Hewlett Packard Enterprise

Business white paper

Augmented intelligence

Helping humans make smarter decisions

Table of contents

The quest to make computers "intelligent" is as old as computers themselves. The phrase **artificial intelligence** produces notions of a robot-controlled future in which humans have been rendered largely obsolete. But HPE IDOL next-generation enterprise search and data analytics platform uses pioneering techniques in artificial intelligence to automate and enhance the processing of human information—not to take the decision **away** from humans, but to help us make the best one. We call this approach augmented intelligence. This paper outlines the most important of these information analytics techniques.

Artificial intelligence as an aid to decision making

The realization that computers could perform calculations that would take humans many weeks, or we could not achieve at all, rapidly engendered the idea that we could make these mere machines do all we can do. It is no coincidence that the golden era of science fiction began around the time of the great burst of computing and space technology that followed the Second World War, and that in much of it the futuristic visions include nonhuman technologies whose capabilities far outstretch those of their mortal creators.

Artificial intelligence is already all around us. The sensors that help determine the optimal time to change traffic lights, washing machines that automatically adapt to the quantity of clothing, and the ever-changing gameplay of our favorite smartphone games are examples. Even the system that prevents a microwave from starting because the door is open is artificial intelligence in action. Computers are good at making decisions when armed with all the relevant information and with no element of randomness—systems known as "fully observable deterministic systems." For this reason, computers are far better than humans at playing checkers, but fall down in the very human game of poker.

But while we strive for a world in which a machine will keep our garden perfectly manicured or tell us exactly when to book our vacation to guarantee perfect weather, most observers of the field of artificial intelligence fail to note one crucial aspect: When it comes to the most human parts of our existence—our interactions, decisions, and interests—machines have far less to offer. We don't actually want the machine to pick which flowers to plant; we just want it to execute our wishes quickly and efficiently. Less still do we want a computer to tell us **where** we will be taking our vacation, even if we are clamoring for it to help us make a smart and informed choice.

From its inception, HPE IDOL has created a number of pioneering data analytics techniques in artificial intelligence that help to automate and enhance the processing of human information of all types. Our aim is not to take the decision away from humans, but to equip them with the information to determine the best course of action. We call this approach to artificial intelligence augmented intelligence.

Within augmented intelligence, IDOL uses a wide variety of theories and techniques to process and extract meaning from human information. We shall outline the most important of these theories to explain how they can be used to power enterprise solutions on some of the world's largest private indices.

Bayesian theory

The theoretical underpinnings of IDOL's approach to human information can be traced back to Thomas Bayes, an 18th century English cleric whose works on mathematical probability were not published until after his death ("Philosophical Transactions of the Royal Society of London," 1763). Bayes' work centered on calculating the probabilistic relationship between multiple variables and determining the extent to which one variable impacts another. At the core of Bayesian theory lies Bayes' Theorem, a deceptively simple equation relating the conditional probabilities of multiple random variables.

$$P(A|B) = \frac{P(A)\,P(B|A)}{P(B)}$$

The power of Bayes' Theorem rests in its application to handling complex data sets. It provides a mathematical framework for describing how a system's model should be updated based on the observations that have been made of that system. This interpretation alone removes the need for complex a priori models that claim to describe how a system should work and instead permits the automatic creation of a custom-built model that actually fits the data in question.

Take, for example, a corpus of text documents used to create a document retrieval system. Complex linguistic models that claim to understand the English language will work well on much of the data, but they will fail to understand industry-specific jargon, neologisms, or customer-specific products and usage. Contrast this with a Bayesian system. Only a very basic language model is initially used, but it is then updated using Bayesian theory to arrive at a language model that is both specific to that industry and tailored to that particular body of documents. Not only is the approach independent of language, but it will automatically update itself as new data containing new words or existing words used in new ways is added to ensure that the model always remains fresh.

Similarly, in the field of security and surveillance, Bayesian theory provides a versatile method of interpreting the wide variety of activity captured by a set of video cameras. The proliferation of security cameras has resulted in humans becoming unable to watch all the cameras for signs of trouble. Surveillance software containing predefined models to spot unusual behavior is, in fact, only capable of spotting activity it has been explicitly trained to look for. IDOL's modules, on the other hand, are unencumbered by predefined models and are able to use video streams to determine normal behavior and thus rapidly spot behavior that falls outside of that model and trigger an alert to the human security personnel.

One of the reasons Bayesian theory works so well on the processing of human information is that humans are Bayesian creatures. We are born into the world without preconceived models of how the world should work, and every observation we make updates our model of the world as to what is normal and what is notable. As a result, it is easy for a human to watch a video stream to determine what normal behavior is for a particular area, and when something unusual occurs it is immediately evident.

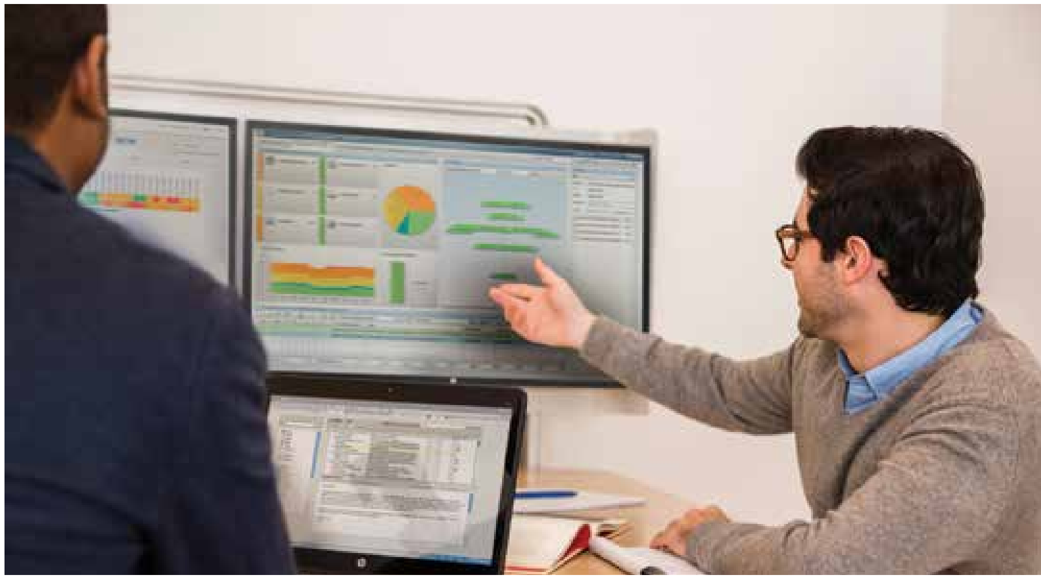
A traditional statistical argument is that if a coin is tossed 100 times and comes up heads every time, it still has an even chance of coming up tails on the next throw. An alternative Bayesian approach is to say that 100 consecutive heads are evidence that the coin is not fair or perhaps has heads on both sides. Once again, the human approach is intrinsically Bayesian. It would be a brave person who bet on a tail coming up after watching 100 consecutive heads.

A typical problem in information retrieval is to judge how relevant a document is to a given query or agent profile. Bayesian theory aids in this calculation by relating this judgement to details we already know such as the model of an agent. More formally, the resulting a posteriori distribution, which is applicable in judging relevance, can be given as a function of the known, a priori models, and likelihood.

We can use the same data analytics techniques to provide adaptive models of a user's behavior. For example, we can use the documents that a user writes, views, or marks as relevant in judging the relevance of future documents. IDOL's adaptive probabilistic concept modelling (APCM) allows this information to be "back propagated." Agents can be improved by retraining.

IDOL's use of Bayesian theory goes farther than simply judging the relevance of a document to a query. APCM analyzes the correlation between features found in documents relevant to an agent profile to find new concepts and documents. And it identifies concepts important to sets of documents, allowing new documents to be accurately classified.

Although no one knows for certain what Bayes' original goal was, Bayes' Theorem has become a central tenet of modern statistical probability modeling. By applying contemporary computational power to the concepts pioneered by Bayes, we can now calculate the relationships between sets of variables quickly and efficiently, allowing software to manipulate concepts.



Information theory

Information theory is the mathematical foundation for all digital communications systems. Claude Shannon's innovation, as described in his "Mathematical Theory of Communication" (1949), was to discover that information could be treated as a quantifiable value in communications. This proves incredibly powerful in the processing of complex data streams in allowing features of interest to be determined automatically, and to extract the highest value pieces of information.

Consider the basic case where the units of communication (for example, words or phrases) are independent of each other. If p(x) is the probability of the x^{th} unit of communication, then the average quantity of information conveyed by a unit, known as Shannon's entropy, is given by:

$$H = -\sum p(x) \log p(x)$$

This formula reaches its maximum when the probabilities are all equal. In this case the resulting text would be random. If this is not the case, the information conveyed by the text will be less than this maximum so there is some redundancy. This result is then extended by more sophisticated mathematical arguments to describe more complex interrelations among variables.

Natural languages contain a high degree of redundancy. You can understand a conversation in a noisy room even when you can't hear some of the words. You can understand the essence of a news article by skimming over the text. Information theory provides a framework for extracting the concepts from the redundancy.

IDOL's approach to concept modeling begins from the tenet of Shannon's theory, which says the less frequently a unit of communication occurs, the more information it conveys. As a result, concepts and ideas that are unusual or distinctive within the context of a communication tend to be more indicative of its meaning. IDOL applies this theory to determine the most important (or informative) concepts within a document.

Structured analytics

Graph analytics

Graph databases provide a new way to model the world around us. They consist of nodes and edges, where a node is an entity—such as a person, place, or even a concept—and an edge is a connection or relationship between two nodes. For example, a graph of a social network might represent people as nodes and communications between them as edges.

Graphs offer a new approach to analyzing data. By putting the relationships between entities at the forefront, we can answer questions like:

- Is our network densely or sparsely connected?
- Who is the most connected person in the network?
- Which friends do two people have in common?

Graphs help us spot complex patterns in data we might otherwise miss. For example, we can use graphs to build recommendation systems that link customers and products based on similarities in customers' purchasing history or to identify friendship groups by spotting cliques of users in a social network based on a higher volume of communication among them.

We can also use graphs to spot hidden relationships by looking for paths between nodes that are not directly connected. Shortest path algorithms find an optimal route between two nodes based on the chosen criteria. In the simplest case, this may be the number of edges used to get from one to the other, so we consider the path that goes along the fewest edges to be the shortest.

Alternatively we might assign each edge a weight or cost. The overall cost of a path is the sum of the costs of all the edges it contains. In this case, we would consider a path of many edges with low costs to be shorter than a path of fewer but more expensive edges. Consider a graph of a transport network in which places are nodes and different means of travel between them are edges—train, bus, or walking routes—with the time taken by each assigned as the edge's cost. If your priority is to get from A to B while making the fewest possible connections, then you should calculate the shortest path using the number of edges. However, if you care about making the journey as fast as possible, then you should use the costs to calculate your route.

IDOL's graph server provides graph functionality that complements IDOL's existing text analysis capabilities and allows users to explore their data in new ways. Once users have decided which entities and relationships they wish to model as nodes and edges, IDOL can create a graph automatically as part of its indexing process. Graph server allows multiple edge types to be configured and provides an inbuilt algorithm for calculating edge weights. That provides a great deal of flexibility in how the data is represented. Additionally, our existing pattern-matching technology can be used to find more complex measures of similarity between graph nodes than are available to standalone graphs.

Predictive analytics

Strategists, sales leaders, and product managers dream of having the ability to predict outcomes, whether it is to plan better, to proactively address issues and opportunities, or to make safer bets. Being able to make an accurate prediction can lead to competitive advantage, improved customer experience, and reduced costs for many businesses.

The basic principle is that there are patterns in an organization that are reflected in its data and that these patterns typically signal and influence where projects, actions, tasks, or opportunities will lead. We can often identify these patterns using machine-learning techniques.

The challenge with machine learning is that it usually requires an experienced data scientist to engineer and optimize an effective predictive model. Data scientists, especially good ones, are rare, particularly as the number of techniques involved is large. Moreover, the cost to maintain a team of data scientists is often prohibitive.

IDOL's predictive capabilities allow untrained teams to create and optimize such analytical models using APIs that involve little more than uploading the data set and posing a question to be solved by the system.

The platform leverages a wide set of algorithms including random forest, logistic regression, support vector machines, and Naïve Bayes for analyzing and creating a machine learning model to perform extrapolation. Rather than restricting to particular techniques it tries all suitable techniques to determine the best performing as well as a machine learning optimizer, which automatically chooses the best set of parameters for each algorithm without any user intervention. Once the models are trained, IDOL automatically chooses the model that is most accurate, while at the same time ensuring that overfitting doesn't occur.

Pattern recognition

Pattern recognition, much like its companion theory of pattern matching, consists of techniques to determine structures (or patterns) in apparently noisy data sets. This seemingly simple task gives rise to a huge number of data analytics technologies that are able to extract patterns of particular types that have clear applications in any number of fields. For example, automatic number plate recognition (ANPR) techniques are pattern-matching methods to locate one or more vehicle license plates in an image or video stream.

IDOL uses pattern recognition techniques on all data types. Within speech processing, pattern recognition is used in techniques such as speaker identification, music recognition, or broader audio model creation such as those that detect gunshots or breaking glass. Within image and video processing, the same theoretical framework is applied to scene detection, face recognition, image similarity matching, or object detection, among many others. And in the field of text processing, pattern recognition allows the “summarization” of a document or set of documents that can then be used for conceptual similarity matching, document clustering, or other applications. We examine each of these fields in the following sections.

Textual pattern recognition

From a pattern recognition standpoint, unstructured text documents generally consist of noisy streams of low-information data and are well suited to the theories of pattern recognition. Following indexing into IDOL, a number of distinct methods are available to allow accurate retrieval of the documents.

Basic search
At index time, all index fields are processed to extract relevant terms and concepts for storage in an inverted index. A large number of properties are stored for each occurrence of a term, including the document and field that it occurred in, its position within the field, its capitalization, stemming, any explicit weighting applied to it, and even the sentence and paragraph that it appeared in. All of this is stored for ease of retrieval at query time.

At query time, IDOL loads this information for all terms that appear in the query and determines the documents that match the query, whether it is Boolean, conceptual, or key word search. IDOL stores the information so that a list of all documents in which a particular term appears can be easily found, along with the information on the occurrences of that term within the document.

This additional information is used to perform relevance calculations. Basic measures such as occurrences in titles or more highly weighted fields or the number of times a term appears are used to give relevance boosts to certain documents. Any Boolean operators or phrase markers also affect the matches and the weighting, and more complex actions such as the proximity of query terms enable a query for **Hilary Clinton** to match the politician in preference to a document talking about Hilary Benn and Bill Clinton.

Alongside this, engine-wide information augments the accuracy of the matching process. Information gleaned from occurrences of terms and phrases across the entire corpus build a data-specific language model that is used to improve relevance. For example, a query for **weather news** should give far greater importance to documents talking about weather than those just mentioning news. This principle can be extended to enable full natural language and conceptual search.

Conceptual search

The probabilistic approach of IDOL's index and retrieval process allows complex operations to occur naturally. This augments basic retrieval to allow more subtle connections to be determined and more relevant results returned than are possible in any key word engine.

As an example, imagine we are interested in the effect of pollution on penguins. The traditional approach to finding information to satisfy our interest would be to select a key word search engine and type in the word **penguin**. This would return useful content, but also a significant amount of irrelevant content about a publishing company, the chocolate biscuit, Batman and Robin, and so on.

In our case, however, we are interested in content that has a high probability of being about penguins, the birds. A document containing the word sea could be about penguins but sea occurs in many contexts, and therefore there is a significant probability that the content is about something else. However, if a document contains the words **black**, **white**, **flightless**, **feather**, **slick**, and **oil**, the probability that the document is not about penguins and pollution becomes extremely low. Furthermore, this has been identified without using the word **penguin** and instead using a larger amount of weaker information, any of which can be taken away without significantly affecting the probability. The Autonomy approach understands context based on either strong concepts and key words or a larger amount of weaker information.

To do this, IDOL needs a framework to encapsulate concepts such as penguins, the birds, or news about the weather. For this purpose conceptual agents are used.

Conceptual agents

For reasons of clarity and scalability, it is a basic requirement that any system wishing to analyze streams of unstructured text be able to distill them into a reduced form that is conducive to additional processing. Within IDOL these are known as conceptual agents (or simply agents).

Agents encapsulate a concept or topic and are generated from a piece or pieces of unstructured text or one or more documents. Internally they are stored as a set of terms and associated weights, where both the terms and weights have been selected to best represent the concept in its future usage. This selection uses a number of pieces of information, including the number of occurrences and the proximity of certain sets of terms, but also the engine-wide language model that IDOL has built up to determine which terms contain the most information or entropy. It is these statistics, gleaned from examination of the corpus of documents as a whole and constantly updated as new documents are indexed, that allows intelligent selection of terms and concepts from a document. The mathematical analysis of the texts within the corpus treats such terms as abstract symbols of meaning, with understanding derived from context without the need for rigid grammars. Typically between 20 and 40 terms are used for an agent, though some concepts can be encapsulated using fewer terms, whereas others may require a lot more.

An agent typically acts as a saved search, so search criteria can be reused to find new documents. In such cases an agent is seeded with a piece of natural language query text, and then documents are chosen from its results set to describe the exact scope of the agent more precisely.

Other uses for agents include hyperlinking, user profiles, categories, and clusters. These are described in the following sections.

Hyperlinking

Hyperlinking uses a document or set of documents to find other documents that are conceptually similar. This functionality is particularly used in interfaces to find “more documents like this,” such as finding news relevant to the email or presentation that you’re writing.

To achieve this, IDOL uses its pattern-matching techniques to create an agent that represents the document or documents and then matches that agent against the corpus of documents to find the most conceptually similar.

User profiles

Security is of prime importance to corporate users of IDOL, and IDOL’s index and workflow are designed to match this. As a result, most of an end user’s interactions with IDOL involve the passing of credentials that indicate the access rights of the individual. If desired, IDOL can then use this to store the activity, preferences, and patterns of usage of each individual, again in a secure way.

The creation of user profiles is one typical use for this is. Profiles are created automatically for users whenever they interact with IDOL. They are agents designed to encapsulate users’ interests and are generated from the documents that they create, edit, or view. For example, a user with an interest in the Latin American telecoms industry will automatically have profiles that reflect this. This means that the latest Brazilian telecoms news can be passed to the user without having to press a key, and that the system is able to correctly interpret ambiguous search terms based on that knowledge (known as intent-based ranking). Moreover, profiles can be used as part of the wider community to automatically create links to users with a similar interest (known as an expertise network).

In addition, users can manually create their own personal conceptual agents for finding documents on particular topics on demand, or to have new documents on that topic delivered to them automatically. And they can share these agents with their corporate community.

Categories and clusters

IDOL's ability to classify unstructured content is a powerful application of machine learning. There are two main techniques. In categorization, examples of documents in each category from a predefined set are provided to IDOL, and IDOL's pattern recognition libraries are used to train category agents that can then be used to categorize unsorted documents. In clustering, IDOL creates cluster agents from an otherwise unseen corpus by identifying major topics in the data, which in turn can be used to track those clusters over time or categorize other documents. We'll discuss both of these techniques in further depth in the Classification section.

Versatile matching—the power of IDOL's approach

We can see the power of IDOL's conceptual agents in how they are able to interact with documents, queries, and each other. For example, when a document is used as a query against an IDOL containing documents, the result is the technique described above as hyperlinking. Similarly, to categorize people into predefined categories (people categorization), a person (represented by a user profile) is used as a query against an IDOL containing agents. Alternatively, query criteria—such as key word, natural language, or Boolean search—can be stored in an IDOL, so when a document is fired against them, only the criteria that apply to that document are returned, giving an incredibly scalable way of storing Boolean expressions that you wish to evaluate future documents against. The following table shows that all combinations are possible.

Table 1: Combinations of sources vs. targets

TARGET / SOURCE	CRITERIA	DOCUMENT	PERSON	AGENT
CRITERIA	Query expansion	Document search	Expertise search	Agent search
DOCUMENT	Boolean categorization	Hyperlinking	Profiling	Conceptual categorization
PERSON	People categorization	Profile search	Community	People categorization
AGENT	Agent categorization	Query-time categorization	Expertise search	Agent suggest

Classification

Classification encompasses a wide range of activities in the field of information retrieval, but we can summarize it as simply the process in which data is grouped or tagged in order to simplify the retrieval process for the user.

IDOL performs classification natively as part of the ingestion stream. Once documents have been classified, the user can take advantage of this via methods such as parametric retrieval, sorting and routing, advanced filtering, or complex visualization techniques.

IDOL's core classification technology is centered on the same advanced pattern-matching techniques as its retrieval algorithms. Its patented methods use key results from Bayesian inference and information theory to automatically identify the patterns that occur naturally within textual data. Via nonlinear adaptive digital signal processing methods, the analysis of text highlights statistics to correspond to specific ideas and concepts. In this way, conceptual questions can be readily reduced to probabilistic equations for mathematical analysis.

Both scalable and language independent, we can now apply the techniques to almost all forms of digital content. The same methods allow computers to achieve automatic understanding of streams of audio and video, such as televisual feeds or telephone conversations, as well as any form of textual content.

Categorization
Categorization is the simplest form of classification and sorts text documents into predefined categories.

In conceptual categorization, categories are automatically created from a set of pre-tagged documents via a "training" stage during which document examples of each of the categories are identified manually and passed to IDOL for training. IDOL then analyzes the documents with reference to a larger untagged corpus of documents to create a conceptual agent that represents that category. Once all categories have been trained, the system is ready to categorize unseen documents.

As well as avoiding costly manual category creation, the use of conceptual agents returns a probabilistic weighting for each category match. This means that rather than a simple yes/no answer, the weighting allows easy thresholding and lets us classify documents into multiple categories.

In addition, IDOL supports the full range of key word, Boolean, and proximity operations from simple Boolean expressions to highly complex explicit weighting constructs. Even legacy systems such as these are subject to IDOL's patented conceptual weighting, bringing new life to otherwise static expressions. And the expressions can be used in combination with agents to act as hard filters on top of the conceptual weighting. In many such systems, the majority of the cost involved in supporting a Boolean taxonomy is the manpower required to create the expressions, iteratively adding and removing terms in order to home in on the desired rule. IDOL can even reduce the costs associated with supporting these rules, offering interactive tools to generate, test, and manipulate new or existing expressions as well as suggesting terms and phrases that you could add to improve the expressions.

IDOL's core categorization techniques are largely based on Bayesian methods, which provide the versatility required to achieve high accuracy on the wide variety of data that it receives. As well as being language independent, the same data analytics techniques work as well on financial data as they do on pharmaceutical data and as well on small equally sized categories as they do on unequal categories. And far from being a black box, all aspects of the classification method can be controlled and tested via configuration and training parameters.

In addition, there are native Bayesian techniques optimized for binary decisions (e.g., "Is this email spam?" Or, "Is this document suitable for children?"). In such situations, most classification techniques are lacking, being designed to identify documents that contain only limited characteristics, and thus only applying to a small subset of any corpus of documents. Broad questions that slice the corpus into two sections such as those listed above are clearly not easily defined by a small set of examples and could never be wholly described by even a large Boolean expression. IDOL's BinaryCat classifier specifically addresses such situations. It can perform detailed analysis on two sets of training documents (one representing **yes** and the other no), and then extrapolate using its statistical knowledge of the corpus to generate a category that can then be used to classify any future documents.

IDOL also offers random forest-based techniques that are optimized for short extracts such as the categorization of tweets. These techniques work on fine-grained pattern matching that is also ideal for categories that are defined by a small number of positional or structural features, such as formatting or the positional placement of a piece of text within a document.

Clustering

Categorization is a supervised learning technology, as it learns to mimic the classification as demonstrated by the human-tagged training documents. Clustering, by contrast, is unsupervised learning as it is given no such human guidance. In clustering, a corpus or defined subset of a corpus is analyzed to group it into self-similar sets of documents. The result is a partitioning of the corpus into a set of clusters, each with its own conceptual agent that we can use to perform further analysis or categorize future documents into the same set of clusters.

At the first stage of clustering, IDOL analyzes each document and determines its primary concepts using its understanding of how the frequency and relationships of terms correlate with meaning. It then associates statistical measures to each concept based on adaptive probabilistic concept modeling (APCM) weighting. APCM is a core proprietary technology within IDOL. Features and concepts are identified within documents using techniques analogous to the creation of a conceptual agent. These techniques assign an importance to the concepts as well as to their relationships. These are used, for example, in finding similar documents or seeing to which type of documents a profile relates. One of the fundamental ideas behind the APCM weights is that the provided corpus of knowledge you are dealing with provides you with an appropriate worldview.

Similarity measures between documents are easily derived from the statistics generated for each document, and a hierarchic agglomerative clustering method is applied to form nascent clusters as the data is agglomerated. These clusters are then analyzed to determine the strength of each and to ascertain coverage and consistency of the set as a whole. Weak, inconsistent, or unhelpful clusters are removed from the eventual set.

Finally, additional information is derived from the clusters, such as a cluster title to allow users to readily identify the scope of each. The finished set is then available for examination or rendering in any one of a number of graphical formats, such as two- or three-dimensional cluster maps. Furthermore, you can analyze sets of clusters covering different time periods to identify information such as that which persists over time (in the Spectrograph visualization), that which is particularly strong (What’s hot?), and breaking stories (What’s new?).

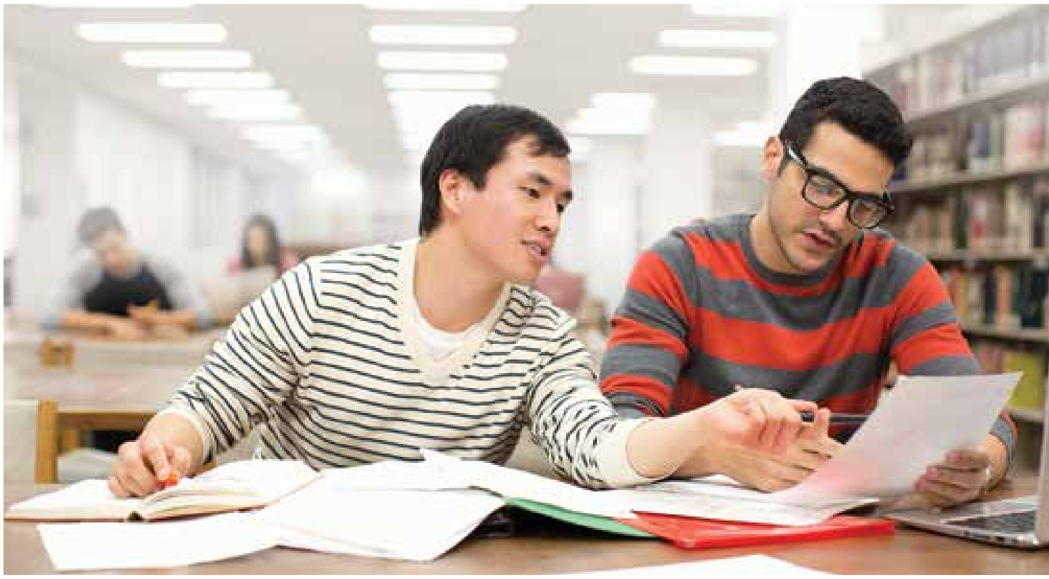
As well as drawing on core Bayesian and pattern-matching techniques, IDOL uses a novel approach to clustering based on results from quantum mechanics in which a quantum wave function is generated around each document so that scalable, incremental clustering can easily be done without the need to restart calculations if a new document is added.

Eduction

IDOL offers extensible, scalable matching of predefined entities via its eduction capabilities. The eduction module performs the full range of intelligent entity extraction within IDOL, automatically identifying metadata from documents. The proprietary data analytics algorithms that underpin the code are able to analyze both semi-structured and unstructured text to extract a huge variety of metadata, regardless of the formatting of the text.

As well as standard out-of-the-box entities such as dates, people’s names, places, addresses, phone numbers, email addresses, SSN, and so on, eduction provides a sophisticated language to allow the definition of custom-made entities. This configuration allows for full dictionary and regular expression matching resulting in a fully comprehensive set of extraction functionality.

Moreover, IDOL’s statistical and pattern-matching technology allows eduction to go one further; metadata rules can also be trained by example. A user provides a few examples of entity formats, from which IDOL is then able to automatically develop internal rules that it can subsequently use to locate similar entities within the text.



Sentiment analysis
The detection of the sentiment of a document—for example, whether a movie review is positive, negative, or neutral—is a specialized form of categorization. It works by identifying positive and negative sentiments within a piece of text through linguistic and statistical methods. The results of this can be used to highlight extracts of a document that show sentiment as well as identifying the topic of the sentiment. For example, in a restaurant review, identifying that the food and the location are reviewed positively, but that the service is reviewed negatively.

IDOL's sentiment analysis relies on two major IDOL technologies. The first is the eduction grammar module that allows the creation of extensible linguistic rules and patterns to define positive and negative expressions. The second is the categorization module that allows classification of text into positive, negative, and neutral categories. This second method has the advantage of performing the machine learning to identify the features that define positive and negative documents automatically, and thus it works well on unusual documents or language usage.

Standard sentiment grammars are available for a number of languages and others are frequently created.

Rich media classification
We can equally apply sentiment analysis techniques to rich media data such as audio files, images, and videos. For example, we can classify audio as speech, music, silence, and so on, and we can identify sounds such as gunshots or alarms. A voice can be classified as calm or angry—particularly useful in call center monitoring systems.

We'll treat the classification of audio, image, and video in more detail in the relevant sections on pattern recognition.

Audio pattern recognition

IDOL uses deep learning, through the use of artificial neural networks, to deliver state-of-the-art audio processing.

Artificial neural networks have been around since the 1950s, and the idea of using them for speech recognition since the 1980s. In fact, IDOL has been using neural networks for parts of the speech technology since its inception in the 1990s. In more recent years, however, neural networks have become the state of the art for speech recognition. This is due to significant investment in speech recognition research over recent years and advancements in hardware. The sheer scale of neural networks being trained and used for speech recognition was simply not possible with readily available hardware until recently.

Neural networks appear to do a much better job of generalizing the speech sounds than the statistical algorithms and are much more accurate. This is because speech sounds are a little more complex than the statistical models used previously and are approximated better with neural networks.

Advancements in hardware have been critical to this progress. Processing power is much faster now. But more importantly for speech, graphics processing units, originally developed for rendering of computer graphics in computer games, have provided optimized matrix multiplication tasks—the most time-critical aspect of the neural network training process—while CPU extensions allowing parallelism have improved run-time performance.

Speech to text

Speech to text is the process of translating spoken words into text. It is used in many contexts to analyze, search, and process audio content, such as command-and-control systems, dictation software, audio and video search, or subtitling.

IDOL's speech-to-text technology is trained on many hours of example speech and language data to learn the patterns of speech. This training process produces language models, which make up our language packs. We model both the acoustics and linguistics of a particular language. The acoustic model finds probable (phonetic) speech sounds in the speech audio, which is then combined with the lexicon and language model to find the most likely sequence of words and phrases.

Language customization and acoustic adaptation

IDOL requires language packs to perform speech processing tasks. A language pack contains a language model and an acoustic model. The two key components of the language model are the dictionaries of vocabulary and pronunciation, plus the corpus N-gram word probabilities.

The language model covers a broad vocabulary, reflecting the general spoken language. However, for a system that covers specialized topics, such as financial or medical topics, the standard language model might be missing specialized vocabulary or sentence structures. In such cases, IDOL can build custom language models.

Building a new language model requires a lot of text—on the order of millions or billions of words—and the standard language packs are usually built with many billions of words of text. Therefore the best way to customize a language model is to build a small custom language model that uses the specialized text, and then combine it with the standard language model.

In addition, IDOL allows adaptation of the acoustic models that are available out of the box to more closely match the acoustic properties of particular sets of audio data. Adapting the model using data that closely represents (in terms of recording quality and accents) the desired audio results in improved speech-to-text results.

Audio analytics

IDOL uses a number of proprietary audio processing techniques to allow additional analytical techniques on the audio streams. For example, speaker segmentation—which determines the transitions between one speaker and another—and speaker identification—which identifies speakers based on their voice characteristics—both use signal processing techniques to extract key features from the spoken audio and then overlay pattern matching to determine the likelihood of a particular speaker for a particular segment. We can train this system using samples of speech from each speaker to create speaker templates or using preloaded templates such as those that determine the gender of a speaker.

Spoken language identification is the process of determining which language is being spoken. It is not necessary to identify the spoken words in the content to determine the language. IDOL first tries to identify the speech sounds or phonemes in the speech and then chooses a language that has the closest distribution of phonemes. As well as detecting a large number of languages out of the box, we can expand the system by building user-defined language classifiers. These are trained with speech samples in the relevant language.

Transcript alignment assigns time codes to all the words in an audio transcript, even if they contain noise and missing sections. The generated time codes are normally accurate to within half a second. This is used in systems such as those generating automatic subtitles from manual transcripts or adding the ability to jump to a given position in audio by word. This functionality can in turn be used for checking script adherence to determine, for example, whether a call center operator is sticking to a pre-agreed script.

IDOL allows automatic classification of audio as music, noise, or speech. This can be useful when you are running speech to text on audio content that may contain music. You can combine these operations such that a speech-to-text transcript is only produced for those audio segments classified as speech. The same pattern-matching techniques extend to allow training of other categories of audio. Standardly available categories include security classification, which can detect and label segments of audio that contain security-related sounds like alarms, car alarms, breaking glass, screams, and gunshots.

In addition, IDOL performs a number of other operations on audio content, such as computing the signal-to-noise ratio and identifying clipping of the audio signal to determine quality of that audio.

Audio recognition

Also known as acoustic fingerprinting, audio fingerprint identification generates a digital summary of an audio sample to identify it quickly or to locate similar samples in a database. This has many use cases, such as the identification of songs or jingles, the detection of advertisements, or the tagging of media tracks such as "President Obama's inaugural speech." In each case the system is able to use an essentially unlimited number of pieces of audio for training its database, and the audio sample to be identified does not need to be an exact copy of the original.

Phonetic phrase search

Phonetic search is the process of searching for words and phrases by their pronunciation.

Phonemes are the fundamental units of sounds that make up a spoken language. For example, the word **catch** contains the three phonemes or sounds: k–a–ch.

The phoneme identification engine first processes an audio file to create a time track of phonemes, which reports the time at which each phoneme occurs in the file. This is a one-time process. IDOL then searches the phoneme time track data for the specified words or phrases. On an average desktop computer, the search process can operate at hundreds of times faster than real time.

Phonetic phrase search is language dependent.

It is preferable to perform a full speech-to-text operation rather than just a phonetic phrase search as the full speech-to-text operation opens up the full set of IDOL operations, including conceptual search. However, there are cases where you may have specific requirements to use key word and phrase identification and want to limit hardware resources. Phonetic phrase search can be used in those instances.

Phonetic search is particularly useful in cases where the inevitable inaccuracies of ordinary transcription may result in documents being missed when a search is performed. With phonetic search, a search for **fraud** could still return results that were mistakenly transcribed as **Ford** and mark them with a lower confidence level, allowing the user to threshold the search according to requirements.

Image pattern recognition

Humans recognize objects, people, or locations seemingly effortlessly. When we see an object for the first time, we study it, memorize its unique visual properties, and make a mental model that stays with us for a while. When we see it again, we try to match the visual properties of the object with those of the models stored in our brains. IDOL uses a similar approach to automate the process of object recognition from images. IDOL provides computer vision algorithms for recognizing repetitive or unusual patterns in images of objects, text, people, and scenes.

Typically, the raw input data varies due to a multitude of factors, so the dimensionality of the input data is fairly large. To simplify the task of spotting patterns in high dimensional input, the input is mapped to a small number of newly chosen dimensions. This process is called feature extraction. Sometimes, the features are chosen to compress the input data, rather than reduce dimensionality. For example, we might want to simplify images of text into simpler patterns like lines, loops, and dots. In other cases, if the input dimensionality is fairly low, and it is not possible to separate the data into patterns, we may map the input into higher dimensions to better distinguish the patterns. When dealing with perspective distortion, we may want to look at second order and third order derivatives of the image to correctly identify the distortion. The features are chosen to maximize the information content of the input data for the task at hand. The traditional engineering approach to pattern recognition uses heuristics to choose features, whereas the machine learning approach to computer vision learns the features automatically from a training image set. In either case, the features are chosen based on some understanding of the problem to be solved and are subject to the desired output quality and accuracy metric.

Once we choose the features, we encode the features and the relationships among them to create a model. Improved Internet connectivity and cheaper cameras have led to a large amount of partially labelled visual data being available. We employ statistical pattern recognition techniques, including neural networks, to learn probabilistic models from such data. In other cases, the problem involves estimating unknown quantities from often noisy input data. There we employ Bayesian inference techniques. In cases where there is a limited amount of training data available, we use engineering approaches to find the best overall model.

Optical character recognition

One of the earliest pattern recognition challenges was that of optical character recognition (OCR), which aims to decipher text from images of machine-printed text. To begin with, we segment the input image in order to separate the foreground text from the background. At this stage, we have to deal with undesirable scanning artifacts or lighting effects such as shadows or specular reflections. Then we group nearby foreground regions to form potential words and feed these to a character classifier. The resulting words are then checked against a dictionary and some language rules before a decision can be made about the word. This process may be iterative, so that several adjoining words may be combined to form larger words, and composite words may be broken down into smaller words. IDOL OCR supports a large number of scripts and languages and also provides automatic language identification. It has been widely used to read text from documents or photos in an automated fashion, saving many hours of laborious manual work.

Bar code recognition

A related problem is that of bar code recognition. The techniques used for bar code recognition are similar to those used in OCR, except that we train the algorithm to look for lines (linear bar codes) or squares (QR codes) instead of alphanumeric characters. Once the lines or squares are located, a potential bar code decoding is determined. Just like OCR, this is then fed to a classifier to classify groups of lines or squares. The classifier checks against common bar code standards and applies error correction where required. The simplicity of the features (lines or rectangles) and adherence to a few commonly used bar code standards makes bar codes robust to noisy input data. This is why bar code recognition is almost universally used in supermarkets for inventory tracking, shipping, and logistics. QR codes contain squares arranged in a 2D matrix such that the pattern includes alignment and sizing information. This enhances the storage capacity. More importantly, they can be read by digital cameras making them the most commonly scanned codes on mobile phones. IDOL provides robust bar code recognition algorithms that can identify multiple bar codes and QR codes within an image at various orientations.

Human image processing

The process of locating objects described by certain characteristics within an image is called object detection. Face detection is probably the most successful example of object detection. It aims to find the locations of all the faces in an image. First, visual features are extracted to look for certain frequently occurring patterns within facial features, such as the classic “T” shape that is formed by the eyes and the nose. Each patch is then classified into a face or non-face region by looking at the patterns in the features extracted within that patch. Next, the results of the patch classifiers are combined using a boosted classifier.

A boosted classifier uses the weighted sum of several weak classifiers that answer simple questions, leading to a much stronger classifier that can answer complex questions. During training, the weights of these weak classifiers are learned, and during detection, a cascade mechanism is used to quickly discard negative samples. We can use IDOL’s face detection in conjunction with face recognition to find and identify specific people within image or video data. We can detect faces in fairly low quality images and deal with group photos in noisy backgrounds. IDOL fast detection and recognition is already used by a number of entities in security, surveillance, and consumer applications.

A related but harder problem is that of pedestrian detection. While a face has a well-defined shape and appearance, pedestrians can look very different depending on what they are wearing, how far they are from the camera, whether they are moving, and whether they are on their own or in a group. In safety applications, the pedestrians to be detected are too far away from the camera for face detection to work reliably. Also, we may wish to detect pedestrians even when they have their back to the camera. There are some similarities between the techniques used for face detection and pedestrian detection, but different features are used and we generally look for the "Ω shape" depicting the head and the shoulders.

Pedestrian detection is heavily used in vehicle safety applications to automatically detect and alert drivers if there are any pedestrians on the road ahead of them. Where video data is available, the motion information can provide very useful cues about the locations of objects and also help with separating the foreground objects from the background.

One application of pedestrian detection is people counting. It is fairly common these days to get alerts relating to overcrowding or congestion at tube stations, railway stations, music festivals, or sporting events.

People counting most commonly uses several classifiers to determine whether each region of the image contains one person, a pair of persons, a small number of people such as a family, or a large group of people. Once the classification output is available, some post processing and prior information about the scene are used to come up with a final count for the number of people in a given frame. Motion cues obtained from video data greatly aid this analysis.

People counting is frequently used to understand what represents normal traffic and what looks like overcrowding. The systems are trained to handle regular patterns such as rush hour traffic at railway stations, but can throw an alert when rush hour traffic is seen at off-peak times. People counting goes beyond pedestrian detection by analyzing not just the number of people, but also their grouping and movement patterns. Overcrowding due to sporting events at a railway station is more likely to be a result of groups of people entering the stations and moving together as a group rather than individual commuters moving at a fairly predictable speed in predictable routes. We also make use of Bayesian inference techniques to flag unusual behavior of people movement based on prior knowledge about the location.

Further applications of human image processing involve analyzing properties such as age, gender, facial expressions, and various other attributes such as spectacles, facial hair, or hair color. Such additional information can be very useful in reducing the number of identities examined when searching over large databases, leading to quicker search results.

Face recognition

Once a face has been located and analyzed in an image, the next step is to associate the face with a person. Face recognition is one of the most popular applications of computer vision in everyday life. We recognize faces with such ease that it belies the highly complex processing that is going on behind the scenes. Computer algorithms started off being much worse than humans in this regard, but several decades of research in this area have greatly improved the performance of face recognition algorithms. In fact, humans' ability to distinguish faces they do not know well is poor, and current state-of-the-art face recognition algorithms have begun to surpass human performance in the case of less familiar faces.

IDOL's face recognition algorithm uses deep convolutional neural networks. The neural network is trained on a large number of faces to learn the distinctive facial features from several photos of a person as well as the differences in the distinctive features extracted from photos of one person and another. A person's appearance might change with lighting, viewpoint angle, facial expressions, or makeup. In addition, the test images may be of low resolution or may suffer from video artifacts. Our training process accounts for these factors to produce highly distinctive features for each face. Once IDOL computes the features, the task of recognizing a person is just a matter of searching for the best match within the database. Face recognition is most often associated with safety and security applications, but a wide range of consumer applications such as photo editing, visual authentication, patient monitoring, and parental control also increasingly benefit from face recognition.

Object recognition

There are situations where you may wish to recognize specific objects, such as the trademark of a company or the packaging of a particular product. Although the object is well defined, object recognition can be challenging because the object may be viewed from a different viewpoint, against a cluttered background, and with partial occlusions. IDOL provides algorithms for 2D as well as 3D rigid object recognition.

The first step in object recognition is to extract distinctive features from the input image. The features need to be descriptive enough to allow identification but also compact enough to allow efficient storage. They also need to be tolerant to various viewpoint and lighting variations. The features and the spatial relationships between the locations of these features are stored in a model in the database. When a test image is to be identified, features extracted from the test image are compared to those in the database.

A tree-based structure can first be used to get approximate matches quickly, a selection of which are refined later. Once the approximate matches are found, a voting algorithm is used to determine the most likely matching model. The voting algorithm rejects sets of matches that are not consistent with the expected geometric position and scale constraints. The identification of 3D objects poses a further challenge as the geometric relationships between the features are more complex, and only a fraction of the total number of features are visible from a given viewpoint. IDOL uses advanced geometric computer vision techniques to deal with challenges in 3D object recognition.

IDOL's object recognition is widely used in the retail industry for inventory management, targeted marketing, and advertising. It's also used in visual authentication to control access to certain areas and in gaming to provide custom-made gaming content.

Yet another familiar example of text pattern recognition from visual data is that of automatic number plate recognition (ANPR). Again, the principles are roughly the same as in OCR, but only once the number plate has been located in the image of the vehicle. This may be challenging if the vehicle carrying the number plate is moving fast, as the captured image may be blurred. The number plate may be dirty, or it may not have enough contrast. License plates around the world come in many different formats. Image stabilization and number plate extraction are among the biggest challenges in automatic number plate recognition.

IDOL uses advanced stabilization algorithms to provide a clean number plate image and then uses a neural network-based classifier for identifying the characters within the number plate. We can also apply advanced object recognition in vehicle monitoring to identify the make and manufacturer of a vehicle, which, in combination with ANPR, allows automatic identification of vehicles with stolen license plates. In addition, IDOL's ANPR has successfully been used to spot uninsured vehicles, detect traffic light violations, or monitor vehicle movement within a certain area.

IDOL's capability to extract sparse yet robust invariant local features and apply geometric constraints enables it to measure similarity between image regions. Until now, algorithms for image similarity have largely relied on crude statistics such as color histograms, gradient histograms, or block averaging. However, these approaches fail when images are altered or deliberately tampered with. For example, one may have a cropped version of the original image, the image may be mirrored, there may be some textual tags added, or the image might contain the same objects but viewed from a different angle. Any such change in the image would make it look different from the original image if only global statistics are used.

IDOL has the capability to compare images based on their content. It uses robust local features to extract similar parts in a pair of images such that the result is unaffected by cropping, mirroring, or textual tags. IDOL can recognize the similarity in spite of difference in viewpoint, partial occlusion, or image degradation. Not only that, it also provides insight into the image transformation (scaling, translation, and perspective distortion) that relates the two images. This capability allows a user to use an image itself as a search query rather than having to describe the content of the image to form the query. Text-based queries for image search are susceptible to mistakes in tags and metadata associated with the images. In general, image-based searches lead to much more relevant results compared to solely text-based queries. Applications of this technology include image similarity search, detection of fake artworks, detection of pirated movies, and shopping based on visual search, to name a few.

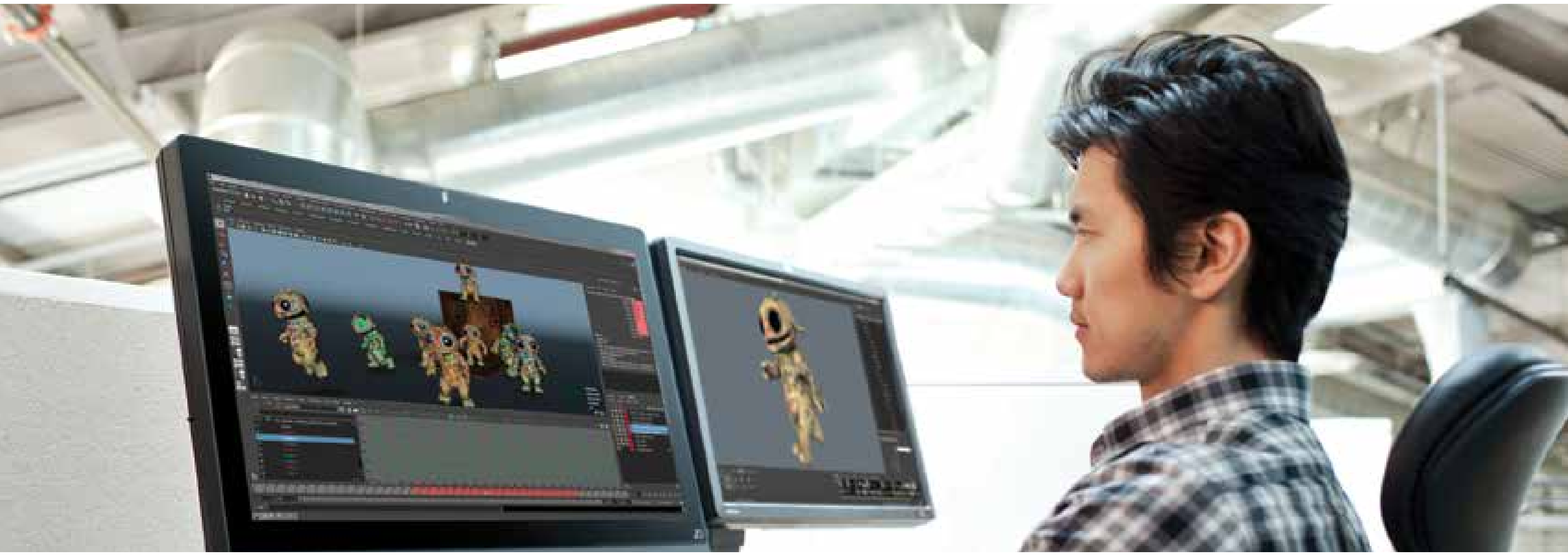
Image classification

Image classification identifies the broad categories that objects belong to rather than looking for specific instances of them—for example, all cars rather than a Ferrari 488 or a Mercedes S-class. Image classification is essentially category-level recognition and is much harder than specific object recognition because it needs to deal with variation within the class as well as variation across the classes. The categories can be as broad or narrow as the user defines them to be. The broader the categories, the harder the problem gets.

Image classification allows users to automatically tag large amounts of visual data with labels that describe the content of the images. This makes it possible to compare images on the basis of the semantic visual content. That opens up a wide variety of applications such as identifying patterns or trends in photo collections, searching for content of a specific type within a large collection of photos, and searching for all images that satisfy a certain set of requirements such as "a man with a red car and a black dog."

IDOL's image classification algorithm uses deep convolutional neural networks to learn the features that best describe the variations within and across object categories. We also provide off-the-shelf operation for users wishing to tag large number of images. For each input image, the classifier produces a set of labels and associated scores that describe the most important content in the image. The labels correspond to the several hundred categories that the classifier is trained to recognize and include labels such as everyday objects, common animals, or locations. Along with the descriptions of the image content, we also provide probabilistic scores for them, so that the most important categories can be automatically searched and stored.

The categories used in image classification may not always be known a priori. For example, a local council might have a website for accepting photos from its residents. Most of the time, the photos will not have anything meaningful, but when there is a natural disaster, there might be pictures of the affected area. IDOL's automated image classification algorithm analyzes such pictures and sorts them into categories such as normal vs. flooded or normal vs. fire. As the categories may only be determined on a case-by-case basis, we provide the facility to train the classifiers. This allows IDOL users to label the images with the categories they wish to use, rather than being limited to the categories that come with the pre-trained classifiers.



Geometric computer vision

Geometric computer vision is an important field of technology for IDOL. It combines knowledge from fields as diverse as geometry, statistics, physics, computer science, and physiology to provide an understanding of the appearance and shape of the three-dimensional world. Any 3D object when viewed from a different angle produces a different 2D image. The process of projecting the 3D world onto a two-dimensional image plane is lossy, and it is the aim of geometric computer vision to recover this lost information given enough views of the scene. A good understanding of projective geometry also enables us to synthesize the appearance given the viewpoint or determine the viewpoint given the appearance. Both of these capabilities can be vital in a large number of applications.

We highlight three particular academic areas that we use to realize computer vision in IDOL: simultaneous localization and mapping (SLAM), 3D reconstruction, and change detection.

Simultaneous localization and mapping
Most smartphones include a visual camera, and these cameras are getting smaller, better, and more powerful. This has led to a growing appetite for intelligent scene learning on mobile devices using just the single camera without any additional infrared input. A class of algorithms called simultaneous localization and mapping (SLAM) allows users to simultaneously track and learn the shape and appearance of a 3D scene. SLAM provides an instantaneous camera position and a sparse point cloud depicting the full 3D positions of the most distinctive objects in the scene.

SLAM can also be viewed as a graph optimization problem. The graph is made up of nodes that represent the camera positions (poses). The common observations of the objects in the scene form the connections between the nodes in the graph. Given the appearances and camera position at various locations in the trajectory, we can predict the camera position at an unknown viewpoint. This can be split into two processes that run simultaneously. Tracking is the process of constructing the pose graph—it matches previously seen objects to objects in the current view and estimates the camera position. Mapping is the process of optimizing the pose graph—it identifies new objects in the scene, adds and updates their representations, and refines the positions and distances of previously seen objects. Tracking then uses the updated map obtained from the mapping process to find matching objects in the current frame.

SLAM is initialized with two frames from a video of the scene. The objects that appear in both frames are matched, and the 3D position of the object is estimated by triangulation. The 3D positions of the matched objects and the camera position associated with each salient view gives us an initial map. Our algorithm automatically chooses these two frames; thereby making the complex process of initializing SLAM transparent to the user. An automatic initialization algorithm is particularly important in situations where the SLAM algorithm has limited control over the camera movement and the scene being viewed or user interaction is not feasible, such as footage captured by an unmanned aerial vehicle (UAV).

To make our tracking robust to rapid camera movements and occasional occlusion, we use position, orientation, and camera pose constraints to limit the area of the map that needs to be searched. This makes our tracker not only fast but also very accurate, as we do not match against irrelevant parts of the map. Tracking can occasionally fail due to occlusion, rapid motion, or lack of interesting features. As pose constraints cannot be used when tracking fails, we use an aggressively optimized approximate-nearest-neighbor tree-based search to find matches in the whole map. Once the matches have been found, we can estimate the camera pose, and tracking can start again.

The 3D positions of the objects and the camera poses (the map) are adjusted by the mapping thread using a gradient-descent-like process called bundle adjustment. Usually, this is the most expensive part of any SLAM implementation, as it is done over the entire map. We use error propagation to selectively include parts of the map in bundle adjustment. We make the decision to include or exclude parts of the map in the bundle-adjustment process on the basis of how they are connected in the pose graph to the currently visible part of the map. This makes our algorithm efficient and scalable, enabling us to build large maps with constant time operation. We can create maps of a run lasting many tens of minutes of a typical scene on a smartphone. On desktops, we can build much larger maps and are only limited by the amount of RAM.

Typically, SLAM algorithms struggle to create large maps as the computation scales quadratically with the size of the map, but our solution achieves initially linear and eventually constant time scalability. This means the algorithm tracks and learns at roughly the same speed throughout a large run. Our system runs in real time on most personal computers and mobile devices. The versatility of our system makes it useful across a wide range of applications. It can be used with body-worn cameras, in mobile phones, or on desktop clusters. Our SLAM system is a robust platform upon which you can build a large number of exciting 3D applications. Applications of SLAM span diverse areas such as 3D scene analysis, 3D printing, UAV footage analysis, surveillance, augmented reality, and navigational aids.

3D reconstruction
Our 3D scanning software lets you create a digital avatar of everyday objects. You just need to capture a video of the object from all directions of interest. You feed the captured video into SLAM, which generates accurate camera position and orientation estimates. Given some prior knowledge of which object in the foreground you're interested in, the algorithm separates the object of interest from the background. This foreground and background information combined with the camera position estimates from SLAM allow the algorithm to create a 3D model of the object of interest. The availability of interactive feedback means you can visualize the resulting 3D model almost instantly. This software comfortably runs on ordinary smartphones, uses the visual input from a single camera, and does not depend on lasers or infrared depth sensors. There is no limit on the size of objects that can be scanned and digitized by our software. And you do not need to put the objects on a turntable to create a digitized version of the object.

3D models created using our interactive software can then be used for various applications including 3D printing. While 3D printing is becoming faster, cheaper, and more accessible to the consumer, 3D scanning is still either too expensive or a preserve of specialists. Our technology makes creating 3D models easy, portable, and inexpensive thereby making the capability available to hobbyists and home users as well as corporate users.

Change detection
Increasing amounts of data are being captured by sources ranging from personal smartphone cameras to sophisticated surveillance equipment. Video data can be highly repetitive, so it is essential to have automated processes which reduce the amount of data that needs to be inspected by a human. Examples of events one may wish to search for include: looking for objects (such as people or vehicles) that have disappeared, spotting new objects that have appeared, or identifying objects that have moved to a different location. You can also use this technology to reveal defects in equipment or suspicious movements in surveillance applications. Automated search for such events is called change detection.

Our change detection system detects changes in 3D scenes in real time on mobile devices using a single camera. To provide instantaneous feedback, we employ a sparse 3D tracking and mapping algorithm (SLAM) rather than perform a full 3D reconstruction of the scene, because, in some situations, this can be comparatively slow.

To use the system, a 3D scene is imaged by a single moving camera. During this time, SLAM learns the 3D environment that is being imaged and stores the information about the relative locations of the objects in view as well the position and orientation of the camera itself in a map. When the scene is imaged for a second time, using the stored map as a reference, SLAM estimates the new camera position, and any changes from the original view are detected and presented to the user. Predictions about whether a part of the scene has changed in comparison to the reference scene can be made based on the knowledge of the 3D geometry of the scene. To harness this knowledge, we use rich feature matching statistics as well as viewpoint-normalized local patch matching.

In the absence of the knowledge of the 3D geometry of the scene, you would only be able to use 2D image registration methods. Any 2D image registration method can be highly inaccurate at detecting changes in a 3D scene because the appearance of an object changes with changes in viewpoint, and objects at different depths are displaced by different amounts in two views of the same scene. Once the changed regions are identified, the user is shown the original scene for reference in addition to the changed scene for inspection.

The uses of this technology are huge and underpin most of our security and surveillance applications, ranging from vehicle tracking and monitoring, to the counting and tracking of people for purposes as diverse as detecting what parts of a shop particular users visit to detecting suspicious or dangerous behavior in an external or internal scene. Even more complex examples include the analysis of footage of similar scenes taken on separate occasions, sometimes years apart, to determine activity of interest such as tampering, leaks, or security breaches.

Learn more at
hpe.com/software/idol

4AA6-4478ENW, March 2016