Sign in

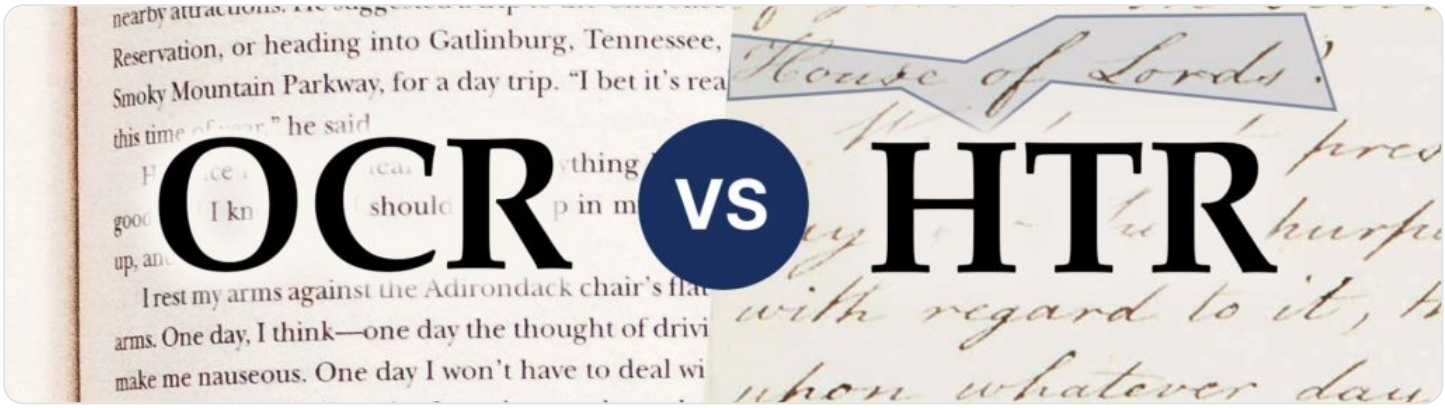
(https://readcoop.eu)

ARTIFICIAL INTELLIGENCE

OCR vs. HTR or “What is AI, actually?”

2 years ago

Felix Dietrich



As of today, the human brain is arguably the most complex computational structure in the known universe. With up to 10^{11} neurons, it contains more electrically excitable cells than a blue whale’s brain. And with up to 10^{15} synaptic connections, it still outclasses the largest machine learning models in terms of learnable parameters. But computers are catching up at an increasing rate. Back in 1997, people were shocked to see the world chess champion, Garry Kasparov, defeated by IBM’s Deep Blue supercomputer. But while many used to believe that beating humans at chess would require some form of “true” intelligence, people were quick to call out Deep Blue for what it really was: An efficient way to search tree-like data structures. It turns out that all it took to defeat humans, was the ability to look at all the possible moves of both players and then pick the best one. While this tree grows exponentially with every move, as soon as a computer can compute 12 or 14 moves ahead in a reasonable timeframe, humans can no longer keep up.

Historic Matches: Humans vs. AI

A game that was widely considered to be truly beyond any ordinary computer’s capabilities was the board game Go. While the rules may seem much simpler than those of chess, Go actually has a game tree complexity that is more than two hundred orders of magnitude larger. Consequently, even much more powerful computers than Deep Blue had no chance at solving this task. Yet, 19 years after Kasparov’s fateful match, people were once again taken aback when Google’s AlphaGo computer defeated the reigning Go champion, Lee Sedol, in 2016. This incredible success came in the wake of a true revolution in the way we see and understand artificial intelligence. A way that much more closely resembles inner workings of the human brain. This technology enabled computers to catch up with humans in many hitherto unimaginable areas – one of them being image based text recognition. The technological step between *Optical Character Recognition* (OCR) and *Handwritten Text Recognition* (HTR) turns out to be equally profound as the one between Deep Blue and AlphaGo.

OCR (Optical Character Recognition)

While it makes virtually no difference to the human eye, machine-written text like the one on the right is completely illegible to a computer. The reason is that it is not made up of characters, but of pixels in an image. On the other hand, transforming characters to pixels is a

trivial task. All it needs is a mapping of every character to a collection of pixels that represent it in a specified font. This is fine for all applications where a human simply wants to read the text, such as displaying it on a screen or printing it on a sheet of paper. But there are also applications where one would like to convert such an image back into actual, digitized text. This doesn’t just make it easier to store it, but also enables one to quickly edit sections or even search for words. The problem is that transforming pixels to characters is no longer an exact, straightforward process. There are a myriad of image compression techniques that can produce all sorts of unfavourable effects at the pixel-level. In addition to that, people would also like to digitize scanned or even photographed documents, which could introduce displacements or dirt and smudges. In the end, the best we can do is an approximation.

Fortunately, these approximations do not require a lot of computational power or even sophisticated algorithms. Pattern matching using matrices like the one on the left has resulted in extremely good character recognition as early as the 1970s. While this trivial pattern matching requires a certain statistical leeway when interpreting characters, there are also more mathematically rigorous algorithms. For example, one could try to take all the lines and closed loops and turn them into a graph. The problem can then be reformulated as identifying isomorphic subgraphs. While modern OCR algorithms still haven’t reached perfect accuracy, they are simple, fast and accurate enough for most fonts to be useful in a wide range of devices – from real-time handheld laser scanners to smartphones through to to copying machines.

HTR (Handwritten Text Recognition)

While the problem of OCR for machine text has effectively been solved, in most applications, for a long time human handwriting displays a near-endless range of fonts and styles. Correctly recognizing these is way beyond classical pattern matching algorithms, which is why handwritten text recognition remains at the cutting edge of science. Nowadays, every few

months a research group or company shows off a new, improved algorithm. But this wasn’t always the case. In the early 21st century, this task was still considered to be practically impossible. The best research groups in the world could not come up with something remotely useful. So what changed? The short answer is: *artificial neural networks* (ANNs). The same technology that has allowed computers to defeat the world champion at Go has now enabled us to tackle human handwriting. This approach fundamentally differs from classical algorithms in the sense that the recognition model is no longer programmed by hand, but automatically learned from a set of examples. The architecture behind ANNs has existed for a long time, but only improvements in parallelized computing, network structures and training algorithms – and, particularly, availability of training data – have allowed them to actually become useful outside of academic exercises.

One of today’s most commonly used network types for image recognition is the so-called *convolutional neural network* or convnet.

Exemplary Structure of a Convolutional Network

The finer details of these networks vary significantly, but the fundamental structure is always the same. The process starts by taking the pixels of an image as input and then goes on to extract features by sequentially applying certain filters. These filters are essentially masks that are swept over the image to see if something fits them. In classical pattern matching, humans would have to pre-determine the way these filters look, but in a convnet they start out randomly and then get refined during the training process. The final set of features is then fed into a densely connected network, which is where the true universal prediction power of this algorithm stems from. There even exists a mathematical proof (which is pretty rare in the field

of ANNs) stating that such a network can learn to approximate any reasonably well behaved function to arbitrary precision, as long as the network is sufficiently big. Unfortunately, though, the proof says nothing about how big it actually has to be, so this usually has to be determined via trial and error. Back in the convnet, this layer is then connected to a sequence of outputs, which could really be anything. For HTR, one would ideally want to have characters or even words in this final layer. During training, a set of reference images with known contents are fed into the network and then its output is compared to the real values. Based on the difference between model prediction and ground truth, the parameters inside the network get updated iteratively. When the training is complete, new images can be recognized by looking at the output that shows the strongest activation. With the right network structure and training setup, one can even get something like a distribution of probabilities in the output layer.

Learning Curve of the Public AI Model German Kurrent M2 (https://readcoop.eu/model/german-kurrent-and-sutterlin-17th-20th-century/)

Of course, this is not exactly as trivial as it might sound. For widely applicable models one usually needs huge amounts of training data and computing resources and even then there are many pitfalls. Software developers and hardware manufacturers have worked in tandem for the

last few years, providing powerful chips that are purpose-made for typical AI calculations, and highly versatile software frameworks to make use of them. At READ-COOP, we are constantly expanding our hardware capabilities to make use of the latest technological developments and we always keep a close eye on newly emerging algorithms to enhance our platform.

Share this article:

Products & Services

Transkribus (https://readcoop.eu/transkribus/)

Transkribus lite (https://transkribus.eu/lite)

Read&search (https://readcoop.eu/readsearch/)

ScanTent (https://readcoop.eu/scantent/)

Useful information

News (https://readcoop.eu/news/)

Download Transkribus (https://readcoop.eu/transkribus/download/)

Public Models (https://readcoop.eu/transkribus/public-models/)

Payment and shipping (https://readcoop.eu/payment-and-shipping/)

Helpful resources

Resource center (https://readcoop.eu/transkribus/resources/)

How-to Guides (https://readcoop.eu/transkribus/resources/how-to-guides/)

Getting started with Transkribus (https://readcoop.eu/transkribus/start/)

FAQs (https://readcoop.eu/transkribus/questions/)

Videos (https://readcoop.eu/transkribus/resources/videos/)

(https://twitter.com/Transkribus) (https://www.linkedin.com/company/read-coop-sce) Community (https://www.youtube.com/channel/UC-txVgM31rDTGIBnH-zpPiA) (https://www.facebook.com/transkribus)

Terms & conditions (https://readcoop.eu/terms-and-conditions/)