Multiplying 10-digit numbers using Flickr: The power of recognition memory

Multiplying 10-digit numbers using Flickr: The power of recognition memory

Andrew Drucker*

1 The recognition method

In this informal article, I'll describe the "recognition method"—a simple, powerful technique for memorization and mental calculation. Compared to traditional memorization techniques, which use elaborate encoding and visualization processes [1], the recognition method is easy to learn and requires relatively little effort.

There is a catch: to apply the method, you need continuous access to a large collection of interesting, unfamiliar photographs. (This is easy to set up on a home computer, using the photo-sharing website Flickr.com, and I'll explain how.) Now, some might object that this is cheating, since the photos act as aids to memory. However, the method doesn't alter or rearrange the photographs during the process, so their role as aids is very limited compared to pen and paper.

The basic idea for the method is not new; it comes from the psychological literature [2], and has been applied by computer scientists to the design of computer password systems [3, 4]. However, it seems not to have been explored before as a systematic mnemonic device, or as a tool for mental calculations.

*MIT CSAIL, Cambridge, MA, 02139. Supported by a DARPA YFA grant of Scott Aaronson. Email: andy.drucker@gmail.com

The method *works*: using it, I was able to mentally multiply two random 10-digit numbers, by the usual grade-school algorithm, on my first attempt! I have a normal, untrained memory, and the task would have been impossible by a direct approach. (I can't claim I was speedy: I worked slowly and carefully, using about 7 hours plus rest breaks. I practiced twice with 5-digit numbers beforehand.)

I chose multiplication as a widely-familiar problem that many people can solve with pen and paper. However, the recognition method can be applied much more broadly. Although I haven't attempted this, it seems that the method could be readily used to mentally simulate general models of computation, such as *Turing machines* [5]. Of course, human memory *is* finite, and this puts a bound on the size of mental computations we can perform; but the recognition method suggests that our brains' capacity for precise computation is much greater than we might naively expect.[1]

My main goal here is to expose more people to the amazing power of recognition memory, and to describe a concrete application (mental multiplication) that readers can try at home. In the last part of the article, I'll also speculate about the ultimate power of the "recognition method," and share some ideas about a wacky new kind of memory competition based on recognition memory—a competition that could astound, entertain, and perhaps even yield new insights into the human mind.

Recognition tasks

Suppose I have a collection of *pairs* of images. From each pair, I select one at random to show to you. Afterwards, I choose a random pair, show you both images, and ask you to identify the one you've seen before. (Note that we are testing *recognition*, as opposed to a more active *recall* of features from the image.)

[1] Conventional wisdom suggests that human brains are inferior to computers in at least two respects. First, we're slow and error-prone when doing basic logical operations; second, we have a feeble capacity for storing "meaningless" numerical data in working memory.

The recognition method is a powerful "hack" to partly overcome this second weakness. Can similarly powerful hacks be found to improve our speed and reliability at logic? I find this question fascinating.

It turns out that ordinary people are incredibly good at this task. In one of the most widely-cited studies on recognition memory, Standing [2] showed participants an epic 10,000 photographs over the course of 5 days, with 5 seconds' exposure per image. He then tested their familiarity, essentially as described above. The participants showed an 83% success rate, suggesting that they had become familiar with about 6,600 images during their ordeal. Other volunteers, trained on a smaller collection of 1,000 images selected for vividness, had a 94% success rate.

As you'd expect, recognition memory works best when the images being learned are interesting and sufficiently distinct. Also, similar training processes based on recognizing word combinations have somewhat less impressive success rates. A number of studies suggest that our *visual* recognition memory is particularly strong [6].

Exploiting recognition

In recent years, Standing's and related studies helped inspire an interesting line of work in an unexpected area: computer security. By familiarizing an individual with a randomly chosen subset of images from a fixed collection, a computer program can create a form of *knowledge* held by that individual alone. This knowledge can then be used to reliably identify that person. In other words, recognition memory can serve as a basis for *password* systems (see [3] for a survey). For example, in the commercially-available Passfaces system [4], a user's "password" is a set of familiar faces.

Passfaces and related systems use, either implicitly or explicitly, a "recognition method" that allows numerical data to be encoded into a user's familiarity with certain images. The basic idea is as follows. Say that we want to memorize 5 numbers, each in the range 0-9. Let's call these numbers $n_0, n_1, \ldots, n_4$. (It's most convenient to index starting from 0.)

We acquire 50 previously-unseen photos, and label them with numeric indices 00 to 49. We think of the photos $00, 01, \ldots, 09$ as forming a "memory cell" which is to "store" our value n_0. Similarly, the photos $10, 11, \ldots, 19$ form a cell which will store n_1, and so on. To store n_0, we first inspect its value. For concreteness, suppose $n_0 = 4$; then we look at the photo 04, and familiarize ourselves with it:

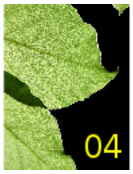


Figure 1: Photo 04.

Now n_0 is "stored," and we go on to store n_1. If $n_1 = 7$ (say) then we look at photo 17. We store the other values similarly.

Later on, when we want to retrieve n_0, we just look at photos $00, 01, \ldots, 09$:

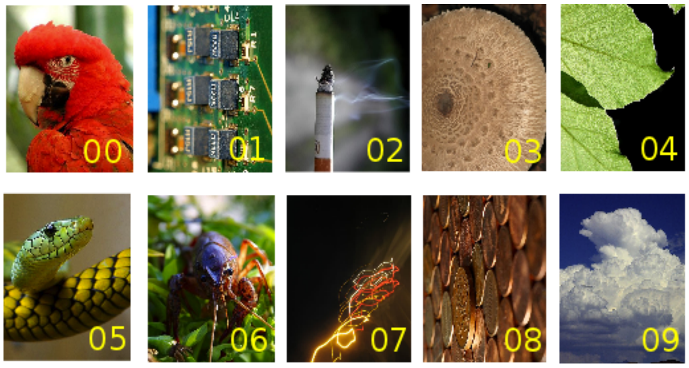


Figure 2: Photos 00-09. One was seen before.

By recognizing the image 04 as familiar, we infer that $n_0 = 4$.

Note that, while retrieving n_0, we may become familiar with some of the other images in the range $00, \ldots, 09$. This may degrade or destroy our "encoding" of n_0. Now in practice, I've found that this is usually not a major problem: if we study the desired image intently during the encoding process, and pass quickly over the others during retrievals, we can reuse an encoding multiple times. But there is also a more reliable fix: we can make multiple copies of our encoding into auxiliary cells, and use each copy only once. (The price is that a larger number of photos and cells are required.) Happily, in the mental-multiplication procedure I'll describe, each stored value only needs to be retrieved once.

2 Mental calculations with the recognition method

We're now ready to see how to multiply large numbers with the recognition method.

Our approach is based on the usual multiplication algorithm taught in American schools. I won't review the method in any detail, but here is a quick, worked example. In Phase 1, we multiply the top number by each of the digits of the bottom number:

$$\begin{array}{ccccc}
 & & 3 & 2 & 7 \\
 & & \times 2 & 4 & 6 \\
\cline{3-5}
 & 1 & 9 & 6 & 2 \\
1 & 3 & 0 & 8 & \\
6 & 5 & 4 & &
\end{array}$$

Figure 3: After Phase 1.

For example, the middle row is obtained as $1308 = 327 \times 4$. In Phase 2, we then pad with zeros and add up the numbers from Phase 1, as follows:

$$\begin{array}{ccccc}
 & & 3 & 2 & 7 \\
 & & \times 2 & 4 & 6 \\
\cline{3-5}
 & 1 & 9 & 6 & 2 \\
1 & 3 & 0 & 8 & 0 \\
6 & 5 & 4 & 0 & 0 \\
\hline
8 & 0 & 4 & 4 & 2
\end{array}$$

Figure 4: After Phase 2.

(I haven't shown the "carry"-digits in this example.)

Now let's see how to perform the same multiplication using the recognition method. The following important diagram will be our guide:

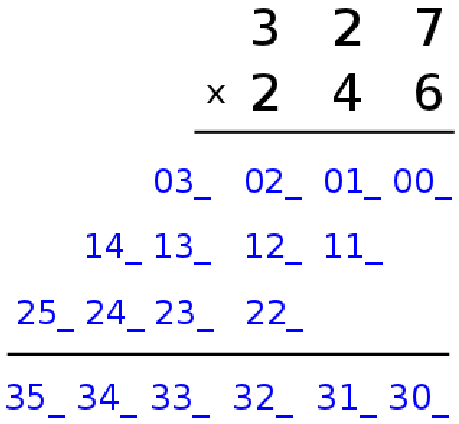


Figure 5: A visual guide to our method.

Here we've taken all the positions in which information would be stored by the pen-and-paper method, and assigned these positions "indices." The indices are coordinates, using the top-right storage position as our reference. For example, the index "12_" is 1 step down, and 2 steps left from the top-right position.

As you might have guessed, each index will become a "storage cell" of the kind described earlier. We begin with a collection of 360 images, labeled 000 to 359, which will correspond to cells 00_ through 35_. To mentally multiply these numbers with the recognition method, we follow the ordinary multiplication method *exactly*. The only difference is that, whenever we would ordinarily write a number into a position, we will instead encode that number into our recognition memory, by the method described earlier. For example, to "write" a 0 into the position corresponding to cell 12_, we would just familiarize ourself with image 120. Similarly, whenever we would ordinarily *read* a number from a position, we will instead look at the ten images of the corresponding cell, and see which one is familiar.

Let's walk through the first few steps of the procedure applied to the product $\begin{array}{r}327\\ \underline{\times 246}\end{array}$, with reference to Fig. 5. To fill cell 00_, we mentally compute the product 6 × 7 = 42. We store the 2 in cell 00_ by looking at image 002.

We then carry the 4. I use my fingers to store such carry-digits, allowing myself this most human of memory aids. (It's best to have a one-handed sign for any number between 0 and 10; you can use the Chinese method, for example [7].)

We go on to compute $6 \times 2 = 12$, plus 4 equals 16. So we store the 6 in cell 01_ by looking at image 016, and we carry the 1. In this way, we carry out all the multiplications in Phase 1.

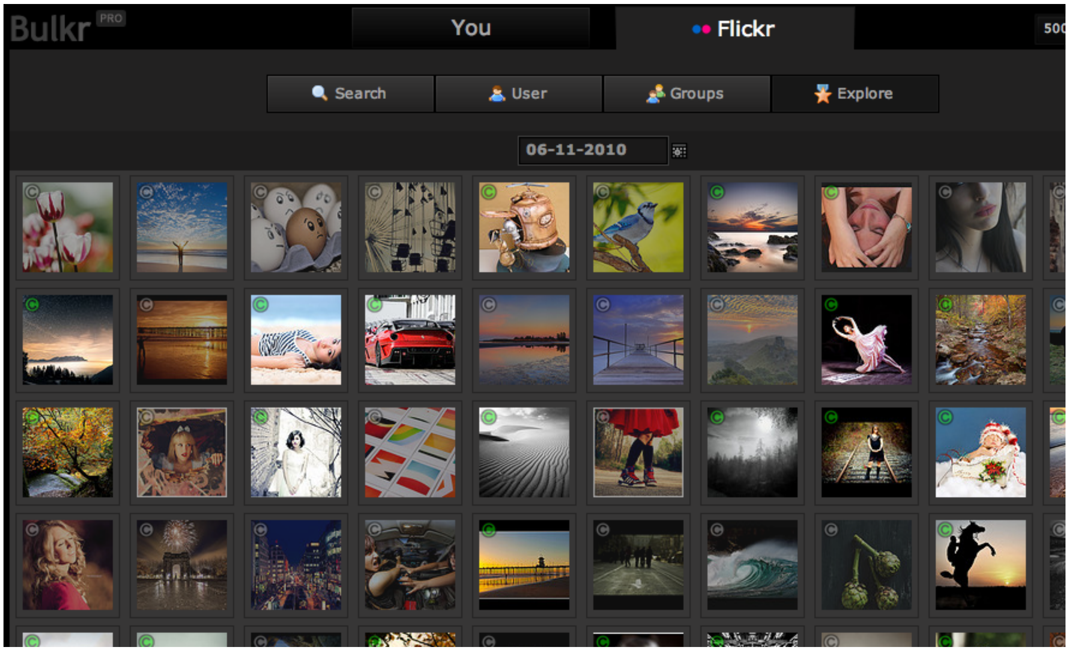
In Phase 2, it's time to retrieve the stored values and add them up, column by column, to fill out the final row. For example, in the rightmost column, there is only a single term to be added, which can be found in cell 00_. So we just look up that value and transfer it into the bottom-row cell 30_. To fill cell 31_, we add up the contents of cells 01_ and 11_. As it happens, we end up with a 1 to carry into the next column. We proceed in this way until we've filled the bottom row; this is our final answer, and can be read off one digit at a time by inspecting the cells.

That's the scheme in a nutshell. But how to set up the needed photo library? And, is it really possible to carry out a large multiplication? If so, what tricks are helpful? These are the questions I'll address next.

Getting a Photo Library

Setting up a photo library breaks down into two tasks: first, assembling a large number of images (hopefully interesting and memorable ones); second, renaming them in numerical order to organize them into "cells."

To get my images, you can use the photo-sharing site Flickr.com, where thousands of new photos are uploaded daily. Step one is to get a free Flickr membership. Next, a free program called Bulkr (Win/Mac/Linux) allows you to bulk-download images from the site. Bulkr will only download a photo whose owner's permissions/copyright settings on Flickr allows downloads; my understanding is that it's legitimate, legal software.



06-11-2010

Figure 6: The Bulkr interface.

Bulkr has an “Explore” feature that brings up the most “interesting” photos uploaded on any given day, as voted by Flickr users. I paid to upgrade to the full version of Bulkr, to access more of these photos and download hundreds at a time.

To rename the files with numerical indices, I used the Automator tool on my Mac. It has a function called “Make Finder Item Names Sequential” that does the trick nicely. For Windows, a program called Bulk Rename Utility seems to do what we want, and for Unix-type systems, there are a variety of options.

Making the scheme work

If you’ve read this far, perhaps you’re even curious enough to try the method. If so, here are some tips to maximize your chances of success.

1. Be wary of not-so-memorable images. I've found that, by and large, the photos I download from Bulkr are of good quality and suitable for the task. But there are some pitfalls. Landscape photos can blur together in memory, as can photographs from a specific niche—for instance, there's a ludicrous amount of Lego photography on Flickr.

The simplest way to address this problem is to spend more time familiarizing yourself with photos you consider less memorable. A more careful failsafe is to build a "backup" collection of images, for use when you hit a dud photo. I used this technique for my 10-digit multiplication task.

2. Familiarize yourself well with the cell-index system you'll be using. It's especially important to be able to easily determine into which cell a particular value is to be stored.

If you find it confusing to locate cells, it might make matters easier to create a different folder on your desktop for each cell, and arrange them according to their use in the multiplication procedure. If my cell-indexing scheme doesn't work for you, invent your own.

3. Make sure you can remember what you need to remember. I don't mean remembering the images you've seen—you'll find that this part is surprisingly easy. The challenge is to keep track of where you are in the algorithm, while also holding on to carry-digits and (in Phase 2) the running column-sums.

I found that with 5-digit numbers to multiply, this was possible with reasonable concentration. For my 10-digit challenge, I used a simple observation to reduce the task to four 5-digit multiplications, plus a final addition (all without pen and paper). For example, we can write

$$\begin{aligned} 1111122222 \times 3333344444 = {} & (11111 \times 33333) \times 10^{10} \\ & + (11111 \times 44444 + 22222 \times 33333) \times 10^{5} \\ & + \ 22222 \times 44444. \end{aligned}$$

If you find the approach I've sketched places too much demand on your working memory, it's possible to enlist additional image-cells to help keep track of carry-digits, running sums, and your place in the algorithm.

Putting it to the test

On a sunny California day in June 2011, I grabbed some numbers from a table of random digits. In 7 hours of work, I used the recognition method to calculate

$$9883603368 \times 4288997768 = 42390752785149282624.$$

I have no special abilities or memory training, and what I did was not really a feat. I firmly believe that an average person who knows how to multiply could do the same thing with a little training (and do it much faster, with practice). Also, as I've said, the technique is hardly limited to arithmetic—you can choose your own wild application. As one possibility: for those familiar with *cellular automata* (CA), I would love to see someone mentally simulate a long run of Conway's Game of Life, or some other interesting CA rule, with this method.

3 Toward new memory feats

The recognition method can enable ordinary people to use their memory in startling ways. I hope readers will be inspired to see for themselves. But I admit, I'm also very curious what the real memory *fiends* out there could do with the technique. Memory competitions are a thriving concern today; for an entertaining account (which helped inspire my own project), I recommend a recent book by Joshua Foer [1]. As an example of what the very best mnemonists (skilled memorizers) can do: recently Wang Feng of China became famous in his country by memorizing 480 random digits in 5 minutes, easily a new world record [8]. Interestingly, most of the top mnemonists seem to have no innate, savant-like abilities; their achievements are the product of technique and training.

Now suppose we allowed competitors to view the digits, encoded as *images* by the recognition method, and allowed them to decode the digits from memory using the same collection of images. How much faster could they take in information? I believe the answer is *much* faster, and I'd love to see this confirmed. It'd also be interesting to know whether current mental-arithmetic records could be broken with the aid of image collections. (Of course, this would be a distinct category of competition. I should mention that mental-arithmetic champions can already do 10-digit multiplications in a matter of minutes, without image aids.)

Such contests could have a side benefit to psychology: by allowing competitors to choose their own image collections, we could see a drive to identify the most memorable imagery available. This could help scientists to better understand what features are likely to make an image memorable. Now, there's been significant scholarly study of image memorability, some of which employs familiarity-testing very similar to the recognition method—see [6, 9, 10]. However, so far this work seems to have been done with ordinary test subjects, not trained mnemonists; and the images used have been supplied by the researchers, not the subjects. I believe that injecting an element of competition into this research could be a fruitful (and entertaining) direction.

The CAMRA game

For the task of memorizing raw data, I'd like to propose another new category of competition—one that generalizes the recognition method, and could be even more interesting than the competition I suggested above. I'll call this new proposal the *CAMRA game*, short for *Computer-Aided Memorization and RetrievAl*.

In this competition, a contestant gets to bring along two computer programs of her choosing; we'll call these *Trainer* and *Retriever*. To be concrete, let's assume the task is to memorize 100 random digits, in as little time as possible. (What I mean by "memorize" is a bit unusual, as you'll see.)

To set things up, the referees provide 100 randomly chosen digits as input to *Trainer*. Next, the game works as follows:

Stage 1 ("Training"): The buzzer sounds, and timing begins. The contestant interacts with *Trainer* until she feels ready for the next stage. (The idea is that *Trainer* is teaching her the random digits, perhaps in an altered form.)

Stage 2 ("Retrieval"): The contestant moves to a second computer, where she interacts with *Retriever*. This second program has not been given access to the random digits, and cannot communicate with *Trainer*. The decoding stage ends when *Retriever* outputs a sequence of 100 digits.

The contestant is successful if *Retriever*'s output exactly matches the random input digits to *Trainer*. Her time used could be measured, either as the time used for both stages, or just the time used in the Training stage.

So what's the point? In this game, the human contestant acts as the only "bridge" between *Trainer*, which knows the random digits, and *Retriever*, which does not. Thus if she can get *Retriever* to output the correct digits, she can meaningfully claim to have "memorized" those digits—albeit in a form that might be very indirect.

I believe that holding competitions for the CAMRA game could have significant scientific value. Here's why. When mnemonists memorize random data (without computer assistance), they typically perform all kinds of imaginative visualizations to transform the random data they want to learn into meaningful, memorable ideas. The trouble is, we can't directly *see* any of this; to learn about mnemonists' techniques, we are largely dependent on their verbal accounts.

One way to think about the *Trainer* program's role in the CAMRA game is as a partial *externalization* of the process by which information is made memorable. The hope is that, by studying the kinds of *Trainer* programs that skilled contestants develop, we might get a more direct glimpse into how this meaning-making activity works. (Of course, the distinctive strengths and weaknesses of computers will also strongly shape the way the game is played.) Also, since the *Trainer* program should help make random data more meaningful, the human contestants should be able to spend more time *absorbing* data (rather than imaginatively transforming/encoding it). Thus the CAMRA game could be one way to "pry apart" the activities of encoding and absorbing information, helping us to study these activities separately.

But I'll admit—mostly, I hope mnemonists try CAMRA because I think it would be fun to watch, and because I'm curious what skilled play would look like. One possible strategy for the CAMRA game is to make *Trainer* and *Retriever* share a collection of memorable images, and implement the recognition method as I've described it. But this could be problematic: if a contestant uses the recognition method at blazing speed, she will very likely make a few mistakes. A certain fraction of mistakes can actually be *corrected*, however, with a little more sophistication. The basic idea is that *Trainer* can use the recognition method to teach the contestant, not the random digits themselves, but an encoding of those digits under an *error-correcting code*—a powerful mathematical tool that's widely used in electronic communications.

Beyond these preliminary ideas, I'd love to see what new innovations contestants might bring to the task. An especially interesting feature of such competitions is their potential for collaboration: a contestant might develop her *Trainer*/*Retriever* programs with the aid of coders and psychologists.

A purely-mental recognition method?

Dedicated mnemonists might also try to take the recognition method in a different direction: it is at least conceivable that the basic technique could be used for memorization, even *without* access to an image library! How could this work? The hope is that with training, there could be a way to mentally "translate" sequences of numbers into meaningful, memorable mental images, in such a way that even very similar sequences would translate to sharply distinct images. With this ability, one would simulate the recognition method as described earlier, with a "virtual" collection of images.

This idea is not without precedents. Designers of computer password systems have proposed algorithms to automatically translate random numbers into memorable images, in order to apply the recognition method for user authentication [11, 3]. It seems that these algorithms would be very difficult to simulate mentally, however.

Classical mnemonic techniques are also built around the idea of translating numbers into memorable images, which mnemonists then try to actively recall [1]. Could some of these visualization practices could be adapted for use with the recognition method? If so, could such a method have advantages over existing mnemonic techniques? I consider this an interesting open question.

Conclusion

Human recognition memory is a marvel, a great gift we all share. I would like to exhort readers to see for themselves by trying the recognition method: Learn π's digits with imagery! Play the CAMRA game! Go forth and (mentally) multiply!

But ultimately, while I hope that the proposals in this article get explored further, they're just a few possibilities among many. I hope readers will be inspired (whether as researchers, as competitors, or as curious individuals) to find more exciting new ways to explore the power of human memory.

Acknowledgments

I'm grateful to Scott Aaronson, Luis Miguel Dickson, and John Ervin for helpful comments—with extra thanks to Scott for encouraging me to "actually do it."

References

[1] Joshua Foer. *Moonwalking with Einstein: The Art and Science of Remembering Everything.* Penguin Press HC, 2011.

[2] Lionel Standing. Learning 10000 pictures. *Quarterly Journal of Experimental Psychology*, 25:207–222, May 1973.

[3] Xiaoyuan Suo, Ying Zhu, and G. Scott Owen. Graphical passwords: A survey. In *21st IEEE Annual Computer Security Applications Conference (ACSAC)*, pages 463–472, 2005.

[4] Passfaces (website). http://www.realuser.com/.

[5] Turing machine (wikipedia article). http://en.wikipedia.org/wiki/Turing_machine.

[6] W. Howard Levie. Picture recognition memory: A review of research and theory. *Journal of Visual/Verbal Languaging*, 8:6–45, 1988.

[7] Brendan Connal. How to count to ten on one hand (in Chinese). http://www.instructables.com/id/HOW-TO-COUNT-TO-TEN-ON-ONE-HAND-in-Chinese/.

[8] Memory achievements (webpage). http://www.worldmemorychampionships.com/MemoryAchievements.asp.

[9] Phillip Isola, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. What makes an image memorable? In *24rd IEEE Conference on Computer Vision and Pattern Recognition*, pages 145–152, 2011.

[10] What makes an image memorable? (popular summary). http://web.mit.edu/newsoffice/2011/memorable-images-0524.html. MIT news office, 2011.

[11] Adrian Perrig and Dawn Song. Hash visualization: a new technique to improve real-world security. In *In International Workshop on Cryptographic Techniques and E-Commerce*, pages 131–138, 1999.