depth, and object recognition. Next we will delve into these subconscious alterations before moving on to even higher level perceptions involving knowledge.

Some of the more relevant perceptual processing to this research is object recognition. Our brain has the wonderful ability to organize information to create higher level “big picture” objects (Sternberg 2003). We draw patterns and fill in missing information in order to create the objects that we want to see — what makes sense to us visually. These ideas are governed by a set of rules known as gestalt perception (Goldstein 2002).

The principles of gestalt perception were developed by German psychologists in the early 20th century, before World War I. The governing idea is that the whole is greater than the sum of the parts. More specifically, visual processing creates higher-level objects that are not dictated solely by what is being seen. Information is being added to create patterns that do not really exist — they are constructed from only sensory information. The principles are: similarity, proximity, continuity, closure, area and symmetry. Together, they help us understand how we draw conclusions about simple objects that are visually presented (Goldstein 2002).
2.1.2.1 Similarity

The Gestalt Principle of similarity deals with grouping objects with similar properties. These properties include shape, size, color, texture value or orientation. Figure 1 illustrates this property concerning color combinations.

![Figure 1 - Gestalt Principle of Similarity, using colors.](image)

In Figure 1, the circles appear to be arranged in rows, and not columns. There is nothing about the image that suggests this at a sensory level – the photoreceptors have no opinion of how to orient the information, they just provide accurate views of the color and shape of the dots. However, in order to better comprehend the image as a whole, your brain makes the items appear as rows, based on the similarity of color. This works the same way with other grouping parameters, such as size. Figure 2 illustrates similar dots, this time grouped by size.
In Figure 2, the dots are organized into columns instead of rows. This is perceived based on the fact that the third column of objects is larger than the other ones. Gestalt Perceptions dictate that the organization matches the similar objects. Therefore these objects now appear to be vertically instead of horizontally arranged.

2.1.2.2 Proximity

Proximity deals with the perception that groups of objects that are closer together will appear to be together. Higher level objects will be formed based on the closeness of individual components. This is illustrated in Figure 3.
2.1.2.3 Continuity

Continuity is based on the perceptual preference for continuous figures. We like to see objects that make the most “sense”. Objects will be perceived as having the largest possible components, instead of lots of small individual pieces. As the guiding principle of Gestalt psychology is that the whole is larger than the sum of the parts, continuity is the idea that the bigger parts are better than smaller ones. This is viewed in Figure 4.
Figure 4 - Gestalt Principle of Continuity, using line segments.

In Figure 4, we perceive seeing a cross, made up of two lines – one on each axis. These two lines meet in the middle, at 90 degree angles. We see two lines instead of four half-sized lines that all originate in the middle. Continuity principles dictate that we prefer seeing the image as two lines, because the lines continue through the center to the other side, creating the largest possible logical parts within the whole image.

2.1.2.4 Closure

Closure is the idea that when information pertaining to the shape of a figure is missing, we still perceive the entire figure. We prefer to see the largest, most high-level object, the greatest generalization and extrapolation of the information presented to us. Therefore even when objects are not fully apparent, we create them out of thin area. This is illustrated through the white triangle in Figure 5.
In Figure 5, there is nothing imbedded in the image that tells you that you are seeing a white triangle in front of three black circles, instead of three circular shapes with pieces cut out of them. The triangle just “appears” in your vision because your brain likes to create the most logical shapes. Closing the white space between the circles into the shape of a triangle makes the most “big picture” sense, and allows a complete object to be created, instead of three misshaped partial ones.

Closure happens not only when objects are partially cut out or covered. It can also happen when objects are arranged in such a way that a higher order shape is created by combining the individuals. This is shown in Figure 6, with a larger circle being formed by the individual circle objects.
Figure 6 - Gestalt Principle of Closure, using small circles to form a larger one.

The larger circle is readily apparent, even though no complete outline of such a shape exists. The individual objects are grouped together to form a circle, because it allows the brain to organize the available information into a higher order image.

2.1.2.5 Area

Area deals with objects that are within each other. When a small object is inside of another object, the small object is perceived as the actual figure, while the larger object is the surrounding ground. The smaller shape is on top of the larger one, instead of being a hole in it. The inverse is also true when shading is used. Figure 7 illustrates this.
Figure 7 - Gestalt Principle of Area, using within object rectangles.

On the left is the original presentation, which is viewed as a small object on top of a larger one. The inverse works as well, when the object is shaded, the white box appears to be a hole in the larger black box. Gestalt perceptions of area help explain when objects appear to be in front of other objects, and aides with 2D depth perception.
2.1.2.6 Symmetry

Symmetry deals with perceiving objects as wholes, instead of as parts. This happens when similar patterns are repeated, and one sees them as whole objects, and not individual identical parts. This is illustrated in Figure 8.

![Symmetry Illustration](image)

Figure 8 - Gestalt Principle of Symmetry, using pairs.

Figure 8 shows six separate lines that appear as three sets of two. In each of the pairs, the two lines seem to create one continuous object. The symmetry of the parts leads you to believe that they are opposite sides of the same object. Your brain prefers to recognize these higher-order shapes than the individual symmetric parts that make them up.

These ideas of Gestalt psychology will be important when understanding the results of the experiments conducted. They will aid in comprehending why certain
patterns were predominant, and why errors with simple objects (dots) existed. It therefore may be helpful to refer back to this section when reviewing the results of this paper.

2.1.2.7 Perceptual Knowledge

Perceptions however, aren’t based only on what exists in the outside world – they are also dependent on what we have learned. Perceptions can differ on an individual level based on the knowledge and experiences of that person. How we see the world is influenced by what we hope to see, what we think we’re seeing, and what we’ve previously seen. Although this might sound very far fetched, some examples will make this obvious.

If one is searching a crowd for someone they are looking for, everyone starts to look like that person. You mistakenly identify people because your visual search is being directed by your desire to find something. Advertising plays heavily on your perceptual knowledge. When presented with red and yellow stimulus, McDonalds would like you to think of their stores, and your desires for fast food. You might start seeing golden arches where they don’t actually exist, in your hopes for finding that all-white-meat chicken nugget. Advertising is designed to direct your perceptions in a certain direction, and plays on your brain’s ability to alter what you think you are seeing.

This type of perception is known as constructive perception (Sternberg 2003). You “construct” the images in your visual field not only by what you are actually seeing, but with information that you already know about the stimulus. If you are driving along and approach a four-way intersection, most likely a red octagonal sign will appear. Even
if a tree branch is partially blocking the sign, so it reads S__P, you will have a strong idea of what the sign says. In fact, you might not even realize that some of the letters are missing (Sternberg 2003). Your brain constructs a complete image based on what it has previously seen in red octagons, and what is expects to see in the current one.

In this project, perceptual knowledge plays an integral role in the hypothesis of the research question. It plays an essential part of the experiments conducted, and is (hopefully!) being used in the minds of our participants when they are partaking in the tasks they are presented with. It is important to remember that our hypothesis that what they are seeing is dependent on what they believe, and what they have seen before.

There is one final area of perception that is important for this research, and that is how attention affects perception. Attention plays a key role in selective perception, when you only focus on certain objects in your visual field, and can completely ignore other ones.

2.1.2.8 Attention and Perception

There are many different kinds of attention, and they affect both our conscious and unconscious processes (Sternberg 2003). Preconscious attention can wake you up when someone walks into your bedroom, and involuntary attention draws you towards the source of large sudden sounds. The kind of attention that this research is concerned with is conscious attention, and the control of signal detection, selective attention, and divided attention.
These three attributes determine how you direct your conscious control of attention. When you search your visual field for a particular stimulus, you are using signal detection, to identify and exclude other objects until you find the one that you are looking for. Humans are generally very good at this process, but the similarity of the confounding stimuli to the search stimuli greatly affects the time it takes to complete these kinds of tasks (Sternberg 2003). How we choose what we detect is based on what we focus on in our environment. This is guided by selective and divided attention.

Selective attention is when you choose to focus on one part of your visual field and not another. Focusing on salient stimuli allows us to attend more cognitive process to them, and aids in both recognition and comprehension. Divided attention is when we are actively focusing on more than one task at once. Humans often do this, as when you are talking on your cell phone while driving. We must be careful however to attend to the task that requires immediate attention, as the balancing of attentional resources has lead to many a car accident.

These ideas will play an important role in the design of the psychological experiment. In order to garner accurate results, conscious attention needs to be focused on our specific visual stimuli, and not elsewhere. Therefore several changes were made to the original design to better control divided and selective attention, so that signal detection is only being done on our experimental visual task.

This ends the discussion on perception. Several important areas of perception play crucial roles in the psychology experiment in this research. The next section will discuss the Computer Science background needed to understand the machine learning approach that was applied in this study.
2.2 Computer Science Background

Machine learning is closely tied to both artificial intelligence and cognitive psychology. It focuses on the computational background that exists in both human learning and artificial learning. (Langley 1996) Although some machine learning techniques are inspired by human processes (such as learning), it is not an essential feature of the discipline. Machine learning is at its heart a set of tools that can be applied to any situation where learning is involved. As human learning methods are not associated with the content of the material being learned, neither are the applications of artificial machine-based learning.

There are several different types of machine learning. Some techniques are focused on learning from experience, and in building systems that improve over time. Such is the case of genetic algorithms, whose focus is to improve results using the random mutation and selection that happens in Darwinian evolution and natural selection. Some systems deal with defining explicit sets of rules, and applying these rules to new situations to learn about them. These rules can also be "evolved" or updated as the system learns to improve its acquisition abilities. Such "learning how to learn" is a technique that humans use, and has been applied to artificial systems. Others focus on the classification of data, either to predict the future classification of similar data, or to group the data into clusters, or patterns. It is these classification machine learning systems that are used in this research.

Classification itself can occur on many different levels. In machine learning, there are two major paradigms – supervised and unsupervised. Supervised classification or
supervised learning as it is more commonly called focuses on data that is already divided into subsections. The clusters of the data have been previously identified (either by humans or another machine learning approach) and the goal of the system is now to classify new data into one of these clusters. Imagine a dataset made up of several attributes, including a "class" attribute – which determines what cluster the data is in. The supervision comes from the humans – they have supervised the direction that the data will be processed in, based on their selection of a class attribute.

For an example, we can use data that is about women who have had mammograms. The attributes can include things such as blood type and age, number of previous hospital visits and perhaps some measurement of previous cancer in the family. The class attribute – the one that decides which group the women are in, is whether or not they have cancer. The machine learning system would learn based on the other attributes which women have cancer and which ones do not. Then in the future, when a new patient checks into the hospital, the system could tell doctors the likelihood of her having cancer based on her attributes, before any expensive test is ever done. Obviously mammograms would still need to be completed, but the system would allow doctor to have a rough idea of what attributes appear more often in cancer victims, and perhaps allow earlier diagnostics, and raise survival rates.

These systems are great if the data is already divided – the doctors knew they wanted the class attribute to be whether people had cancer or not. But what if they wanted instead to find out what the most common patterns were that already exist in the data? This system couldn’t tell doctors if age is related to having cancer, or if number of hospital visits relates to blood types. Because the data came into the system already
divided by the class attribute, there is no opportunity for other possible patterns. The only thing the system focuses on is whether or not the person has cancer.

Unsupervised learning approaches data organization from a “hands off” perspective. Instead of seeding the data with a class attribute, humans instead give the system the raw ungrouped data, without any direction or guidance for what they are looking for. The system itself then determines what the patterns are in the data, based on clustering data with similar attribute values, and keeping the most difference between the clusters (Fisher 1987). Using this system, doctors might not have patterns that show if people have cancer or not, but they would instead have the most prevalent pattern in the data. Whatever the pattern is, the system would create structures that define it. One of the unique aspects of these systems is that although it might not find what the doctors are looking for, it might find something that they hadn’t even considered. Because the system is given no guidance into what the direction of the researcher’s focus is, there is no opportunity to skew the data to look for a specific pattern. Therefore new patterns might be discovered that researchers had never even dreamed of.

It is this unsupervised machine learning that was applied to the psychological data in this experiment. One of the goals of the system was to be able to evaluate the hypothesis of the psychologists, by seeing if the patterns that they were expecting actually existed in the data. An unsupervised approach allows us stay “hands off” with the data, so that no experimenter biases could be involved. If a supervised method had been used, the class attribute could have been whether or not the data has the expected psychological results or not. Although this very well could have validated the
psychological hypothesis, it wouldn’t have allowed the existence of other patterns in the dataset.

Unlike the doctors in the previous example, we were less concerned with finding specific patterns, as we were with making sure they were the only patterns in the dataset. Perhaps many patterns exist in the data that are equal to or more significant than our expected pattern. A supervised method would never allow such an occurrence to happen. However using an unsupervised approach, the most prevalent patterns come to the surface, whether they are what you were looking for or not.

The unsupervised machine learning program that was used in this experiment is called Cobweb. The version used is Cobweb/3 developed at the NASA Ames Research Center. Cobweb is an unsupervised method that does conceptual clustering. This idea follows the generalized definition of unsupervised learning; where clusters are created that contain similar data, while keeping the biggest contrast to the data in other clusters (Fisher 1987).

Cobweb coincidentally was originally based off of the psychological definition of concepts. To psychologists, concepts are generalizations that are created to store higher-level information in memory. For example, when you think of a dog, you don’t think of the thousands of dogs you have probably seen in your life. You think of the defining features of the typical dog – furry, four legs, barking and has a cold wet nose. These generalizations make up your concept of the dog – the required attributes that every dog must have. As you see more dogs, your concept of what a dog is becomes more generalized - you learn of all the options that exist within an animal being a dog, and incorporate these into your overall idea. Cobweb was designed to create groups of
concepts based off of these required attributes (McCusick and Thompson 1990). It models information the same way that humans create concepts, through incremental adding of new ideas and samples.

Cobweb as a machine learning entity has five features that define it as a conceptual clustering system. These are hierarchical organization of concepts, top-down classification, unsupervised learning, incremental learning, and hill climbing. Together, they define Cobweb as a unique option amongst machine learning systems.

Hierarchical organization is due to the nature of the structure that Cobweb creates. Cobweb uses trees, a structure that allows a depth of generalization. Instead of a flat structure where the clusters are only grouped on one level, Cobweb allows you to decide at what level you would like to view the concepts (Fisher 1987). To use the dog example from before, the leaf nodes (the bottom of the tree) might contain individual dog species, such as French Poodle and German Shepard. The next level up might be done by size, big small and medium. Above that could be short or long haired. And finally at the very top is the root node, which contains all the examples, and is the most general version of the concept – dogs. The depth of the tree (the number of hierarchies) depends on in one hand the raw number of examples that are in the tree, and in another, how different they are. Cobweb will only make a tree that is as complex as necessary to retain the similarity within and difference without that it strives for.

The next section that defines Cobweb is top-down classification. This means that each new example that is included into the tree structure starts at the top (the root node) and works its way down to the clusters that it best fits in. This allows accurate
generalizations to be formed, because the new instance transcends the tree in the same manner as the concepts are created – from general to specific.

The third attribute is what has already been mentioned – Cobweb using unsupervised learning. There is no guide to the classification of data; Cobweb only uses its own internal algorithm to guide similarity and dissimilarities amongst the examples in the data.

Cobweb also uses incremental learning. This means that it adds one example at a time. Some systems require reading in all of the data before they start organization. Cobweb’s format is much more efficient because it is never necessary to reprocess instances that have already been placed in the tree. This efficiency can be crucial when dealing with large datasets.

Finally, Cobweb uses hill climbing. This means that Cobweb makes changes to the tree only based on new knowledge (examples). There is no need to store massive sets of alternative tree structures. Because each new hierarchy in the tree is dependent only on the hierarchy before it, such a hill climbing approach is possible. This saves vast amounts of computer memory, as other possible tree options do not need to be saved.

These five attributes make Cobweb unique as a machine learning algorithm. Let us now go into a little more detail concerning the process that Cobweb uses to determine placement of examples into the tree structure.

The algorithm that Cobweb uses to place examples into the tree is based on conditional probabilities. When placing new data in the structure, it calculates the probability of that example appearing in that node. This probability is based on an average of all the attribute values of all the data in the node. This creates a very
generalized idea of what the pattern is in that node. In order to create more finite placements, Cobweb will compare the individual value of attributes.

To do this, Cobweb keeps track of the average and standard deviation from the mean of the individual attribute values within each node. It can then compare the attribute values of the incoming example to see if it is within a certain range of the current node values. This range is called the acuity. The acuity allows the user to decide how similar they would like the node to be. The higher the value, the easier it is to gain access to that node (as a new example), because the possible range of standard deviations is very high. As the acuity gets lower, (closer to zero) the range of possible values the node are much more limited, causing the node to be much more homogeneous.

There are several operators available to Cobweb to keep the values within each node at the appropriate acuity level. Most obviously, Cobweb can incorporate a new example into that node if it is within the acuity parameter. However if it is not, Cobweb can either create a new sibling node, or split the node into two children. Cobweb will choose whichever solution creates the most similarity within the new nodes, and the most difference between them. It can also merge two nodes back together again, if they start becoming very similar.

These operators allow the construction of the tree structure. The tree grows and morphs as new items are added. Therefore there is an inherent ordering risk to the data. The tree structure is based on the order in which the data is read in, and therefore the arrangement of the data can in some cases determine the placement of examples, and therefore the patterns that come out of the system. Although this risk is small, it is
important to remember, and might call for running the system on the same data in a
different order to test for inconsistencies.

Overall, Cobweb is a very robust system that allows customization on several
levels to control the depth and acuity of the overall tree. The patterns that come out of
Cobweb are easy to understand, yet contain a plethora of information.

The next section will focus on the details of the psychological experiment, and
will give detailed explanations of both the task of the participant, as well as experimental
design and implementation.
3 Details of the Task

This research focuses on using a computer science approach to validate a psychology question. The question posed by the psychologists deals with the application of perceptual knowledge (as explained in the background section). The experiment more explicitly deals with the ability to correctly perceive a visual stimuli, and the effects that perceptual knowledge have on the incorrect answers.

3.1 Experimental Overview

Subjects are asked to view two side-by-side die faces. These appear as single-sided 2D faces. The participants are asked to calculate the two individual values based on the dot configurations, and respond with the sum. What makes the task interesting, however, is that the dice are displayed for only a fraction of a second (50 milliseconds). This brief amount of time allows only a single glance at the figures, with no time for eye movements or advanced visual analysis. Figure 9 illustrates example 2D faces that are presented to the participants.

Figure 9 - Side-by-side dice faces.
In order to simplify the task, only one configuration of each dice number is possible. This becomes an issue with numbers that could be flipped to create a different arrangement. Numbers such as one, four and five are the same in all rotations, but the numbers two, three, and six appear differently when rotated. Subjects were presented the following diagram (Figure 10) to allow them to understand which rotations of the die faces would be legal.

![Figure 10 - The correct possible faces a subject could see.](image)

It is important to note that the light gray outlines of the dots that are not darkened are not present in the actual stimuli; they are used in this diagram as placeholders. Participants were allowed to study this diagram for as long as they liked before...
continuing on with the experiment. We did not want them incorrectly guessing because they thought they saw a different rotation.

With the die faces comes a particular set of perceptual knowledge. Most educated people are aware that when two die are randomly rolled, there is uneven chance that certain sums will appear. Seven is the most likely, because it has the greatest number of combinations (1-6, 2-5, 3-4, 4-3, 5-2, 6-1). The sums of six and eight are then equally second most likely, and so on. This pattern creates what is known as a normal curve – a peak in the middle with an equal slope to either side. Figure 11 represents the different possible sums, and the likelihood of each appearing.

![Normal Curve for the Rolling of Two Dice](image)

Figure 11 - Normal Curve

We are interested in seeing if participants' answers are affected by their knowledge of the normal curve. When presented with die combinations that don't naturally sum to seven, will participants incorrectly answer to create the most likely appearing combination – seven? This could happen because the die faces are appearing
so quickly. When people think about what they are seeing, they are most likely recalling a mental image of the dice faces. This allows cognitive processes (most importantly perceptual knowledge), the chance to alter the information. Because of the sub-conscious processing, people would not be aware that they are answering incorrectly; they will think that what they saw is actually two dice that add to seven.

In order to further investigate this process, we added a second layer of doubt into the system. Dice that are harder to see, and have higher dot ambiguity, have greater chances of being perceptually altered, because the participants are less certain of what they have seen. They will make their judgments more heavily on what they remember seeing, than what they actually saw for the brief flash of the dice.

The doubt that we added are faded dots. Individual dots in the number arrangements can be lightened to make them harder to see. When multiple dots are faded, the "black" dots can appear to form one number, while the addition of the faded dots creates a different number. An example of this is a five becoming a three, through the use of two faded dots (Figure 12).

![Figure 12 - Is it a 5 or a 3?](image)

When combined with the second die face, the participants can be skewed to see one combination instead of another. This can be used to achieve any sum, not just seven.
If participants always choose the “lighter” combination (when they summate using only the dark dots) then there is no proof that perceptions are affecting the answers. However, if they only ignore the lighter dots when the sum then goes to seven, then we know that their perceptions are erasing the lighter dots to achieve the more probable solution.

It is important to remember that these dots are light, but not invisible. If given more time (like a second) then there would be near 100% accuracy. What makes the dots so hard to see is that they are visible for only a fraction for a second, not that they are so light that our eyes have trouble picking them out.

Figure 13 shows a series of possible die combination, using a variety of faded dots. Dots can be faded so that you either need to add them or subtract them to reach sums of seven. They create another level of complexity which makes the users’ task that much harder.

\[
\begin{array}{c}
5 + 4 = 9 \\
3? + 4 = 7 \\
5? + 2 = 7
\end{array}
\]

*Figure 13 - Possible die combinations using faded dots.*
Now that the experiment in general has been laid out, the following sections will explain the details of the program that presents the die, the way the die layouts are chosen and created, and how the user’s answers are recorded and compared.

3.2 Program Design

The experiments were run using a proprietary program written in Java. Java was chosen because it was the only language the research assistant (Patrick Rodjito) who wrote most of the code for the project was familiar with. A C and Open GL implementation was also considered, but was eventually discarded due to unfamiliarity with the languages. Both myself and my psychology advisor (Joseph Atkins) were familiar with C, but neither of us had any Open GL experience. As previously stated, our RA also only knew Java. Java therefore became the default choice.

One of the major issues with using Java was the speed at which it could draw the die faces. The faces need to be created in their entirety before the participants can view them, and then after fifty milliseconds taken down as a singular unit. Because Java draws the dots and square background that make up the dice faces in the order that they are called, someone watching could see the dice faces being constructed, thus defeating the purpose of only viewing the completed die for a very short period of time.

To combat this problem, a black mask was used to hide the construction of the die faces. Once the faces were completed, the mask would be removed for 50 milliseconds, and then returned to the foreground. This would allow the process of building and
deconstructing the dice to be hidden, and allowed a streamlined presentation of the stimuli.

The program was designed as a shell that would read in "recipe" files to determine what dice combinations would be displayed. These experimental files are explained in detail in the next section. In addition to reading in files, the program also outputs the results of the participant's trials to a file. This output is also explained in a later section.

Figure 14 is a screenshot of the dice faces being presented. The dice are white boxes with dark dots on them, surrounded by black space. The program was run on a secondary 21" monitor in full screen, so there was no menu bar or other distractions, and the images were properly centered.

![Figure 14 - Screenshot of dice presentation program](image_url)

In addition, it is crucially important that the participants view the two dice faces equally. In order to do this, they must be looking directly at neither die, but instead in the
center of the screen. This allows equal painting of the images on the retina; with no one
die getting greater fovea area. If the participants are looking at one die instead of the
other, than their ability to recognize the pattern of dots on one die is greatly increased,
while the other die has little chance of accurate depiction. This would greatly affect their
answers to the sum of the die, and would skew the results in unintended direction.

To combat this, a stimulus was placed in the middle of the screen to attract the
user's attention. Flashing non-sequential non-repeating letters were displayed in the
center of the screen before the die faces appeared. There were a random number of letters
displayed, ranging from four to eight. This way the participants could not predict when
the actual stimuli would appear, so they couldn't move their eyes in anticipation. To
further guarantee that they were looking at the letters, they were requested to recite them
out loud as they appeared. Following the letters the die would appear (there was no
overlap), and then the participants would have an opportunity to type in their sum answer.
Further issues concerning attention and perception are explained in Section 2.1.2.8. The
next section will explain the experiment files that determined what die faces were
presented to the users.

3.3 Experimental “Recipe” Files

Recipe files were used to specify what dice faces would appear in the
experiments, as well as universal parameters for each experiment. In order to aid in the
design of these files, the individual dots on the die were given letter designations. This
allowed an easier representation of dot patterns in the recipes. Figure 15 shows the lettering for each dot position.

![Left Die](image1)

![Right Die](image2)

**Figure 15** - Letters associated with each dot location on the left and right dice.

Therefore, in order to create a correctly presented three on the left die, the locations A D G must be activated. In the recipes, each line represents one die face (one trial). Figure 16 shows an example recipe file. In this figure, there are four columns of data. The first column is labeled #sum, and is the sum of the two die values. This includes all faded dots. The next column is die1, and is the dots that are “ON” on the left die. ON dots are “black” by default. Each pattern is described by its letters, as in Figure 15. The next column is the same, but for the right die, die2. Finally comes the modify column. This is where fading is taken into account. Each dot (by letter) that is faded is described here, along with its new contrast value. The contrast value tells the system how light to make the dot. The higher the value, the lighter the dot. In Figure 16, the dots are faded to three different contrast levels, 90, 100 and 110. A value of 125 is OFF, when the dots no longer are visible (are white with the background of the die).
#sum die1 die2 modify
5 D MJKL
5 AG HSN K=90
5 ADG HN D=90
5 ACEG K
6 D HJKLN
6 AG HJKL J=90, L=110
6 ADG HSN D=110, K=90
6 ACEG HN
6 ACDEG K c=110, E=110
7 D HJLMN
7 AG HJLMN K=90
7 ADG HJLN D=110
7 ACEG HSN K=110
7 ACDEG HN D=90
7 ABCEFG K
8 AG HJLMN I=110, M=110
8 ADG HJKN J=100
8 ACEG HJLN
8 ACDEG HSN K=90
8 ABCEFG HN B=110, F=110
9 ADG HJLMN D=90, I=110, M=110
9 ACEG HJKLN
9 ACDEG HJLN C=110, E=110
9 ABCEFG HKN
#end_of_file

Figure 16 - Example recipe input file.

The secondary recipe file tells the system global parameters for the entire experimental run. These are values that are the same for every trial in the experiment. Figure 17 is an example global recipe.

Visual Expectations data file
Format: a predefined word followed by a number (operator is optional).

#end_of_data

Figure 17 - Example global recipe file for experiment parameters.
The variables in this recipe determine the timing and contrast of certain features in the experiment. All of the time values are in milliseconds, so 1000 equals one second. The first variable, letter_interval is how long each letter is displayed for. The letters appear in the center of the screen before the dice appear. The letter_pause is how long of a break there is between each letter, the amount of relative off time. This is set to one second. The third variable is flash_interval and is set to 50 ms. This is how long we want the die to appear on the screen. This value doesn’t include the time it takes to build the die image; this is just how long the black mask is down for, how long the participants actually see the die.

The flash_pause variable is how much time is given to the participant to answer. In every example, participants answered in less than 3 seconds, so using 4 seconds as a pause is more than enough time. The min_letters and max_letters and the range for the number of letters that appear. The actual number is determined randomly, to prevent the participants from predicting when exactly the die will appear. Dot_color_on is the ON value for the dots discussed early. It is the “black” value, the contrast of the non-faded dots. Die_color and dot_color_off are the same because we want the dots that are not black or faded to be invisible, i.e. the same white color as the background of the die.

Generally speaking, in all the experiments that were run in this project, the only global value that changed was die_color_on. In different experiments the “black” value was changed to make it easier or harder to see the ON dots. The lighter the dots became (the higher the contrast value) the harder they were to see. The rest of the timing values remained constant across all the experiments.
The next section will explain the details of the ten experiments that the participants took part in, and how they were ordered and presented.

3.4 Organization of Experiments

Each participant went through eleven different experiments. They were run back to back, with slight pauses in-between to rest the eyes. All together, the experiments took roughly an hour to complete. Each experiment used its own global and configuration recipe, allowing for customizing of dice sums, fading, and ON contrast values. The first experiment for each participant was a demo, containing only a few combinations so that he/she could get use to what they were going to be seeing. Because of this, the answers from these practice trials were discarded. Therefore, we will in the future always refer to there as being ten experiments, and not eleven.

In order to narrow down the plethora of die combinations which could be shown to the participants, the extremes on the end of the normal curve were discarded. The sums of two, three, four, ten, eleven and twelve were not included in the experiments. This was done in order to focus on the sum of sevens, and not worry about solutions that were statistically uncommon and perceptually easier to answer. Early testing of the system using these fringe combinations found that they have a higher accuracy rate, and do not provide as interesting results. Therefore, in all of the experiments run on actual participants, only the sums of five, six, seven, eight and nine appear.

Each experiment contains the same number of die combinations, from this point on referred to as trials. There are 24 trials in each experiment, which means that all
together each participant views 240 die combinations (24 in each of 10 experiments).

However, in order to combat ordering phenomenon in the results, the order in which the trials are presented are randomized by the program for each individual. This means that although each participant is viewing the same die combinations for each experiment, the order in which they appear will be different.

Two rounds of testing were performed. One in the early fall, and one in mid winter. The first round contained fifteen participants, the second round twelve. Therefore in total, twenty seven participants took part in the research experiment. Within each round, the experiments used were identical. However, between the two there are some slight differences. The experiments in round two had slightly harder to see die combinations, using more fades and lighter ON values. This was because preliminary results from the first round showed that subjects were having very high accuracy rates. High accuracy rates do not provide interesting results, because our hypothesis only comes into play with wrong answers. If people are always responding with correct answers, then no perceptions are being used, because the sensory information is accurate and non-ambiguous.

In the following table, the important variables from each of the ten experiments from the two rounds are shown. Because some experiments are identical, there are not ten configurations in each round. Figure 18 below illustrates the differentiating values between the experiments.
The information in the columns can be divided into three separate parts – experiment label, ON value and faded value. The label is the name of the experiment. Generally speaking, experiments with higher number labels have harder combinations using fading and lighter ON values. The experiments in the graph are sorted by label, and therefore are in some ways sorted from easiest to hardest. The next value is the ON value, or the contrast value of the “black” non-faded dots. This is either 70, 80 or 90 depending on the experiment. Remember that the higher the number, the lighter the dot, because 125 is OFF, or white.

The next four columns represent the possible fading levels used. Dots can be faded to 90, 100, 110 or NONE. NONE implies that there are no faded dots at all in the experiment, and therefore all dots default to the ON value. Some experiments use only one faded value (like B2B2), but most use multiple levels to make the task of recognition easier.
more difficult. Reading across the rows, it is easy to see for each experiment what the ON value and faded values are.

Now that the individual experiments are explained, let us clarify the ordering of the experiments. Each participant (from within a round) are presented the same ten experiments in the same order. This is important to compare learning effects, and to see if some people improve over the course of the experiments, and others don’t. So although the trials within the experiments are random, the ordering of the actual experiments are the same. Figure 19 shows the ordering of the experiments for each of the two rounds, using their experiment labels.

<table>
<thead>
<tr>
<th>#</th>
<th>Part I</th>
<th>Part II</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B0D1</td>
<td>B0D2</td>
</tr>
<tr>
<td>2</td>
<td>B0B1</td>
<td>B2B1</td>
</tr>
<tr>
<td>3</td>
<td>B1B1</td>
<td>B0B1</td>
</tr>
<tr>
<td>4</td>
<td>B2B1</td>
<td>B2B3</td>
</tr>
<tr>
<td>5</td>
<td>B0B1</td>
<td>B0B1</td>
</tr>
<tr>
<td>6</td>
<td>B0B2</td>
<td>B0B3</td>
</tr>
<tr>
<td>7</td>
<td>B1B2</td>
<td>B1B3</td>
</tr>
<tr>
<td>9</td>
<td>B0B3</td>
<td>B2B3</td>
</tr>
<tr>
<td>10</td>
<td>B0B1</td>
<td>B1B4</td>
</tr>
<tr>
<td>11</td>
<td>B0B1</td>
<td>B2B1</td>
</tr>
</tbody>
</table>

Figure 19 - Ordering of experiments.

Each round starts with the demo trial, and then continues on. The first “real” experiment is identical to the last (B0B1 in round 1 and B2B1 in round 2) to allow for learning comparisons. When compared to Figure 18, one can note that both rounds follow a similar progression. An “easy” experiment followed by a difficult one, followed by an
easy one, followed by a more difficult one, etc. As the experiments progress, they get harder and harder, broken up by easier experiments. This ordering was done for two reasons. First, interspersing harder experiments with easier ones increases participant confidence and keeps them focused on the task. If they were just seeing harder experiments, they would get frustrated and perhaps give up on the task. The last thing we want a participant to do is actual random guessing, where they are not even looking at the stimuli and are just responding with random sums.

The second is that we want the hardest experiments at the end. This is so that the participants would have gotten used to the task, and their minds would have the longest time to allow perceptions to “leak in”. As they continue to look at similar stimuli over and over, their knowledge will influence their visual images more and more as the stimuli becomes habituated and familiar.

The next section will discuss the participant experience - what they did from the movement they came into the lab to the moment they left, so that the experiment in its entirety can be properly understood.

3.5 Participant Experience

Experiments were run in the Visual Expectations Lab in the Psychology Department on the 4th Floor of Roberts Union. All of the participants were current students in Psychology 121, and were participating in the experiment to fulfill a research requirement for their class. Their participation however was completely voluntary, they could discontinue at any time with no academic penalty.
The room was set up using a table with a 21” monitor and a modified keyboard placed on it. The keyboard contained only the numbers two through twelve, which are all the possible sums of two die. The shades to the windows were closed, and the lighting in the room was set at a constant level, approximately 50% of the possible luminance. The dim light was used to allow a clearer view of the monitor, with less glare and easier recognition.

The participants were seated in a chair facing the table and the monitor, and given a consent form to read that briefly described what they were doing. It asked them for their initials, their gender, and their handedness. It explained there was no risk involved in the experiment. However it did not mention faded dice, or the fact that we were limited the range of the sums (from 2-12 to 5-9).

Next the participants were given the die configuration form (Figure 10). They were allowed to view the layouts for as long as they wanted to make sure they were aware that only certain configurations would appear. After satisfying their understanding of the die faces, the participants were explained the importance of hand placement. Because the program was recording the amount of time it took the participant to answer, it was crucial that their hand went back to the same spot every time. If they crepted up closer to the keyboard as the trials progressed, the response times would go down, independently of how quickly they were actually responding. Participants were guided into having an easy spot to remember to return their hand to in between trials.

After figuring out a hand position, and getting lined up close to the table and the keyboard, the experiments started. The demo trials came first, with a pause afterwards. At this point the participants’ were asked if they had any questions, or if they were confused
about any part of their task. After all questions were answered, the experiments started. Brief pauses in between experiments allowed the participants to rest, and a mandatory five-minute break in between experiments five and six allowed them to get up and walk around and stretch.

After all ten experiments concluded the subject was given a debriefing form to explain the "truth" behind the research. Because many psychology experiments use deceit to trick the participants into thinking the researchers are measuring one thing to actually accurately measure another, debriefing is usually crucial in explaining the actual focus of the research. However, this research did not use any deceit, and therefore the debriefing form was less important than it would be in other situations. Participants were still asked to answer a few questions, concerning their answering patterns, and if they believed any knowledge or outside influences were affecting their answers. After making sure they didn't have any more questions concerning the research, the participants were thanked and allowed to leave. The entire process took roughly an hour.

The next section will discuss some of the debriefing comments made by the participants, and how they believed they were responding to the dice combinations and summation task.

3.6 Participant Debriefing Comments

Each participant filled out a debriefing form that asked them questions concerning the experiment. The questions asked what they would change about the experiment, how
they found the experience as a whole, and if they felt they were using a pattern, or outside knowledge to complete the summation task.

Most participants stated that they felt the total number of trials was too many, and that by the end of it they were slightly tired and bored. This was for the most part expected. Although I personally never sat through a complete set of experiments, just suffering through one set of 24 trials was bad enough.

More interesting however, was that some participants felt that their answers were being skewed. They stated that they knew that the sum of seven appears more often when rolling two random die, and therefore they were purposely trying to combat answering seven all the time. However, they claimed their attempts failed, because as soon as they weren’t actively thinking to not answer seven, their sub-conscious took over and they fell back into this pattern. These comments were amazing in that participants seemed to be at moments aware of their perceptual knowledge affecting their answer, but once their attention shifted, they were once more unaware. This acute recognition was far greater than we were expecting from the participants.

Most likely, this acute awareness to their own perceptual cues comes from the fact that our pool of participants is neither random nor stereotypical. Using psychology students means that they have a much higher level of psychological knowledge and awareness of sensation, perception, and the vision system. Although this does not affect their ability to complete the task, it does affect their knowledge of the situation, and makes them more aware of their internal processes than a normal person. We don’t believe that these perceptual attention shifts affected the results, and therefore they are noted, but not mentioned in explaining results.
The next section will focus on the data gathered from the participants by the program during the experiments. These raw output files form the basis of the rest of the research, as they contain information pertaining to the correct vs. answered sums, and the die combinations that were presented.

3.7 Experimental Output

For every participant, ten separate output files were generated - one for each experiment. The output files contain line-by-line information for each trial in the experiment. They also contain basic statistics concerning the accuracy of the participant’s answers. Figure 20 shows an example output file.

Figure 20 - Example raw output file.
There is an enormous amount of information in this file; not all of it was used in the data processing, however it will all be carefully explained. The top two lines in the file are a header which contains information pertaining to the time, data and experiment input file. VE_EEM_BOB2_3-11-20_2030.dat is the file name, and includes abbreviations for; the research project (Visual Expectation – VE) the participant’s initials (EEM), the experiment recipe (BOB2), and the date and time the experiment was run (November 20th, 2003 at 8:30).

The next line describes the positioning of the two dice on the screen. X and Y coordinates of each die face are defined, as well as the white background color (125). This information is constant across all trials in all experiments, and is not used in any way shape or form in the data analysis of this research.

The following line is a column header which explains the meaning of all the rest of the data in the file (excluding the #end of file line). Every line after the header is a one-to-one correlation to a trial – each trial gets one line. The first column header is #ct, which is the trial counter. This describes the order in which the trials were presented. The ON value comes next, which as dictated by the global recipe file, is the contrast value of the “black” dots on the die. Similarly is the following OFF value, which represents the invisible dots which are not currently active. They are the same contrast level as the white background of the dice.

The next 14 columns are labeled A through N, and correspond to the letters of the dots as explained in Figure 10. The values for each of these are the contrast value of that dot. If the value is the same as the ON value, then that dot is black. If it’s the same as the OFF value, than that dot is invisible. If it’s any other value, than that represents a faded
dot. Next to N is the sum column, which represented the sum value of the dice (including faded dots) and then the res column, which is the participant’s response. After that is the letters_ column, which is the letters (in order) that were displayed one at a time before the dice were presented. These are preserved in case there was a correlation between incorrect answers and the letters that were displayed before the dice. (It turns out there isn’t one.)

The next four columns have to do with timing within the experiment. The ltime is the amount of time in-between the last letter shown and the dice presentation. This time is roughly half what it is between letters, thus catching the participant off guard when the dice are shown. This is yet another way to prevent expectations of what is coming, which could allow eye movements. The next two times have to do with how long the dice are visible for.

Ftime is how long the program asked to have the dice displayed for. Because Java bases its timing off of clock cycles, it is impossible to get exactly 50 millisecond presentation times. Instead, they are usually grouped in roughly 15 millisecond intervals. Thus, the times show up at 47 milliseconds. Ptime is the actual presentation time. Usually it is identical to ftime. However, in some cases it is too long, usually by one interval, (62 milliseconds). When this happens, the trial is repeated at the end of the experiment. Trials that are repeated are identified by the asterisk (*) at the end of the line. These trials are not included in data processing.

The final column is rtime, or reaction time. This is how long it took the participant to respond with a sum after the dice presentation is removed. Participants are given four seconds to respond. If they don’t respond within this time period, rtime is sent
to 0, an asterisk is added to the trial, and it is repeated at the end. These repetitions are why there are more than 24 trials in the output file, trials are repeated due to either program error (timing) or human error (no response).

The final line of the output is the `#end_of_file`, which contains statistics concerning the accuracy and correlation of the data in the output. %correct is the number of correct answers divided by the total number of trials (excluding invalid ones). Pt_mean is the mean average of the presentation times, and pt_std is the standard deviation of the presentation times. Rt_mean and rt_std are the same, but for the response times. Pt_rt corr is the correlation between presentation times and response times (out of 1). This was calculated to make sure that the minute differences in presentation time (46 to 47 milliseconds) were not affecting the response times of the participants (how quickly they were able to sum the values). A low pt_rt corr value means that there is no statistical correlation.

The next section will explain how these raw output files are processed to simplify and organize the data into formats that can be used with Cobweb, the machine learning program. Processing was also done to pull out higher-level features and basic statistics on the data.
4 Data Processing

The raw output data that came from the psychological experiment was reorganized into different arrangements for two main reasons. The most important being that the raw output files had to be transformed into a format that was compatible with Cobweb, the machine learning program. Because Cobweb is written in Lisp, it requires a certain formatting (involving parenthesis) for the data to be recognized. This process also allowed us to greatly simplify the number of attributes that were fed into Cobweb (this will be explained in a moment). Secondly, parsing allowed the gathering of averages and totals of patterns throughout the datasets. The pattern generalizations from Cobweb would therefore be complimented by a set of finite values.

In total, three parsers were written. The first parser dealt with Cobweb formatting, the second with generalized statistics, and the third with pattern totals. The first two parsers dealt directly with the raw output files, the third from the Cobweb formatted data. All of the parsers were written in the Perl programming language, a scripting language useful for processing data. All of the parsers were written by Marc Attiyeh, a senior computer science major.

4.1 Cobweb Parser

The most important step in creating the Cobweb parser was deciding which attributes from the raw output would be included in the data that would be used to find patterns. Because the focus of the research dealt with the die combinations and their
participant answers, they were immediately flagged for inclusion. Other values such as participant response time, dot contrast level, letters shown and trial position within the experiment were considered. Eventually only the contrast level was chosen. Cobweb considers all attributes equal; there is no way of specifying one as being more important than another. Therefore, we didn’t want Cobweb sorting by trial position or letters; they are secondary information to the goal of our experiments. Likewise the participants’ initials were not included, or else Cobweb could create a clustering pattern where each cluster is one participant. We were interested in global patterns, and therefore needed to make the data more fluid and anonymous in order to allow communal generalizations based on dice combinations and answers to float to the surface.

Contrast levels were added to help explain error rates. Because the faded dots are being used, it is important to be able to compare those figures to those where all the dots are relatively faded (by using a lighter contrast). However, a complexity decision was made that favored simplicity — individual fading values were not included. This means that although some datasets contained multiple fade levels (90, 100, 110), to Cobweb they were all the same. Dots were either on, off or faded (although the on values were relative to the contrast). This allowed Cobweb to create patterns by faded dots, but not within the fading spectrum. Future work might allow Cobweb to build more specific clusters, but at this time we felt single-level fading was sufficient.

In addition to the contrast and dot information, Cobweb was given a value concerning the participants answer. Instead of providing the correct answer, the participants answer and the difference (if there was one), the system was just given the difference. If the correct sum was 7, and the participant answered 5, the difference value
would be -2. This was done because Cobweb already could create patterns based on the sum value by the individual dot values that it had. Repetitive attributes would not bring any advantages to the system. Figure 21 represents the final data formatting.

Figure 21 - Cobweb data formatting.

In Figure 21, there is an example dice combination (using faded dots), and the appropriate Cobweb data string. The first value is the contrast for the ON ("black") dots. The next three values represent the left die face. First come the number of "black" dots, then the number of faded dots, then the total number of dots. As seen in the example, there are three black and two faded, making for a total of five. (3 2 5). The next three values are the same, but for the right die face. Four black and zero faded make four total. (4 0 4). The last value is the difference between the correct answer and the participant answer. In Figure 20 it is an X, because the participant answer is unknown. In a real data string the value would be zero if the answer was correct, a negative value if the participant under responded, and a positive value if their total is over the correct value. In
all circumstances, the correct value is considered the one that includes all the dots, including faded ones. Therefore in Figure 21 the correct answer is nine.

In its entirety, the Cobweb string has two parts – a data section (what is actually processed by Cobweb) and a label. The label created uses the identical data string, with the addition of the experiment name (BxBx) and the initials of the participant. This additional information allows us to see if any coincidental patterns involving experiments and users appear. Although Cobweb cannot sort directly by user or experiment, it does have the possibility to create patterns that fall along these lines based on the other attributes. The addition of this information into the label tag allows us to determine this from the Cobweb results.

These Cobweb strings are combined into a singular file, where each line represents one trial. Figure 22 shows the equivalent output file from Figure 20 in Cobweb formatting.

```
(EEM_BOB2_70_2_0_2_2_1_3_0 70 2 0 2 2 1 3 0)
(EEM_BOB2_70_4_0_4_1_0_0 70 4 0 4 1 0 1 0)
(EEM_BOB2_70_2_1_3_2_1_3_0 70 2 1 3 2 1 3 0)
(EEM_BOB2_70_4_0_4_4_0_4_0 70 4 0 4 4 0 4 0)
(EEM_BOB2_70_2_0_2_4_1_5_0 70 2 0 2 4 1 5 0)
(EEM_BOB2_70_6_0_6_1_0_1_0 70 6 0 6 1 0 1 0)
(EEM_BOB2_70_5_0_5_2_1_3_0 70 5 0 5 2 1 3 0)
(EEM_BOB2_70_4_0_4_2_0_2_0 70 4 0 4 2 0 2 0)
(EEM_BOB2_70_2_1_3_2_0_2_0 70 2 1 3 2 0 2 0)
(EEM_BOB2_70_4_1_5_2_0_2_0 70 4 1 5 2 0 2 0)
(EEM_BOB2_70_4_0_4_5_0_5_0 70 4 0 4 5 0 5 0)
(EEM_BOB2_70_2_1_3_4_0_4_0 70 2 1 3 4 0 4 0)
(EEM_BOB2_70_1_0_1_6_0_6_0 70 1 0 1 6 0 6 0)
(EEM_BOB2_70_2_0_2_4_2_6_0 70 2 0 2 4 2 6 0)
(EEM_BOB2_70_4_2_6_2_0_2_0 70 4 2 6 2 0 2 0)
(EEM_BOB2_70_4_0_4_2_3_3_0 70 4 0 4 2 3 3 0)
(EEM_BOB2_70_2_1_3_4_2_6_0 70 2 1 3 4 2 6 0)
(EEM_BOB2_70_2_1_3_5_0_5_1 70 2 1 3 5 0 5 1)
(EEM_BOB2_70_1_0_1_4_0_4_0 70 1 0 1 4 0 4 0)
(EEM_BOB2_70_2_0_2_2_4_0 70 2 0 2 2 4 0)
(EEM_BOB2_70_3_2_5_4_0_4_0 70 3 2 5 4 0 4 0)
(EEM_BOB2_70_6_0_6_3_0_3_0 70 6 0 6 3 0 3 0)
(EEM_BOB2_70_1_0_1_5_0_5_0 70 1 0 1 5 0 5 0)
```

Figure 22 - Data parsed into format for Cobweb.
The raw output file is parsed line by line, so that the strings in the Cobweb file appear in the same order as they do in the raw file. The parentheses around the data are necessary because that is the format Cobweb looks for in its input data. Parentheses are a common character in the Lisp language, and therefore make sense as defining instances in the data.

The next parser deals with general statistics formed from a group of trials.

4.2 General Statistics Parser

The general statistics parser is used to gain insight into the accuracy of the participants' answers within a set of experiments. The actual content of these sets are explained in the next chapter (Chapter 4 Datasets and Data Organization). From a set of raw input files, the accuracy percentages (how many combinations the subject got right) are combined to create a set average. The highest and lowest accuracy is also recorded. All together, the statistics parser returns a file with 3 values in it – the accuracy average, and the high and low value. The number of experiments that are included in these calculations is based on the number of raw output files that exist in the same directory as the parsing program.

The next parser deals with totals of individual data patterns within a larger dataset.
4.3 Totals Parser

The total parser was designed to accent the results from Cobweb with finite values. Basically, it calculates the number of occurrences of each answer of a dice combination. If a dice combination had a four on the left and three on the right, at 80 contrast with no fading (80 4 0 4 3 0 3 X) was shown a total of ten times. For these ten times there were three different answers – 0, +1 and -1 (or sums of six, seven and eight). Participants answered correctly 80% of the time, so there were eight occurrences of 80 4 0 4 3 0 3 0, with one each of the other two possibilities (80 4 0 4 3 0 3 -1 and 80 4 0 4 3 0 3 1). The totals parser formatting for this information would appear as it is in Figure 23, below.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>80 4 0 4 3 0 3 0</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>80 4 0 4 3 0 3 -1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>80 4 0 4 3 0 3 1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 23 - Output from totals parser.

Although this small example might seem trivial, when dealing with thousands of trials and tens of dice combinations, these totals are extremely useful. It should be duly noted however, that without the generalized clusters created from Cobweb, these totals are inconsequential. Cobweb creates the big-picture patterns that these totals augment with finite values. Although the information at the end is similar, Cobweb’s easy viewing and pattern extraction makes it a clear winner for data generalization.

The next chapter will discuss the division of data into four subsets that allow for specific goals within the hypothesis to be recognized. The parsers described above were
run individually on each dataset, so that the totals and statistics from each could be compared.
5 Datasets and Data Organization

From the psychological experiment, an enormous amount of data had been gathered. Each of twenty-seven participants performed ten experiments, in which every experiment contained twenty-four trials. This created a total of 6480 trials. Because the ten experiments contained different dice combinations, different contrast levels, and different faded dots, it didn't make sense to process all the data as a singular entity. More finite patterns could be found by comparing different experiments to each other – by seeing the progression of the participants' responses. Comparing individual experiments also allows us to test for learning across the participants' evaluation period, and to understand more precisely how fading contrast affects accuracy. Comparison also allows the testing of progressions of perceptions. It would enhance our hypothesis if we knew that perceptions were used more as experiments got harder (lighter dots and more fading). This would make sense psychologically, as perceptions are used more often in ambiguous situations when patterns are not readily apparent.

To this extent, the data was organized into four subsets – referred to from this point on as datasets. Some datasets contain a single experiment, some contain multiple. Put together, they represent a progression of participant accuracy (and therefore hardness of dice combinations) from easy to very hard. Some of the data is from the first round of participants, some is from the second, and some represents both. Every dataset was evaluated separately. Each was processed by Cobweb as a singular entity (so therefore they each have their own tree structure). The three parsers were run on every dataset, so the totals and statistics apply only within the data groupings.
The first dataset was a control. This was designed as a baseline — where one could see what the accuracy was, and what patterns developed with the displaying of the easiest dice combinations. The dataset contained the trials from only one experiment — BOB1. As explained in Figure 18, this experiment contains no faded dots, and has the darkest “black” dot value — 70. This makes for the easiest dots to see — black with no fading. The arrangements included in this dataset are every possible combination of the sums five, six, seven, eight and nine. These are all the possible combinations (excluding fading) that could happen within the range of our experiments. This dataset contains a total of 2016 trials from both subject pools. It allows the creation of a baseline reading across all participants.

The second dataset focused on comparisons of ON value contrasts. In addition, it uses single-level fading to increase the difficulty level. This dataset is a combination of two separate experiments, B1B4 and B2B2. B1B4 uses an ON value contrast of 80, while B2B2 uses an ON value contrast of 90. Both use a single level fading value of 100. These datasets will be used to see how contrast changes (and the use of slight fading) compare to the baseline results from the first experiment. It will be interesting to see how much the accuracy of the participants' changes with these modifications to the dice combinations. This dataset uses 552 trials from 12 subjects.

The third dataset focuses on the use of multiple levels of fading. It is very similar to the second dataset, except that instead of a single level of fading (100) it has three — 90, 100 and 110. It also uses two contrast levels. These are set to 70 and 80. Two different experiments make up this dataset (one for each ON contrast value). They are BOB3 and B1B3. In total, this dataset uses 288 trials from 6 subjects. This small subject
pool is due to mid-research changes to the second participant's group. The second half of
the group had slight changes to their experimental files, which resulted in having
experiments that were run on only six people. This dataset will allow the comparison of
multiple faded dots and multiple contrasts to singular faded dots and multiple contrasts
(from the second dataset). Although Cobweb cannot directly analyze multiple levels of
fading (it isn’t included in the attributes of the data), it can use the relative accuracy of
the participant responses to see how multiple fades affected the outcomes.

The fourth dataset includes the hardest to see dot combinations. Originating in
experiment B2B3, these trials utilize a single contrast level, and two levels of fading.
This single contrast level however is 90, which makes it the closest to the white
background (125) of any ON value. The two levels of fading are also the lightest (100
and 110). Combined, these three possible dot values are the lightest of any experiment.
Other experiments utilize 100 and 100 fading, but use a darker ON value (70 or 80). The
accuracy is expected to be the lowest in the this experiment, and we hope that this leads
to the greatest use of perceptions to aid in completing the summation task. There is also
an inherent risk of actual participant guessing. This happens when they have absolutely
no idea of what they’ve seen and just guess randomly. Some level of ambiguity is good to
enhance perceptual usage, but too much leads to complete guessing, which is not what we
want to have happen.

The next chapter will discuss the results of the research, when these four datasets
are parsed and run through the Cobweb conceptualization process. The created trees will
be analyzed, as well as the information gleamed from the totals and statistics parsers.
6 Results

In this chapter, the results will be displayed in several different formats. First the output of the accuracy parser will be explained. Next, the results from the four datasets will be displayed in four different formats. The first format deals with the occurrence rates of answers from the experiment. The second format shows the clusters that Cobweb found, and how they neatly group the data from within each dataset. Next, two different types of graphs are utilized. The first displays the occurrence rate of only incorrect answers for each dataset. This gives one an idea of any perceptual patterns that are visible. The second shows the same data, but by percentage instead of individual occurrence, so that results can be compared between datasets.

The results from the experiments are obtained through two different methods. Generalized patterns are found by through using Cobweb, and these are then recalculated using exact proportions using the totals parser. Together they form a very solid platform to support the patterns that were found.

From Cobweb, two distinct patterns were discovered. This was a surprise, as the psychological hypothesis only predicted the existence of one pattern. The first pattern validated the psychological hypothesis, it showed that the perceptual sum of sevens were one of the leading patterns in the data. The output from Cobweb strongly supported this pattern, and it was validated as well with the total parsers (this output will be explained in more detail in a minute).
The second pattern dealt with a different perceptual issue—correct object recognition. It appeared that a large number of participants were mistaking the number six for the number four. This is shown in Figure 24.

![Figure 24 - Participants often mistook the value six for the value four.](image)

There are two explanations for this incorrect pattern recognition. The first is simpler, and involves basic probability. The two extra dots that make a four become a six are only ever used in the number six. The four dots in the corners also appear on fours, fives, and partially in twos and threes. Therefore the likelihood that the value is actually a six and not one of these other values is statistically low. People might be ignoring these dots just because they appear far less often than the four dots in the corner.

The second possible explanation deals with Gestalt perception, as explained in Section 2.1.2. When people view the dice, they first build up the pattern that creates the white perimeter box (the die background). This is due to the ideas of continuity and closure. Because the four dots in the number four appear in the four corners of the box, they will be more easily recognized. Gestalt perception rules would suggest that these four dots are included in part of a bigger abstraction of the white background box, because they are equally placed in important junctions of smaller level objects. Basically,
because the dots are situated where the two sides join (in the corner) the brain will include them in its higher order perception of the box, thus greatly aiding in their recognition. However, the two extra dots that make up a six are not linked to the outside box, and therefore do not receive this special perceptual treatment. Therefore they can be considered "lost", floating in the blur that is not linked to the corners.

Combined, these two ideas provide adequate psychological reasoning to help explain why sixes were so often mistaken as fours. Although this pattern was not something that the psychologists were expecting to see, it can easily be explained through existing psychological theories.

The possibility also exists that both of these patterns can happen concurrently. This would happen when a mistaken six leads to the sum of seven. This is illustrated in Figure 25.

![Figure 25 - Both perceptual patterns happening concurrently.](image)

In this example, a six seen as a four leads to a sum of seven. This pattern was also found in the data, and is a conglomerate of the previously two stated patterns. Interestingly, it appears second most often, sandwiched between the 6's seen as 4's and the perceptual sevens. This makes logical sense, as it is a combination of the weaker and strong pattern.
Now that we have outlined the general patterns that were found across all the data, let us go into a detailed explanation of the results that came out of the four datasets. Remember that each of the four datasets were parsed and run through Cobweb separately, so that there would be no competing influences. We will start with a brief explanation of the accuracy across the datasets, which will set us up for the details of the perceptions of incorrect answers.

### 6.1 Analysis of Accuracy

Accuracy was measured by the extremes (high and low for one experiment), as well as the average mean for all the trials in the dataset. As expected, accuracy levels matched the level of fading that was used in the experiments. The first dataset, which contained no fading and the darkest ON value dots, had the highest accuracy (as seen in Figure 26). The second dataset had similar values. The third and fourth datasets had a large decrease in accuracy, as the faded dots got lighter, and were using multiple contrast values.

<table>
<thead>
<tr>
<th></th>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
<th>Dataset 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>100</td>
<td>95.63</td>
<td>83.33</td>
<td>83.33</td>
</tr>
<tr>
<td>Low</td>
<td>25</td>
<td>41.67</td>
<td>37.5</td>
<td>37.5</td>
</tr>
<tr>
<td>Average</td>
<td>74</td>
<td>72</td>
<td>62.5</td>
<td>59.7</td>
</tr>
</tbody>
</table>

**Figure 26 - Accuracy percentages of the four datasets**

These patterns of accuracy closed mimicked what we were expecting to see. It is interesting to note that only in the first dataset was there an experiment where a
participant got all of the answers right. In all of the other datasets, the highest single experiment accuracy did not reach 100%.

Next we will discuss the data that came out of Cobweb and the totals parser, and the division of dice combinations into clusters by the participants' answers.

6.2 Results, By Occurrence

The following series of diagrams will display the patterns from within the four datasets in two different configurations. The first series is sorted by occurrence. This means the string that appeared the most number of times in the dataset comes first, followed by the second and so on and so forth. Because each of the dice combinations were displayed an equal number of times (Because each participant completed the same experiments) the data is in effect sorted by answers. The most occurring answers (for each particular string) are appearing first. So the dice combination with the highest answer occurrence rate (where the most number of participants answered the same) will come first.

Occurrences were recorded down to the level where they were no longer statistically valid. This was determined by the answer occurrence rate being less than the number of times each participant viewed a combination. It would be assumed then that some participants were utilizing random guessing, and the patterns at such low occurrences were meaningless.
In order to aid in the comprehension of the data, the patterns explained above are highlighted. Each color represents a different pattern or non-pattern. The color scheme is shown in Figure 27.

<table>
<thead>
<tr>
<th>Color</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>correct answer</td>
<td>4 to 6</td>
</tr>
<tr>
<td>both patterns</td>
<td>sum of 7</td>
</tr>
<tr>
<td>blank</td>
<td></td>
</tr>
</tbody>
</table>

Figure 27 - Color schemes for varying patterns.

The five possible colors correspond to the five possible "patterns" in the data. The gray is for answers that are correct. Although these answers are perceptually boring, because no perceptions are affecting the system when the answer is correct, they are in most cases the most occurring answers, and therefore are important to recognize. As the accuracy in the early datasets is very high, correct answers are by far the leaders in occurrence rates.

The next three colors represent the three patterns explained above. Blue is for the incorrectly identified sixes. Yellow is for the perceptual sum of sevens. Green is when both patterns appear simultaneously. Finally, orange is for when there is no pattern in the data. This happens when commonly occurring wrong answers are neither of the three patterns. These occurrences do not fit any overall pattern that Cobweb found, and therefore are not important to our results.

The arrangement of the occurrence results goes from most occurring to least occurring. The first figure in this series is from the first dataset.
In this figure, the first column represents the occurrence position of the dice combination. This is provided so that readers understand that the second swath of values is simply a continuation of the first. The second column is the data, as described in the data section. The third column is the occurrence value. This is the number of times that that particular dice combination and answer appeared in the dataset. Based on the color scheme, it is easy to see that the correct answers appear first, followed by the 6’s to 4’s.
then the double pattern, and then the sum of sevens. Orange non-pattern combinations are also mixed in with the pattern data.

The next figure is for the second dataset. It shows a similar arrangement of the patterns within the data.

![Figure 29 - Occurrence data from the second dataset.](image-url)
The arrangement of the patterns is similar for all four datasets. The next two figures will illustrate this for datasets three and four. Below them is a summary of the information contained within all four figures.

Figure 30 - Occurrence data from the third dataset.
Together, these four figures illustrate the most commonly occurring answers to the dice combinations. The reason that some of these tables are much longer than other ones (contain more lines) has to do with the variance of the participant answers. If there were many different answers for each dice combination, then the table gets bigger. When there is more continuity amongst the participant responses, then the table shrinks.

Datasets that had more individual subjects (like the first one) have the highest chance of different answers, because there are participants, and therefore more random guessing. However, this is counteracted by the lighter dots in the later datasets, as they cause more random guessing – and therefore a greater range of answers.
In these four figures, the information is sorted by occurrence allowing one to get an idea of the general flow of the patterns within the datasets. The next dataset organization will be sorted by clusters, mimicking the patterns that came out of Cobweb.

6.3 Results, By Cluster

Sorting results by cluster instead of occurrence paints a different picture of the output. Instead of being arranged by the number of times a certain combination and answer appear in the dataset, the information is arranged by dice combination. Cobweb clustered first by combination, and then by answer. Looking at the datasets this way allows one to see the range of answers for each combination. Usually (but not always), the correct answer is the most common, followed by one of the three major patterns.

In the figures below that show the clusters for the four datasets, it is important to recognize that not all of the possible answers are shown. In some cases, the number of values for each answer for a dice combination will not add up to the total number of occurrences. This is because once again the statistically insignificant random guessing occurrences have been excluded. This has been done to decrease the amount of information that one has to look at. Although this will cause some slight adding inconsistencies, one can see where the random guessing values are missing by doing some simple math if one is curious.

The arrangement of information is the same as it was for the previous four figures. The first column represents the string placement in the data. The information is sorted by clusters, from the smallest left die value to the greatest. Therefore dice
combinations with a one on the left appear first. The combinations appear in the second column, followed by number of occurrences for that particular combination and answer in the third. In almost every example, there is at least one correctly identified combination. Therefore one could consider the gray patterns as being the boundary between clusters. First the correct answer will appear (the gray bar) followed by all of the other incorrect answers for that particular dice combination. The next pattern starts with the next gray bar, signaling the start of a new combination, and a new cluster.

This is true generally speaking; however there are some clusters that only have incorrect answers. This is particularly true in the third datasets, where there are only six subjects. In this set, some of the dice combinations have only incorrect answers. One must be careful when looking at the patterns in that dataset, as gray bars do not always break up the clusters. Below are four figures, one for each dataset, grouped by clusters.
Figure 32 - Conceptual clusters for the first dataset.
| Conceptual Clusters | Figure 33 - Conceptual clusters for the second dataset. |
Figure 34 - Conceptual clusters for the third dataset.
Figure 35 - Conceptual clusters for the fourth dataset.

These four figures are nearly identical to the tree structure that Cobweb created. Because Cobweb was using a hierarchal construct, there is some fuzziness between the depths of the children nodes. Therefore the patterns are not as exacting as they appear to be in the previous four figures. This is due to the acuity values that determine the placement of the data. Because the acuity value is fixed, in some situations it causes splitting and merging of nodes that is not optimal for that particular subtree. However, as a whole these conceptual clusters mimic the Cobweb trees very closely.

Put together, these eight figures allow an excellent detailed view of the results from the datasets. However it is hard to look at them and pull out generalized
information. It is hard to see if there is any progression in the strength of the three patterns within the four datasets. Therefore a third method of results analysis is needed.

6.3 Results, By Graph

Graphing results is a way to show a lot of information in a very concise and informative visual manner. Although the two tables for each dataset provide a plethora of particulars concerning occurrences and clusters, they are not very easy to look at when trying to compare the datasets. The goals of the following four graphs are to be able to clearly show the progression of the three patterns across the four datasets.

To aid in this ease of viewing, all of the correctly answered combinations were removed. This allows the reader to focus on what is perceptually interesting, the three prevailing patterns, and their progression through the dataset. There are two graphs for each dataset. They contain the similar information, the difference being the value of the Y axis. In the first graph, the values are the number of occurrences of that particular combination and answer. This allows a comparison within the data, and one gets an idea of the actual number of occurrences of each data string. The second graph is in percentage, as compared to the total number of occurrences for each dice combination. This is similar to the cluster tables, where one can see how often one answer appears as compared to other answers for the same dice combination.

Using percentages allows the generalization across all four datasets. It then doesn’t matter that one dataset has more total occurrences than another, because the
occurrence rates have all become relative. This is quite helpful when trying to illustrate the relative changes from one dataset to another.

In all of these graphs, the color schemes are the same as they were for the previous tables. Refer to Figure 27 to refresh your memory if necessary. Figures 36 and 37 on the following two pages are the graphs for the first dataset.
Figure 36 - Graph of occurrences of only incorrect sums in the first dataset.
Figure 37 – Graph of the percentage of incorrect occurrences in the first dataset.
In these two figures, the same overall pattern is apparent, this time a little more clearly. The sixes mistaken as four appears first, followed by the green combination pattern, and then the yellow perceptual sevens. Figure 37 illustrates that the most commonly occurring incorrect answers accounted for less than half of the total number of answers for their respective dice combinations. This means that the patterns are visible, but they are not too strong. A large number of correctly identified combinations are suppressing the need for perceptual cues.

In the next two figures (38 and 39) the same information will be shown for the second dataset. Notice how the pattern arrangements remain the same, but their persistence has increased – the occurrence rates and percentages have grown stronger.

![Occurrence Rate of Incorrect Responses, 2nd Dataset](image)

*Figure 38 - Graph of occurrences of only incorrect sums in the second dataset.*
In these two figures, the patterns have grown to consume fifty percent of the answer occurrences. The number of non-pattern occurrences (the orange) has also been greatly reduced. Overall, the accuracy rate is very similar to the first dataset (72% vs. 74%), nevertheless within the incorrect answers, the patterns have increased. However, because this data comes from only twelve subjects, it contains a much higher level of individualism than the first dataset, which decreases its validity. This is due to the fact that individual preferences can start to affect the overall generalizations when the number of participants is small. In order to truly validate these patterns, the same experiments would have to be run on a larger subject pool.

The next two diagrams graphically represent the data from the third dataset. It is important to remember that the accuracy for this data overall is much lower than the first
two datasets. It drops from the mid seventies to the low sixties. This causes much more random guessing and extraneous “patterns” (the orange bars). However it also raises the predominance of the three noted patterns. As we projected, the harder the experiments get, the more perceptions come into play, thus increasing the strength of the patterns. Figures 40 and 41 below demonstrate this.

![Graph of occurrences of only incorrect sums in the third dataset.](image)

**Figure 40** – Graph of occurrences of only incorrect sums in the third dataset.
Figure 41 - Graph of the percentage of incorrect occurrences in the third dataset.

In these figures, the blue (six to four) pattern becomes stronger than it has in the other three datasets. This is because of an increase in the use of 4-2 fading. This means that the extra two dots that make up the six (the middle two) are being faded (so that the die looks even more like a six). This increases an already strong perceptual image, based heavily on the Gestalt principles.

Overall, the patterns have all become more prevalent. They have jumped from roughly fifty percent to slightly over eighty. This means that eighty percent of the time, when presented with a certain die combination, the participants answered the same – and answered incorrectly. This is an amazingly high percentage considering the simplicity of the task, and the basic levels of perception that are being tested. As with the second dataset, it is important to note that the number of participants in the third dataset is far
less than the first. Although this does lead to individualism becoming more apparent in
the data, there are still some very strong patterns that circumvent individual decision
making, and allow for generalizations.

The final fourth dataset contains the hardest to view dots. They use multiple levels
of fading, as well as the lightest ON contrast values. Although the overall accuracy drops
only slightly from the third dataset (to 58%), the greater number of participants makes
these results more valid. Figures 42 and 43 illustrate the findings from dataset four.
Figure 42 - Graph of occurrences of only incorrect sums in the fourth dataset.
Figure 43 - Graph of the percentage of incorrect occurrences in the fourth dataset.
There is a fair amount of orange bars in the fourth dataset – due to the increase in fading. Most of these bars are caused by a secondary pattern similar to the six to four. Often, a three is mistaken as a two when the central dot is faded. This follows similar Gestalt guidelines as the six being seen as a four. Because the two corner dots are aligned with junctions in the higher order image, they are easier to remember. The faded central dot gets lost as it has nothing to perceptually hold on to. This less dominant pattern only becomes somewhat apparent in the fourth dataset, and therefore is not as important as the 6’s to 4’s pattern, which is seen across all four datasets.

In it interesting though, that although 3’s with a faded center dot are often incorrectly identified, people do not have a problem with the number five. It as well uses the center dot that can be perceptually lost in the middle. Perhaps, with all four corner dots, the center dot is more easily recognized. Although Gestalt principles do not have any strong arguments to explain why this happens.

It is important to remember that we have only highlighted three of the major patterns that exist in the data. These are the major conceptual clusters, as identified by the Cobweb output. There are however many secondary patterns that exist. Remember that seven lies in the center, the apex of the normal curve. It is the most occurring combination. However, the sums of six and eight are only slightly below seven in the probability of appearing. Therefore secondary patterns with incorrect answers leading to six and eight should also exist in the data. Because our dataset is not large enough to truly evaluate these two secondary patterns, they have not been mentioned previously. Although if one was to look at the orange non-pattern bars, a fair number of them are
these incorrect sixes and eights. With a larger dataset covering the full spectrum of sums (instead of just five through nine) a broader analysis of these patterns would be possible.

These secondary patterns help explain the increase in orange non-pattern bars in the fourth dataset. These bars however are also covering the random guessing that occurs when die acuity decreases. Although a larger number of the incorrect answers are guided by perception, some of them are simply wrong – most likely caused by random participant guessing. When a larger number of participants are shown the same dice, there is a small probability that some of the random guessing will lead to what appears to be patterns, when in fact they are just background noise. This explains some of the lesser occurring orange bars towards the right hand side of the graph.
7 Future Work

There are several areas in which I would like to further this research. Perhaps the most interesting would be to change the subject pool. If one used a subject pool that had different perceptual knowledge about dice, then one would hope that the perceptual patterns would change to reflect this knowledge.

Such an experiment would be possible if compulsive gamblers were used as participants. In games involving the rolling two die, when the payout in attributed to a number different then seven, one would expect a different set of perceptual cues. Because the gamblers would want that combination to appear the most number of times, their normal curve would be skewed in that direction. This would (hopefully) cause them to start incorrectly summing dice to reach this number.

Applying different perceptual knowledge to an otherwise identical experiment would further validate our psychological hypothesis. It would allow us to further prove that perception plays a large part in decision making when using semi-ambiguous visual stimuli. The concentration of incorrect answers around a value different from seven would show the flexible and individualistic nature of our perceptual systems.

A second goal of future work would be to further validate the normal curve as mentioned at the end of the results section. If we used the full unrestricted curve, from sums of two to twelve, then we would be able to have a much larger set of data. This would allow us to see if the sums of six and eight were also being incorrectly summed to.

One of the major problems with doing this however is the raw number of combinations each subject would have to see. With our ten experiments done within an
hour subjects were already complaining of boredom and the task being too long. The increase in experiments needed to construct the full normal curve would probably exceed the capacity of our subjects to say interested. The last thing we want is for them to get tired and start randomly guessing at combinations, as the data becomes useless. It may be possible to combat this problem using multiple sessions with the same subjects. This would allow us to gather more data, without having the subjects get tired and bored.

A third direction that future work could go in would involve the machine learning aspect of this project. Cobweb has now been validated as an excellent pattern finding mechanism for psychological data. It is relatively simple to configure, and its output is understandable to non-computer scientists. Applying this conceptual clustering engine to different sets of psychological data would perhaps allow further discoveries of unexpected results. As the psychology department is always running experiments and always gathering more data, Cobweb could be integrated into their toolsets for gathering information about their data. Although many psychologists might be hesitant to try something new, the success of this project might sway them.
This research concluded at a level that is satisfying from perspectives of both the
computer science and the psychology. The results verified the psychological hypothesis,
as well as provided new information. This is truly a best of both worlds situation –
confirmation of previous ideas and the creation of new ones.

The machine learning system performed as expected, and provided clear and
accurate conceptual clusters that neatly displayed the patterns within the psychological
data. Cobweb provided hierarchies of clustering, allowing patterns to be analyzed at
many different levels. These patterns were extrapolated back into the field of psychology
to help verify their ideas.

Furthermore, this project laid the groundwork for interdisciplinary applied
computer science, and presented a toolset that was both robust and flexible. One can only
hope that other departments and research experiments will take advantage of what
applied computer science has to offer.

Overall, this project was an exciting opportunity to work in an interdisciplinary
setting. It allowed me to excel both in the fields of psychology and computer science, a
combination that I plan on continuing. The research focused on areas within these two
disciplines that I was equally independently interested in – cognitive psychology and
machine learning. Combined, it gave me chance to complete an extensive yet reasonable
project within the time span that I had. Future work has been noted, in the hope that a
future student interested in this subject matter can pick up where I left off. In the end,
although my honorable efforts will not appear on my diploma, I will leave Colby with an
independent research project that glorifies both my academic interests and exemplifies the research skills that I have learned in my four years of undergraduate education.
References:


