A Guide to Touchless Mortgage Automation

SoftWorks

Optimizing Performance for Mortgage Automation

Among recent changes in the mortgage industry, perhaps the most significant has been the transition from paper to PDF documents. The introduction of electronic documents to mortgage processing has enabled increased use of automation technology across the financial services. More directly, it has led to the emergence of “Digital Lending”. Digital lending is the area of FinTech involving the disintermediation of lending, connecting the lender and borrower directly based on their digital information and automating much of the background work that goes into loan processing.

The critical first step in any digital lending workflow, before machines can connect anyone or automate anything, is to accurately understand the user’s digital information. This means that machines must be able to read and understand a “loan packet”, a several-hundred-page file into which all sorts of documents, from paystubs to home appraisals are dumped. To collect all that data with the same accuracy as a human requires a lot of processing. When it comes to automated processes of that size and complexity, changing just a few aspects of the process workflow can increase or decrease the processing speed fivefold.

This article will discuss the technologies that most impact machine performance, along with strategies and techniques to maximize the value and efficiency of automated systems in mortgage processing and in general.

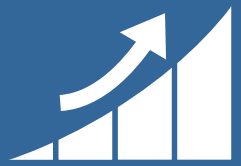
Topics:

4 Primary Factors Moving the Mortgage Industry towards Automated Processing

Cost Reduction

Real-Time Opportunity

Processing Equilibrium



Scalability

Mortgage Automation Capabilities

Different mortgage solutions automate unique sets of tasks, pertaining to different areas of the industry, such as underwriting and pre-underwriting, mortgage insurance, and compliance. Before exploring the specific performance factors for mortgage automation systems, it is important to briefly explain exactly what these systems are about and, in a general sense, how they work.

A complete mortgage automation solution, as discussed in this paper, performs the following steps:

1. **Text Recognition – Scanned PDFs are put through an OCR engine to identify text and, in some systems, other types of structured data, such as tables, lists, and images.**
2. **Document Classification – Using the identified text, the system sorts the pages by type, separating out and labeling individual forms and documents.**
3. **Document construction – Once it knows which document each page belongs to, the machine splits them apart and can reorder, or "stack" them to make the loan packet consistent and easier to process.**
4. **Data Extraction – The machine looks through the loan packet to find and extract specific data fields from within specific documents.**
5. **Data Validation – Once extracted, data cannot be used until it is verified. In an automated system, this means evaluating the accuracy of each machine-performed task relative to a pre-set accuracy threshold.**
6. **Underwriting Decisions – Once all necessary data has been extracted and structured, the system can apply a set of rules or algorithms to make complex decisions and conclusions based on that data.**

Automated document classification and data extraction can greatly reduce labor costs by minimizing the need for manual processing.

Text recognition (OCR) technology has existed for years and has been widely adopted across the financial services industry. Even automated classification, construction, and extraction have been widely, if not entirely, adopted in the mortgage industry. However, industry dependence on these technologies has been heavily checked by the lack of widespread effective data validation methods. Many lenders allow automated systems or "bots" to do some of the data entry grunt work but generally cannot confirm the results of that work.

The more technology-forward FinTech sector is shifting that dynamic, and today's most advanced mortgage solutions can automatically evaluate the accuracy of the data they extract to eliminate the need for human review in many cases. This auto-validation process paves the road for fully automated underwriting decisions, further reducing human "touchpoints" and saving time and money in mortgage processing.

Uptake of this technology by large banks and lenders has been slow, but the potential ROI is huge and growing. American financial institutions are still spending more than $25 billion annually[1] on mortgage underwriting processes that could be done more accurately and in real-time by a machine.

Identifying Document Types

There are two general types of PDF documents: (1) scanned and (2) electronic. Scanned PDF documents typically originate on paper and are captured or scanned to create a digital image. Electronic documents, on the other hand, are created by computers and, therefore, typically contain much more embedded information. Though these different document types may look the same, the text and other information in electronic documents can be easily understood and processed by a machine, while scanned documents are essentially a 2D grid of pixels (typically about 9 million of them) that must go through several complex computational processes in order to convert them into electronic format. That conversion process is computationally expensive, non-trivial, and error prone.

1 https://www.huduser.gov/Publications/pdf/HUD-11648.pdf

Electronic, or "digitally born" documents are easier to process for several reasons:

1. **Don't require OCR**
2. **No chance of a recognition error (such as mistaking the letter "O" or "I" for the number "0" or "1")**
3. **No need to determine the document's "read order"**
4. **Stored as distinct characters, instead of millions of individual pixels**
5. **Process hundreds of times faster than scanned documents**

In practice, approximately 30%-40% loan packets are hybrids[2] , containing a mix of both scanned and electronic pages. Consider a loan packet that might normally consist of about 35 or 40 different forms or files. A "truth in lending" statement might have been digitally generated by the lender (e.g., eLoans), while the borrower tax documents (e.g., 1040) might have been scanned in by the borrower as part of his/her application. All these documents are often collected into a single composite PDF, but each page may be constructed very differently.

There are three general strategies for handling hybrid documents:

1. **Assume the document is entirely electronic, extracting embedded text but not recognizing or extracting any scanned or otherwise non-embedded data.**
2. **Treat the whole document as scanned, passing every page through image capture, OCR, classification, and extraction engines, and not using any embedded data.**
3. **Parse the loan packet and separate the electronic and scanned pages, extracting all embedded text and only applying image recognition to scanned pages.**

Assuming every page is electronic will not work well on mortgage documents, which are primarily scanned. To fully automate mortgage processing with a knowledge bot requires understanding every page. Failing to extract non-embedded text is not an option. On the other hand, OCRing every single page slows the process down up to 100-fold, introduces sources for error, and makes it virtually impossible to develop a working real-time solution. The third option, separating scanned and electronic pages, is by far the more intelligent, efficient and effective strategy. However, understanding a PDF document requires building a parser for the PDF language, which is a complex language. That parser must analyze each page of the PDF document and determine whether it has any image dependencies requiring OCR processing. This technical obstacle poses a barrier-to-entry or, at the very least, a barrier-to-optimization for many organizations looking to automate their mortgage processing.

A typical loan packet may be a hybrid of approximately 30% electronic and 70% scanned pages. If parsed and handled efficiently, this hybrid document would process about 30% faster than an entirely scanned document of the same length, but without neglecting any non-embedded data. Very importantly, often 80% of the required data fields can be found in the electronic pages. This means that 80% of the critical data can be collected with zero recognition errors and can even be "auto-validated" (automatically certified as correct), reducing manual data-validation time by 80%. The effects and importance of "Auto Validation" are explained later in the eponymous section.

Processing equilibrium ensures turn around time and processing accuracy are approximately consistent across different input sources and document types.

Reducing Document Dimensionality

The dimensionality of a document corresponds to the number of independent variables, or "dimensions", used to define the contents of that document. A scanned page in black and white has rows and columns, giving it a 2D structure. It is essentially a grid of 9-10 million pixels, each of which is a single bit that is either "ON" (black) or "OFF" (white). Grayscale documents add another dimension. In a grayscale document, each pixel in the grid contains eight bits, representing a scale of 256 unique shades of gray. Full-color scans are an even higher level of dimensionality.
Each colored pixel contains twenty-four bits, 8 bits each for the respective

2 https://www.fanniemae.com/content/news/hybrid-arm-components-commentary.pdf

intensities of red, blue, and green. Based on these extra-dimensional descriptors, each pixel is assigned a specific color, from a range of 16,777,216 unique possibilities.

Effectively reducing the dimensionality of certain pages while preserving important data is a critical, if challenging, step for any efficient mortgage automation system. This is because each document dimension requires analysis. The higher the dimensionality of the document, the more computation and analysis is required. Also, the higher the dimensionality, the slower the processing rate and the greater the likelihood of errors in converting the data to an electronic data stream. In that sense, introducing shades of gray to a document quite literally blurs the lines that differentiate one character from another, confusing the OCR engine and slowing it down dramatically.

An OCR engine typically expects a binary or bitonal image[3] (black and white). Where an image is captured in color or grayscale, a segmentation module needs to parse the image into components and correctly separate out text, graphics and picture regions. This segmentation phase is slow and very prone to errors. Documents that are already segmented out into these components will parse better and faster, with more reliable OCR. When an image is already "binarized", then text, graphics, and picture regions can be appropriately rendered using distinct data representations.

Binarization, in this context, means converting an image from color or grayscale to black and white. Most business documents are already black and white but stored in a full-color format. Each of these pages must be tested to confirm that it's black and white. Some machines can even detect and "cut around" color pictures that were embedded on top of an otherwise black and white document. They do this by testing locally at several different points on each page. Once a document has been identified as black and white it can be binarized, or converted to a bitonal format.

Parallel Processing

The primary hardware-driven method of speeding up any automated workflow is by "parallel processing". Parallel processing is when different algorithms are executed at the same time – or in parallel. Alternatively, it's when the same algorithm is executed in parallel on different input data, and by different processors.

3 https://www.digitisation.eu/download/website-files/BPG/OpticalCharacterRecognition-IBPG_01.pdf

For mortgage processing, this means that multiple loan packets or documents can be processed simultaneously on different machines or, more commonly, different "brains" within the same machine.

The thinking component of a computer is known as the CPU (Central Processing Unit). For many years, probably about the last 15, the CPU clock speed has been maxed out at about 3-4 billion clock cycles[4] , or computer instructions, per second. This limitation is connected to actual physical limitations in chip design and will probably not speed up much in the next decade. As such, CPU speedup has been achieved effectively by developing multi-core chips, where a chipset might consist of 4, 8 or 16 cores, which work simultaneously.

There is a generally linear relationship between the number of cores and the speed of the system. This means that doubling the number of cores, for instance, would double the processing speed. This can be achieved by doing either or both of the following: upgrading to larger servers with more core in each and/or increasing the total number of servers. A server represents a machine or computer that includes CPU, memory, read/write control, and non-perishable memory (disk drives). Each core is effectively its own "central processing unit" (CPU) that can process the general instruction set. The multiple cores can each function independently, running their own distinct computations while sharing the server's other resources, such as memory.

A multi-core environment can make document processing more efficient in at least one of two ways:

1. **Binding each document to a physical or logical core**
2. **Binding each page to a physical or logical core**

In the first method, each whole document is processed by a single core, with multiple cores processing different documents simultaneously. The second method splits up each document and sends each page of the document to be processed in parallel by multiple cores.

These methods are very different in their approach, but typically, both methods yield an approximate n-fold speedup for an n-core server. There is some additional efficiency achieved using the first method, where each document maps to a CPU core. This is because there is some additional overhead in splitting and later reconstructing the same document.

4 https://www.bhphotovideo.com/explora/computers/tips-and-solutions/boost-processors

Performance Metrics Overview

The above processing techniques each have a quantifiable effect on the processing speed of an automation system. This section will demonstrate the amplified effects of combining these techniques, and the degrees of optimization and computational power required to reach different loan processing rates.

As explained above, for a given server configuration, a fully electronic loan packet will process much faster than a scanned loan, and a bitonal document will process much faster than a full-color document. Similarly, for a given loan document, a server configuration with 64 cores will process the loan much faster than a single 4 core machine.

The table below provides basic runtime performance figures, depending on both document type and machine configuration. These runtimes assume amortized processing rates, i.e., overall loan throughput, rather than time to process a single loan:

Document Type	Machine Hardware			
	1–core	4–core	16–core	64–core
Electronic	1 pg./sec	4 pgs./sec	16 pgs./sec	64 pgs./sec
Scanned	0.147 pgs./sec	0.588 pgs./sec	2.352 pgs./sec	9.4 pgs./sec
Bitonal	0.16 pgs./sec	0.64 pgs./sec	2.56 pgs./sec	10.24 pgs./sec
Color	0.11 pgs./sec	0.44 pgs./sec	1.76 pgs./sec	7.04 pgs./sec
Real Loans	0.18 pgs./sec	0.72 pgs./sec	2.88 pgs./sec	11.52 pgs./sec

Let's take, for example, a stack of 100 loan packets of about 400 pages each being processed on a 4-core machine. The "Real Loans" above are assumed to be made up of 80% scanned and 20% electronic pages. Within that, the scanned documents themselves are assumed to be 80% bitonal and 20% color. The given processing rate for real loans on a 4-core machine is 0.72 pgs./sec. Using this number, the total time required for OCR, classification, doc splitting, and data extraction for our 100 loan packets is 40,000 / 0.72 = 55,556 seconds, or almost 15.5 hours. On average, this system would be spending about 9 minutes on each loan packet. Of course, as explained above, the system's speed scales linearly with the number of processors in the configuration. Thus, on a 64-core machine, the processing time would be just 35 seconds per loan packet. At this rate, the entire stack of 100 loan packets could be processed in under an hour.

This example can be extended to a larger, 192-core configuration consisting of 12 individual 16-core servers. This machine configuration can process the same 100 loan packets in just 20 minutes. Assuming a 12-hour processing day, this system could process over 3,600 loans per day, and nearly 1,000,000 loans per 250-day work year.

Scaling

While some of the above metrics should be relatively constant for a given automated system, the amount of available computational power can be varied, or "scaled", with fluctuating work volumes. This means that when business is busier, more servers and more processors can be added. And, as explained above, adding processing power to a machine linearly increases its processing speed, allowing more documents, or in this case loan applications, to be processed in the same amount of time or less.

Scaling is critical in many areas of financial services. The mortgage industry in particular sees significant volume fluctuation[5] in response to seasonal shifts, changing interest rates, and other dynamic market conditions. Knowledge work, however, is very difficult to scale. Knowledge work is any profession which requires a considerable amount of training. This training poses a barrier to hiring additional staff for the busy seasons. Each employee is an investment that, when volumes subside, will no longer be fully needed and might be left idle.

There are approximately 350,000 loan officers in the US, each supported by 2 other workers[6] , on average. That's about 1.05 million employees involved in US home mortgages. The amount of training between them will vary, but it's certainly knowledge work. Some aspects of loan processing are easier, like pre-underwriting, which is converting data found in documents to actionable data in a mortgage origination system. Other aspects are more complex and require greater expertise, like underwriting a jumbo mortgage for a self-employed borrower. Lenders cannot hire and train workers like this quickly enough to keep up with a potential demand spike. And even if they did, those workers wouldn't be needed anymore once demand decreased and that training would have been wasted.

5 https://www.brookings.edu/wp-content/uploads/2018/03/5_kimetal.pdf
6 https://www.bls.gov/ooh/business-and-financial/loan-officers.htm

Typically, mortgage companies have very non-uniform workloads. Monday loan applications are higher than later in the week as they accrue over the weekend. People often purchase homes over the summer so that they're able to move before the new school year. When the Fed lowers rates, people are more likely to refinance an existing mortgage.[7]

So, lenders, insurers, and other sectors of the mortgage industry have difficulty staffing an industry where the workload is non-uniform, but the knowledge work requires expertise. This is one area in which machines offer tremendous value. A well-designed mortgage automation system can handle many of the aspects of pre-underwriting and underwriting and be scaled quickly and easily to accommodate changing volumes and eliminate bottlenecks in mortgage processing.

Scaling a mortgage automation system requires only some simple math and possibly some extra hardware. The most easily scalable type of system is one which is run on the cloud, i.e., in an external datacenter. These cloud-processing providers have the capacity to add servers to a project as needed. So, as volumes spike or drop, they can instantly add or remove servers from the system to speed it up or slow it down, keeping the turnaround time for loan applications relatively constant regardless of how many there are. If all the processing is being done "in-house", i.e., on the organization's own servers, the scaling process is almost identical. The only difference is that the organization must add their own servers, which may require taking them off other tasks, or just having a few extras on hand. In organizations running multiple automated systems for different aspects of their business, server distribution may be relatively fluid, allowing resources to be redistributed on the fly according to the dynamic needs of each software system.

There are usually two types of process automation: semi-automated and fully-automated. The semi-automated process requires some degree of human supervision or involvement while the fully automated process does not. Once human involvement is required, however, that involvement limits the degree to which a system can be scaled. If, for example, a human underwriter is still required to review each document to confirm the data, then that person becomes the limiting factor for the entire system. The whole process can only be made to go as fast as the slowest manual step.

7 http://business.time.com/2012/10/10/the-best-time-to-buy-or-sell-a-house/

Auto-Validation

Auto-validation is the capability that most singularly transforms a mortgage solution from semi-automated to fully-automated or, at the very least, minimizes human touchpoints. Simply stated, auto-validation is when a system knows that it knows. A machine claims to perform the same task as a human. However, like human workers, machines are imperfectly accurate, occasionally misreading data or incorrectly classifying a document. To catch these mistakes, many automated processes are double-checked by humans. (Even manual processes are often double-checked by humans.)

The process of checking the correctness of each document and data field is faster than doing all the work from scratch but is still tedious and time-intensive. If a system can "auto-validate" some of the tasks, double-checking its own work with near-100% certainty, then it can eliminate some, even most, of that manual processing.

The challenge with even the most accurate machines is that the error could be anywhere. Whether a machine is right 98% of the time or just 75%, the only way to catch its mistakes is by going over every single automated decision, because those mistakes are equally likely to be in any individual action or data field. Auto-validation adds another layer of evaluation, using several different techniques to generate a specific confidence rating for each task that the machine does, from splitting a document to extracting a particular field. Being able to evaluate its own accuracy means that the system can automatically confirm high-confidence results, and flag only the low-confidence results for review. With auto-validation, human workers only check the data and decisions that are the most likely to contain errors.

In the abstract, auto-validation works by creating a probability space around each automated task. A probability space is the set of possible values for that particular data point or action, and the likelihood that each one is correct.[8] For example, one character may have a 74% chance of being a "7", a 22% chance of being a "1" and a 4% chance of being an "i". The probability space for a whole field (word, number, phrase, etc.) is a combination of the probability spaces of its constituent characters, coupled with contextual elements, like spellcheck. These elements can work in reverse too. So, a character may look like either an "i" or a "7", but if it's in the middle of the word "impossible", then the machine can be confident that it's an "i". A word with a number in the middle of it signals to the system that there is likely something wrong.

8 https://www.automl.org/wp-content/uploads/2018/12/automl_book.pdf

Creating a probability space around each event is not trivial. In fact, it is probably the most difficult aspect in developing a touchless automation system.[9] For example, data extracted from electronic documents can often be given a confidence rating at or near 100%, since the data was embedded and never underwent OCR. (This form of auto-validation, of course, requires that a system be able to recognize and appropriately handle electronically generated PDF documents, instead of treating everything as scanned.) Scanned data fields, on the other hand, vary widely. Each character of each data field has its own probability space. If the same data point appears in multiple places throughout the loan packet, the module may have to combine several different probability spaces, each of which is itself a combination of several characters, each with their own probabilistic complexities. Even the accuracy of each initial document split and page classification factors into the final confidence of the output data.

A thorough auto-validation module combines and distills all these probabilities to a unique accuracy level for each automated task and individual data point. These values provide the flexibility to set customizable and field-specific accuracy thresholds. The same system, for example, may require a 99% accuracy level for critical fields, 95% for semi-critical fields, and only 80% for non-critical fields.

A more nuanced system could even assign each field its own unique confidence threshold. In effect, loan officers can spend the bulk of their time reviewing only the most important data, with the highest likelihood of error. The rest of the data, all of which has been auto-validated, can pass through the system without ever needing to be reviewed by a human.

Auto-validation is what enables touchless automation, and what allows automated systems to so dramatically speed up mortgage processing. The percentage of tasks that can be auto-validated correlates directly with the percentage of touchless automation the system will exhibit. This, in turn, correlates directly with the ROI of the system.[10]

Even a very high recognition rate of 97% means that nearly 1 in 30 characters are incorrect. Given an average of 5 characters per word and 500 words per page, that amounts to 75 mistakes per page. And while many of these mistakes are easily caught by a simple spellcheck, others (such as numbers) could slip by under the radar and pose serious analytical problems down the line. Auto-validation defeats this propagation of error and allows high recognition rates to translate into real, highly accurate data and decisions.

9 http://www.its.caltech.edu/~mshum/stats/lect1.pdf
10 https://www.worksoft.com/files/resources/critical_considerations_for_adopting_automation_eb-ook.pdf

System Feedback, Analytics, & Optimization

Intelligent automation solutions continue to learn and improve their performance over time. A system, in order to improve, needs feedback on how it's doing. Based on this feedback it can analyze and optimize its performance via machine learning (ML) and/or manual adjustments.[11]

The first step towards generating system feedback is maintaining a database of all relevant performance information. Maintaining such a database allows both the machine and the user to run new queries, either in real-time or after the fact. Queries that the database would be able to address include, for instance:

- **Which forms have the lowest recognition rates?**
- **Which data fields have the highest false-positive or false-negative rates?**
- **How many times, on average, must a human interact with each loan packet (i.e., the number of human touchpoints)?**

The feedback/analytics/optimization cycle is critical for maximizing a system's performance and ROI. For example, let's say that a mortgage automation system is processing loan packets at a rate of just one loan every half hour. Your "back-of-the-envelope" calculations indicate that it should be processing a loan every 3 minutes, so your machine is running about 10x slower than expected. System feedback and analysis may tell you that 34% of the pages flagged for human validation came from a single type of form, creating a bottleneck for the human reviewers. Identifying that problem allows it to be addressed, perhaps by retraining the system to correctly recognize the problematic form.

Similarly, the system may report that a particular data field is consistently flagged for review, despite being correct 99.9% of the time. Based on this feedback, the system's auto-validation measures for that field can be adjusted, either manually or automatically. This will allow the system to auto-validate that field more frequently, further reducing human touchpoints and improving the processing speed.

11 https://www.gartner.com/binaries/content/assets/events/keywords/catalyst/catus8/preparing_and_architecting_for_machine_learning.pdf

Intelligent Mortgage Automation

There are many factors that influence system performance, and each one propagates differently as systems expand and change. The most valuable features in any mortgage automation solution are the ones that intelligently grow with the system and continue to add value as the volumes are increased and new tasks are automated.

Continuous, automatic, iterative improvement is characteristic of intelligent systems, or ones that incorporate machine learning and AI. Effective implementation of these technologies is what enables features like auto-validation and system analytics. But AI and ML algorithms can even be used to optimize simpler processes [12], such as efficiently distributing work across an array of servers and processors, or more accurately converting color images to bitonal.

> **"A simple degree of automation can cover the basic, purely repetition-based aspects of mortgage processing. But it takes an intelligent solution to automate the more complex, knowledge-based parts of the process, to begin making automated lending decisions in real-time, and to achieve the constant analysis and optimization that makes automation so effective and valuable in financial services and in mortgage automation in particular."**
>
> **-Dr. Ari Gross - CEO SoftWorks AI**

A simple degree of automation can cover the basic, purely repetition-based aspects of mortgage processing. But it takes an intelligent solution to automate the more complex, knowledge-based parts of the process, to begin making automated lending decisions in real-time, and to achieve the constant analysis and optimization that makes automation so effective and valuable in financial services and in mortgage automation in particular.

12 https://medium.com/datadriveninvestor/differences-between-ai-and-machine-learning-and-why-it-matters-1255b182fc6

About SoftWorks AI

SoftWorks AI is dedicated to helping businesses enhance operational efficiency by providing state-of-the-art computer vision and automation solutions. We strive to leverage our deep expertise in computer perception, OCR and AI technologies to convert raw information into actionable insight, equipping knowledge workers with the means to drive business value faster and more intelligently.

Contact us at:

SoftWorks Phone: +1 (888) 575-9299

Email: info@softworksai.com

www.softworksai.com