ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

P.PORTO

M

MESTRADO
ENGENHARIA INFORMÁTICA

Machine Learning Powered Serverless Fraud Detection

Christian Chostak

P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

M

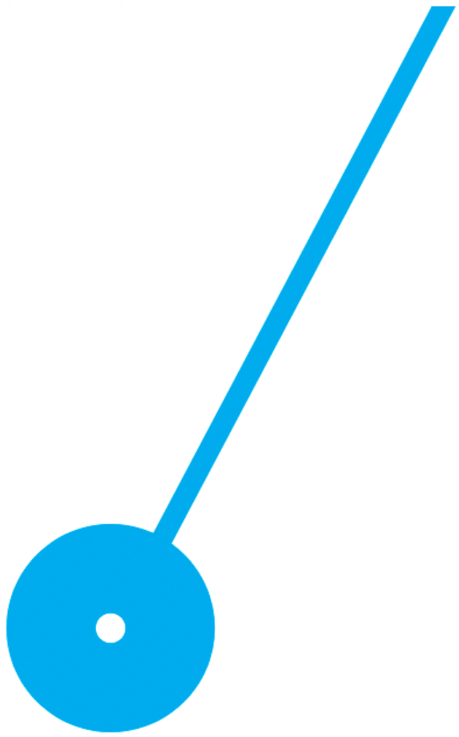
MESTRADO
ENGENHARIA INFORMÁTICA

Machine Learning powered Serverless Fraud Detection

Christian Chostak

07/2020

P.PORTO



ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

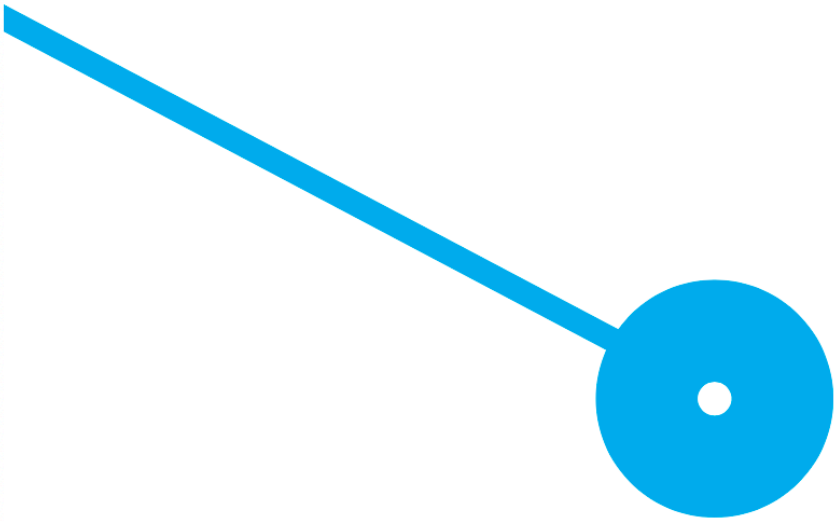
M

MESTRADO
ENGENHARIA INFORMÁTICA

Machine Learning powered Serverless Fraud Detection

Christian Chostak

07/2020



ABSTRACT

There is an increasing concern about fraud in all market sectors. Although there is a great fuzz about fraud and fraud detection, just a small fraction of it was fully incorporated into real world applications. Counterfeited documents are reproductions or imitations of the originals ones. The present work aims to fulfill a gap in fraud analysis by automating and identifying those documents in seconds. Generally speaking, a payload containing a suspect fraudulent document will reach an Application Programming Interface gateway, which will redirect the request to Lambda functions and based on the event store it on SQS - Simple Queue Service, this queue will trigger a fleet of micro-services powered by Lambda functions as well. The non-exhaustive list of functions will proceed to read this queue and in the first moment create the metadata of the received document, registering on a Serverless Relational Database, whilst storing the document itself on S3 - Simple Storage Service. After that, it will call the second batch that will start the process of machine learning on the already saved image. Triggered by the finished process, a message will go to the SNS - Simple Notification Service - alerting the user. The output of the given analysis contains a sample of the input document showing where the fraud is if there is one. With the percentage and area given, the operator will be able to see what portion of the image was considered a fraud and from that moment forward, the user will have technical basis to accept the document or not.

Keywords: Serverless, Machine Learning, Lambda, Fraud

RESUMO

Existe uma preocupação crescente sobre fraude em todos os setores da sociedade. Apesar de existir grande alvoroço sobre fraude e detecção de fraude, apenas uma pequena parte dela foi implementada em aplicações reais e ainda sim, em setor relacionados a streaming de mídia. Documentos falsificados são reproduções ou imitações, inteiras ou parciais de seus originais. O presente trabalho tenta preencher uma lacuna na análise de fraudes automatizando, e identificando-a em segundos. Em termos gerais, um payload contendo um documento fraudulento atingirá uma Interface de Programação Aplicacional - API, que então direcionará os pedidos para funções Lambda, e, baseado no evento, armazenará em SQS - Serviço de Queue Simples. Esta queue iniciará o gatilho para uma frota de micro-serviços, também executados em Lambda. A lista não exaustiva de funções prosseguirá e lerá os eventos da queue que nessa fase contém apenas um identificador único do arquivo, bem como uma breve descrição informada pelo utilizador, e, em primeiro momento criará a metadata do documento recebido, registrando-o em uma base de dados relacional, enquanto armazena o próprio documento no S3 - Serviço de Armazenamento Simples. Depois disso, iniciará o segundo lote de processamento sobre a imagem já salva, neste momento começam algoritmos de Machine Learning, bem como, processamento habitual de imagem. Iniciado pelo fim do processo, uma mensagem irá passar pelo SNS - Sistema de notificação simples, alertando o utilizador final. O relatório da análise conterá uma amostra do documento que foi processado indicando onde está a fraude, se existir uma. Com a percentagem e área indicada, o utilizador poderá ver quais porções do documento foram possivelmente alteradas e poderá considerar ou não o documento, afinal, terá base técnica para fazê-lo.

Palavras-Chave: Serverless, Machine Learning, Lambda, Fraude

Contents

List of Figures

1 Introduction

The benefits of new technology changed the way we work and live. Currently companies rely on computers for storing and sharing data, creating documents and communicating. However, Information Technology (IT) also opened up new opportunities for criminal activity and new versions of old crimes, such as white-collar fraud, and entirely new crimes, inherent to the medium itself. [1]

Smart phones and mobile devices have become the de-facto way of receiving various government and commercial services, including but not limited to e-government, fintechs, banking and sharing economy. Most of them require the input of user's personal data. Entering data via mobile phone is time consuming and error-prone. Therefore many organizations involved in these areas decide to utilize identity document analysis systems in order to improve data input processes. [2]

There is an increasing concern about fraud in all market sectors. Daily business transactions are done digitally, without the communication partners getting to know each other. The web, the advancement of technologies and the trivialization of gadgets have made it possible for documents, copies, cards and financial documents to be transferred instantly to any person or organization anywhere in the world. All of this, coupled with the alarming need of governments to de-bureaucratize their current modus operandi, makes eminent the creation of tools for control.

A simple web search can reveal more than 200,000 results for "fraud in the world". The following table (Figure 1) shows the amount of monetary loss in credit card fraud in the United States between 2012 and 2018, in billions of dollars. [3]

Although it's clear that fraud is a pejorative term and its use is counterproducent, there can be hundreds of ways of fraud. Fraud generally means obtaining services, goods or money by unethical means. Fraud deals with cases involving criminal purposes that, mostly, are difficult to identify. Credit cards are one of the most famous targets of fraud but not the only one; fraud can occur with any type of products, such as personal loans, home loans and even retail. Furthermore, the face of fraud has changed dramatically during the last few decades, as technologies have changed

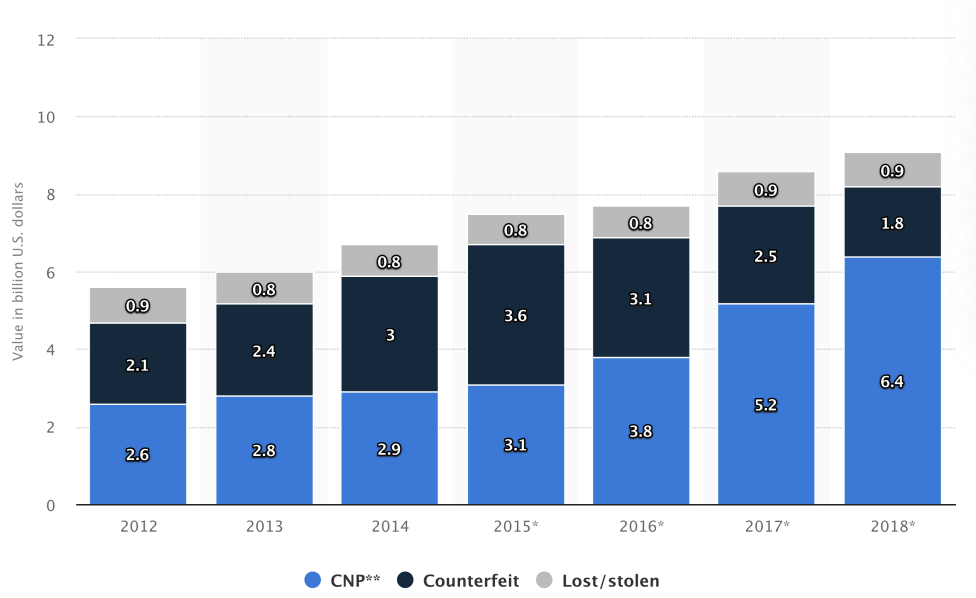


Figure 1: Credit Card fraud numbers in USA

and developed. A critical task to help businesses and financial institutions, including banks, is to take steps to prevent fraud and to deal with it efficiently and effectively, when it does happen. [4]

In Europe, the scenario looks exactly the same. Markets across Europe have made significant gains in the fight against plastic fraud (Frauds primarily concerning Credit and Debit cards), specifically in France and the United Kingdom, which achieved 6% and 8% reductions. Despite this, losses across the EMEA region grew by 30 million.

The threat of card not present fraud continues to be a key battleground for banks and retailers, as we now see a global migration of fraudulent activities. This has forced the fraudsters to migrate their efforts to new markets, with Austria, Denmark, Norway, Sweden, Poland and Russia all seeing an escalation in losses. [5]

Also, powerful publicly available image processing software packages such as Adobe PhotoShop or PaintShop Pro make digital forgeries a reality. Feathered cropping enables replacing or adding features without causing detectable edges. It is also possible to carefully cut out portions of several images and combine them together while leaving barely detectable traces. [6] Generally,

there are two known applied technologies for detecting changes in images: digital watermarking and content-based copy detection. [7]

Recent research in document forensics are mostly focused on the analysis of images of documents. One of the frequent tasks consists in retracing the course of document images, identifying the use of different printers or scanners for a single document to detect inconsistencies. Some other tasks concern the analysis of the contents of images. Printed text is analyzed to find all abnormalities: characters with identical shapes or irregular fonts, lines that are skewed, misaligned, bigger or smaller than others. Graphical elements of documents are another subject of research, such as signatures, logos or stamps, for instance [8].

By "realistic forgeries", we mean modifications that could happen in real life, as in the case of insurance fraud when fraudsters declare a more expensive price than true for objects that were damaged or stolen. Victims of fires or theft have to provide evidence of purchases, namely receipts or invoices, to prove their existence. Falsifying a receipt to earn more money from insurance is very tempting and quite simple [8].

What this particular project intend to do is not only create yet another solution to solve a common problem in the same way, but to add value to a process that needs recent technologies. The main goal is to make possible to identify frauds in any type of document, to aid a myriad of different digital processes, in a sustainable way, leveraging the best of microservices to build a scalable tool. A tool that if not solves entirely the problem, at the very least points the user to the right direction.

To be able to accomplish the difficult task of running a heavy and dense image processing application in near real time, the use of Cloud Computing is imperative.

The fact of distributed compute brings yet another problem, the cost. Solutions in this area are prone to demand a lot of money; trying to lower this cost leaves us with Serverless computing, a on-demand compute distributed system which will only occur in costs when there is a job to be ran.

2 State of the art

Although there is a great fuzz about fraud and fraud detection, just a small fraction of it was fully incorporated into real world applications, and even then, they do not focus on detecting documents per se, instead, are used to identify manipulation into digital streams. Nevertheless, it could not be different as it is one of the main sources of mass media.

Counterfeited documents are reproductions or imitations of the originals ones. The process of counterfeited documents identification is mostly manual and supported on expert's past experience. The manual analysis of all the constituent elements of the questioned document is mainly based on a digital version of the original document produced by using materials and printing techniques from available technologies. It is then carried out through different techniques and methodologies (physical and chemical examinations). Those elements may include printing process, watermarks, fluorescent fibers and planchettes, guilloche pattern, fluorescent and magnetic inks, optically variable inks, rainbow printing, microprinting, latent images, scrambled indicia, laser printing, photos, signatures, embossing stamps, optically variable devices, protective films, perforations, machine readable security, retro-reflective pattern, among others. This analysis provides information that may lead to the classification of the original document as genuine, false or forged [9].

There is little doubt that although technology keeps pushing forward to develop techniques for exposing photographic frauds, new techniques will be developed to make better fakes that are harder to detect. As with the spam/antispam and virus/antivirus game, an arms race between the forger and forensic analyst is somewhat inevitable. The field of image forensics, however, has made and will continue to make it harder and more time-consuming (but never impossible) to create a forgery that cannot be detected. [10] Though simple digital forgeries are easier to spot, the mechanical ones require a complete different approach.

Fact, there is a wide range of companies working with digital forensics, but even then, the process is automated, not automatic. And that is the gap which this product is aiming for. The idea is that it empower its users to automate a already used process identifying previously what

sort of manipulation was done in a document, so to speak the client can be totally aware and proper make up its own mind. [11] That inflicts a simple wish, to be able to detect all possible and known frauds in the market through a extensive scan in every single algorithm ever built to do so. The next part will focus on reviving the current status of this war, showing what authors around the world are doing to prevent fraud.

2.1 Image forgery detection

From the tabloid magazines to the fashion industry and in mainstream media outlets, scientific journals, political campaigns, courtrooms, and the photo hoaxes that land in our e-mail in-boxes, doctored photographs are appearing with a growing frequency and sophistication. Over the past five years, the field of digital forensics has emerged to help restore some trust to digital images. [12]

Digital watermarking has been proposed as a means by which an image can be authenticated. The drawback of this approach is that a watermark must be inserted at the time of recording, which usually cameras do. In contrast to these approaches, passive techniques for image forensics operate in the absence of any watermark or signature. These techniques work on the assumption that although digital forgeries may leave no visual clues that indicate tampering, they may alter the underlying bits of the image.

There are at least five categories mentioned [12] where image forensic tools are applied, which one focus on a simple and single statement. They are:

1. Pixel-based are the ones used to detect anomalies at pixel level, and can be subdivided in:

    - Copy-moving: Occurs when parts of the same image are used to retouch the image or hide objects (such as changing a crime scene photograph) [13].
    - Resampling: The act of resising portions of the image to adjust to the scene.
    - Splicing: Is a very common type of manipulation where two or more images are spliced together, normally to create an impression that a foreground object is part of the background taken from the other image. This may involve blurring and other kinds of additional touch-ups to make the image look authentic. This can be used by well trained forgery experts to tamper documents where letters or whole words may be removed or manipulated, changing the entire meaning of the document. Another instance would be moving a person beside another in an unlikely image, say a politician beside a terrorist to create a rumor. [13]
    - Statistical: The act of estimate if a given image is possible for a given scenario.

2. Format-based techniques try to exploit vunlnerabilities on the medium which the image has been saved.

    - Jpeg-Quantization: Most cameras encode images in Jpeg format. The compression used by each manufacturer leaves a signature on its own.
    - Double Jpeg: JPEG images which are tampered suffer from a double phenomenon known as double compression, with inconsistencies between DCT histograms of singly and doubly compressed regions. DCT coefficients of unmodified areas undergo a double JPEG compression thus exhibiting double quantization (DQ) artifacts, while DCT coefficients of tampered areas will result from a single compression and very likely present no artifacts [13].
    - Jpeg Blocking: Jpeg compression lies on a DCT block, which basically infers that the edges have horizontal and vertical traces, that can be easily disrupted on a forgery.

3. Camera-based techniques try to exploit vulnerabilities made on camera lens, sensors or on-chip postprocessing.

    - Chromatic Aberration: Tries to determine the source of light passing through the lens.
    - Sensor noise: As a digital image moves from the camera sensor to the computer memory, it undergoes a series of processing steps, including quantization, white balancing, color correction, gamma correction, filtering and, generally, JPEG compression. This processing introduces a distinct signature into the image.

4. Phisically-based techniques try to exploit vulnerabilities made by the impossible usage of light on a image.

    - Light Direction: Tries to determine the source of light on the image and if the given source is the same throughout the file. Usually, photos used on media streams have multiple light sources.

5. Geometric-based techniques try to measure objects to see if its proportions are indeed possible.

- Principal Point: In authentic images, the projection of the camera center onto the image plane is near the center of the image. When a person or object is translated in the image, the principal point is moved proportionally. For instance, on Figure 2, (a) The original image. (b) A close-up of the license plate, which is largely illegible. (c) The result of planar rectification followed by histogram equalization.
- Metric Measurements: Tries to project geometric a given image in order to allow rectification of planar surfaces.

Figure 2: Demonstration of the rectification of planar surfaces

Not only images suffer from forgery, but documents of any given place and type.

2.2 Types of document forgeries

Print, Paste and Copy (PPC) forgeries: This is done by printing the new text (presenting higher values or more items) on an empty sheet, pasting this part onto the genuine document and copy using a color copier. This technique is mostly used by people without a computer science background.

Reverse Engineered Imitations (REI) Forgeries: This is done by scanning the genuine invoice and using it as a template to generate a new document by retyping all the text, putting the logos into place, etc.

Scan, Edit and Print (SEP) Forgeries: This is done by digitizing the invoice and manipulating the digital image, e.g. increasing the price of a invoice or increasing the number of items from two to four.

Different approaches from optical document security exist. But, as normal documents do not contain any extra security features, so called intrinsic features have to be used. Intrinsic features are features that are integrated into the printout by the normal document generation process, in contrast to extrinsic features that are added solely for the task of securing a document [14].

2.3 Detection And Localization of Image and Document Forgery

As seen earlier, images can be manipulated in various ways. Different touch-ups and image manipulation techniques are applied to augment or enhance a given image.

Images are regularly re-sized and recompressed, such that they can be more easily exchanged over the Internet due to the proliferation of cloud-based photo sharing and editing websites like Flickr and Picasa, spurred by the social media applications like WhatsApp, Instagram and Snapchat [13].

These manipulations are typically not recognized as Image Tampering as the intent to manipulate the information content of the images is minimal.

The process of detecting such manipulations is termed Image Forensics. An image forensics algorithm shall output information indicating whether the image has been tampered with, as well as more importantly, identify the portions of the image that have been altered.

It is important to be able to distinguish between image enhancements from image tampering. It has been noted that image enhancement mostly do not involve local manipulations (notable exception would include Bilateral Filtering for touch-ups) while tampering at times involves local manipulations [13]. For instance, contrast and brightness adjustments are some operations which are normally not useful per se for image tampering, whereas sharpening and blurring operations may aid in image tampering to some extent and copy-move, splicing operations make up the

malicious end of the spectrum.

2.4 The challenge of fraud identification in global context

Identity documents are special in a sense that they contain sensitive personal information. This creates complications in several aspects. First, storing any personal data presents a security risk in case it's leaked, resulting in identity fraud and significant financial damages both for corresponding ID holders and the party responsible for leaking the data. Second, people understand that risk and are not as easily convinced to share their personal data with someone they don't trust. Third, identity documents are uniquely bound to their owners which makes them very rare, increasing the costs of data collection even further. Fourth, there are very few publicly available samples of each identity document and most of them are protected by copyright law which makes them unusable in research. Finally, in many countries it's illegal not only to distribute personal data but even collect and store it without a special permission. [2]

Besides the fact that a document contains sensitive information, a system that will try to recognize its forgeries must account for its special properties.

In order to provide additional level of forgery protection identity documents often have complex graphical background possibly containing guilloche, watermarks, retroreflective coating which is prone to glare, holographic security elements which change their appearance depending on relative positions of camera and document [2]. Moreover, text is printed with special fonts, sometimes using indent or embossed printing techniques.

To account for all those variables the system should try to normalize the input source. Which generally means to add a new layer of compute and save the image once again.

Modern mobile device recognition use cases usually involve unknown and uncontrolled environment and a person who handles the recognition system often is not familiar with how such system operates. Uncontrolled or unconstrained environment means that scene geometry, lighting conditions and relative movement model of the object and the capturing device are unknown. It also implies that there is no guarantee for the input data to be of high quality and indicates the possible risk of information loss. Depending on which particular frame has been captured, the

document data information might be partially or completely lost because of document not being fully present inside the frame, camera not being fully focused, bad or uneven lighting conditions, glares, camera noise and other common distortions [15].

Beyond documents, images are also used to trespass ethical and moral barriers amongst ourselves. From proving an alibi in court, to sending a receipt to the insurance company, there exists many instances where nefarious intent is the sole purpose of manipulating images, such as manipulating the dollar amount on invoice images. The proliferation of image processing software, such as Photoshop and GIMP, provides the necessary tools to achieve malicious manipulation of images with ease. [13]

2.4.1 Passive detection techniques

Passive detection techniques normally do not involve the study of the contents of the image and only concentrate on various image statistics that can be used to discern from non-tampered regions from tampered regions. Some of the techniques involve exploiting the artifacts and inconsistencies that are created due to JPEG compression used widely as an Image format. Some techniques exploit the inherent noise present in the image due to difference in Color Filter array interpolation in different cameras or inconsistencies in the local noise pattern caused due to splicing. Yet another class of algorithm looks at the lighting inconsistency [13].

2.4.2 DCT and block-level artifacts

JPEG images are compressed according to 88 Discrete Cosine Transform (DCT) blocks. Algorithms use this fact to detect tampering operations under various principles [13].

2.4.3 Error Level Analysis

Error level analysis works by intentionally resaving the JPEG image at a known error rate and then computing the difference between the images. Any modification to the picture will alter the image such that stable areas become unstable [16].

2.4.4 Block Artifact Grids

For manipulated images, when the tampered part is pasted into the background image, the DCT blocks do not match and some block artifacts will be left [17].

2.4.5 Camera and local noise residuals

Image features like Local Noise or Camera Noise arising from the image acquisition process or due to the manufacturing or hardware characteristics of a digital cameras, provide sufficient infor-

mation to determine an image's authenticity since they are sensitive to image manipulation as well as being difficult to forge synthetically [13].

2.4.6 Color Filter Array

During acquisition, every pixel receives only a single color-channel value (red, green or blue). To produce the final image, the raw data undergoes an interpolation process,using Color Filter Array (CFA) to obtain a color image with different cameras using slightly different parameters to perform the interpolation [18].

2.4.7 Purple fringinig aberration

Tamper identification based on the effects introduced in the acquired image by the optical and sensing systems of the camera. Tries to identify local artifacts arising from chromatic-abberation (Purple Fringing Aberration or PFA) due to image acquisition procedure of a camera lens. The geometric center of the image can be inducted from the PFA events. For localization, the PFA "normal flows" are used to detect tampered areas [19].

2.4.8 Local Level analysis

The noise levels are estimated by a median based estimator. The main drawback of this method is that authentic images also can contain various isolated regions with totally different variances which can be denoted as inconsistent with the rest of the image [13].

2.5 An autonomous approach

The problem is greater than what was already mentioned, take for instance the next section where machines and trained experts seek for forgeries in documents.

Airports are a huge concern for governments and should be, they receive people all over the world and have to analyse passports and IDs in real time.

Great part of officers in the border fails to identify correctly if the picture on the passport really belongs to the passenger [20]. As seen previously, images can be easily exploited, so to say, how come Airports can be effective on analysing tampered documents? The study conducted [20] leverages three conditions: Human only, Human with machine aid, and Machine only.

- Human only: Border control officers inspecting and assessing the authenticity of a document without the help of advanced technical equipment (Only UV light and magnifying glass) with and without time constraints.
- Human with machine: Border control officers inspecting and assessing the authenticity of a document with the support of an automated, not automatic, document inspection system and without time constraints.
- Machine only: Automatic document inspection systems inspecting and assessing the authenticity of a document alone without an officer physically handling the document and without time constraint.

The dataset used in the study can be referred as Test121 and used 121 real genuine and false travel documents detected at UK and Dutch borders, where ca.78.5% of the documents are passports, 19.8% are ID cards and 1.6% are Visas. Approx. 66% of all documents are EU documents. The dataset was built in such a way as to provide a realistic representation of the travel documents inspected at Schengen borders.

The types of forgeries used were:

- counterfeits, 26.7%
- replaced images, 38.4%
- substituted pages, 27.9%
- stolen blank, 7.4%

Test121 dataset was later on subdivided into:

- Test121(96) - equal number of genuine and false documents.
- Test121(25) - basically false documents.

About the participants, the testers were composed of 2 master testers, experienced members from UK and netherlands assigned to the machine only scenario. 39 EU border guards assigned to the Human only and Human assisted scenario. From the 39, 20 were from Lisbon, 16 were from Frontex reference manual working group while 3 were administrative SEF staff members, with no expertise on document inspection.

The machines were composed of a scanner and software for optical and electronic authentication of travel documents. Seven systems from seven different vendors participated. All but one were capable of:

- Authenticate passports/biopage, visa stickers and EU ID cards.
- Generating images in several spectra, from Infrared (IR) light to one or more frequencies of Ultraviolet (UV) light.
- Comparing the document to a template base.
- Read the electronic chip in ICAO compliant passports.
- Presenting the authentication results in a automated and clear way.

The results of the test were not what should be expected of a border control system.

Human only:

- On average humans are 65% accurate in the classification of documents as genuine or false.
- Non experienced testers are not better than chance (50:50).
- As expected, humans perform better in the time-unrestricted scenario.
- On average humans performed better on the genuine document detection task than on the false document detection.
- Document experts are better than border guards in both coverage and precision.

Machine only:

- Interpretation of unclear results poses a serious vulnerability.
- The same machine returned different results for the same document.
- The systems tested tended to have a higher indecision rate when inspecting false documents.

Simply put, without the experts, once again the machines wouldn't suffice the demand imposed. The reason for that is quite simple, the lack of good datasets and the myriad of possible frauds on a single given document.

Passports, IDs, Credit Cards, are just a fraction of a much bigger problem. Every process that relies on the task of identify a document nowadays is prone to receive a fraudulent one, this is because our society and the organizations inside it demands for a higher speed on dealing with its own bureaucracy.

Take for instance a Passport or an ID, you can find forgeries on the signature, photo, text and medium elements. Each of the previous mentioned elements requires different strategies. The same algorithm will not be enough to determine if the document is doctored or not.

To elucidate that, the next sections are focused on specific portions of a given document.

2.6 A handwritten signature

Take for instance a handwritten signature, used singulary in every single document that you possess. It is used to sign contracts all over the world and is, till today, one of the oldest forms of proving that you are who you claim you are.

A handwritten signature can be defined as the scripted name or legal mark of an individual, executed by hand for the purpose of authenticating writing in a permanent form. The probability of two signatures make by the same person being the same is very low. So detecting this kind of forgery is a extraordinary task [21].

There are two types of verification:

- static: offline, it is the process of verifying an electronic or paper signature after it has been made.
- dynamic: It's an online verification where the subject signs electronically on a tablet, digital board or similar device.

People from the lower level of hierarchy in society generally prefer to write their signature in free hand writing, becoming an easy target. In this particular case, 4 forgeries are possible [21]:

- Simulation forgery: It takes place when the forger has a sample of the victim's signature, and so can through trial and error forge an almost identical signature.
- Unknown/Random/Blind forgery: This is when the forger does not have any idea of how the signature looks like, so to say it's the easiest forgery to spot.
- Tracing: It's made by using light behind a paper with the signature and a blank paper, where the signature is them copied as it is.

- Optical Transfer: Used to transfer the signature from one document to another through photocopier, scanner, facsimile or photography.

The electronic capture of handwritten signatures presents novel opportunities and challenges in forensic signature analysis. Biodynamic signatures allow for the analysis of temporal handwriting characteristics, characteristics not previously possible in the examination of traditional manuscript signatures signed with an inking pen on paper. With the increasing use of electronic signatures, document examiners need to develop methods of analysis in order to reliably conduct examinations of these new technology-based signatures [22].

However, experimental research needs to be conducted to establish whether these systems are adequate in capturing handwriting features that would allow forensic document examiners to recognize the possible false negatives caused by handwriting variables (i.e., illness, disguise) or the possibility of false positives resulting from system attacks (simulation, forgery). [22]

If these concerns are not addressed,the increasing use of biodynamic electronic signatures may create significant forensic problems in signature identification cases, in that document examiners may not have the expertise or methods to examine biodynamic electronic signatures; and forensic analysis may not be reliable because of the low resolution graphic images with limited or no temporal data. [22]

Signature forgery find its way through net banking, passport verification systems, public examinations, credit card transactions and bank checks. Usually to spot an offline forgery, template matching can be used.

2.7 Applications

The applications for electronic signature technology are extensive, and they are in widespread use at an international level. Biodynamic signature software and hardware is manufactured and marketed by major corporations to areas such as finance, banking, healthcare, and mortgage lenders. Biodynamic signatures are used for access control, network access control, client identification purposes, document workflows, and electronic transaction security. [22]

They are used for contractual agreements, delivery verification, biometric security checkpoints, bank signature cards, and point-of-sale transactions. Traditional business has incorporated them into use for contractual negotiations, even in conservative business markets. It is inevitable that various forms of electronic signature technology will increase internationally as it maintains popularity over other forms of biometric analysis. Because signatures are intuitively associated with identity and are unique to the individual, they are more user-friendly and less invasive than other forms of biometric identification such as fingerprint, iris, facial, and gait recognition. [22]

Aside from the variables associated with tablets,significant changes in temporal and spatial dimensionsoccur when signatures are written with digital writingimplements in comparison to signatures written withink and paper. These differences are caused both bythe writing device and the writer's response to thewriting device. Hardware factors such as the enlargedtip of a digital writing pen and the lack of friction on adigital tablet can cause changes to a writer's naturalsignature. [22]

The differences were significant, in thatthe form details that changed between the signatureconditions could be attributed to either an altered writingenvironment or could be mistakenly attributed to theeffect of forgery.There are many limiting factors to consider whencomparing an electronic signature to samples ofmanuscript signatures. [22]

2.8 Domain knowledge based approach

Just identifying the signature is not enough, one must also retrieve valuable information from the medium, and that can be anything. But every document has its own particularity. A driver's license has expiry date, while a birth certificate has not. An insurance policy might contain the case scenarios where it might be used, while a Passport do not.

Being sensitive about the medium and retrieving information about what document is being dealt with is crucial.

The automation of information extraction is a complex task that can be divided in two main sub-problems: the extraction of the desired information and the automatic correction of OCR errors. Regarding the former several approaches have been proposed specially regarding the correction of text obtained from low quality materials or poor printed documents [23]. The problem can be broken down into pieces using domain knowledge based metadata, where first are identified the type of the field, and only then the OCR based scan.

There are two main ways to detect OCR errors: dictionary lookup and character n-gram-matching. Other methods for automatic OCR errors correction can be found, like the use of topic models, which automatically detect and represent an article semantic context [23].

The approach is composed of two main steps: in the first step we exploit domain-knowledge about possible OCR mistakes to generate a set of variants of the string extracted by the OCR. In the second step we perform a suite of syntactic and semantic checks to select from the correct string from the set of proposed ones. We also perform an aggregate semantic checking order to verify the correctness of two or more elements that are semantically linked to each other (e.g., total, taxable amount and vat in an invoice document). This approach is able to detect and correct OCR errors also in noisy documents without introducing more errors to it [23].

The usefulness of classification of fields is just as good as the correct identification of the document itself. As seen on [24] other papers, to correct classify a document a bag-of-words might be used. In that sense, text is extracted from the document, and using tf-idf the most important

words are extracted, intuitively, documents that have the same bag of words might be grouped together and be scanned in the same way.

Essentially, TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. Words that are common in a single or a small group of documents tend to have higher TF- IDF numbers than common words such as articles and prepositions [25].

2.9 Document classification

An essential step in the understanding of printed documents is the classification of such documents based on their class, i.e., on the nature of information they contain and their layout. This task is usually accomplished by extracting a suitable set of low-level features from each document which are then fed to a classifier. The quality of the results depends primarily on the classifier [26].

As referenced earlier, document classification is a crucial premise for high level document analysis, for instance when extracting and processing information from large volumes of printed documents.

Consider a system capable of extracting automatically a specified set of information items from a document, once the class (roughly corresponding to the layout) of the document is known. For example, the system could extract Date, Amount and Number from an invoice, once the emitter of the invoice is known. Or, it could extract Authors and DOI from a scientific paper once the publisher is known [26].

A classifying system is usually characterized by the following three key aspects: the features nature (i.e., what each feature means), the features representation (e.g., graphs, numeric vectors of fixed or variable size, etc.) and the classification algorithm itself zd.

Furthermore, features can be easily grouped in:

- Image features: which are extracted directly from the image, like the density of black pixels in a given region, or the number of white separation blocks in the segmented image or the gaps between column and rows.
- Structural features: which are obtained from physical or logical layout analysis. Like size, position of blocks and font size.
- Textual features: that may be computed without performing an OCR or after processing the document through an OCR.

2.9.1 Rolling image

Notice that images obtained by scanning real-world documents of the same class could be significantly different due to human errors made during their digitalization.

A frequent cause consists in positioning errors on the scanner area, due to non-standard document sizes, cut documents, and so on. To address this problem, one might find it useful to apply an automatic method called rolling which aims at aligning each document in the same way, hence obtaining images whose actual content position is substantially constant [26].

To do so an edge recognition algorithm is applied to a low-resolution version of the image obtained by resizing the original with a 1/6 scaling factor, and reducing the image in order to remove the noise caused by the scanner and small texts.

To maintain the image size, the algorithm removes all the content between the upper pixel and the top border to append it at the end of page, the flow can be easily seen on Figure 3.

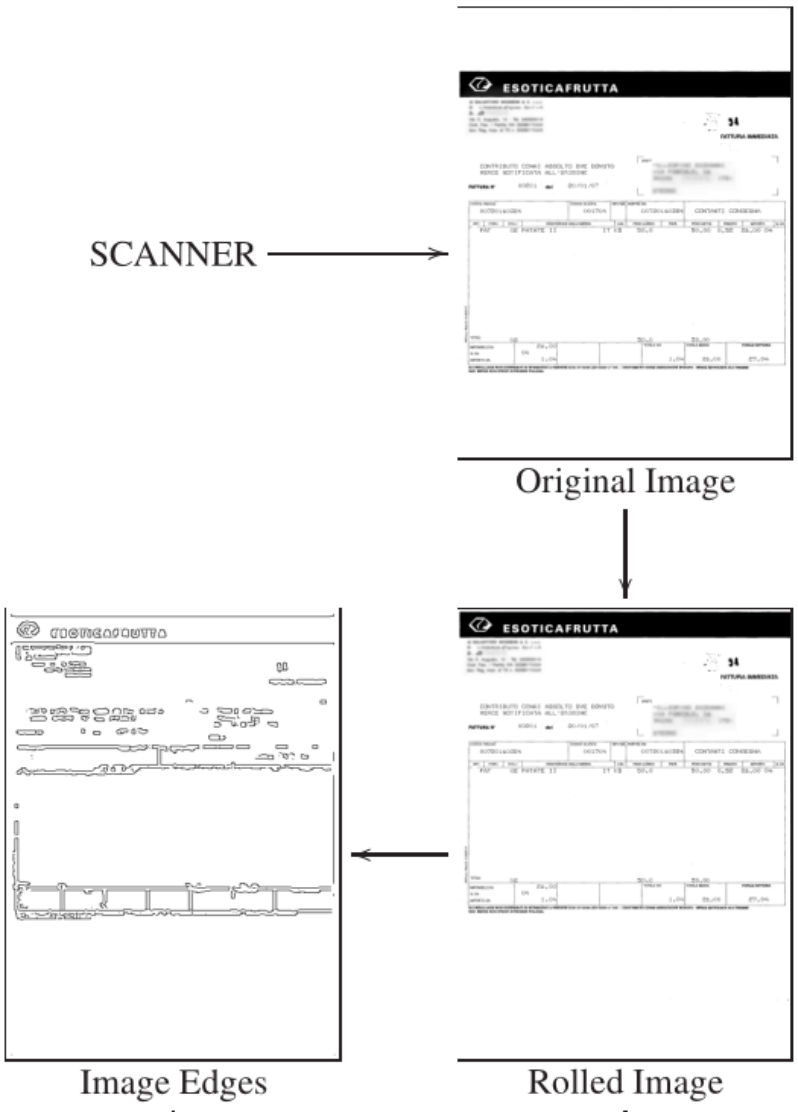


Figure 3: Example of the rolling algorithm

2.10 Text-line Examination

In every day life, document verification is an important task as many documents present a potential value. A typical example is bank notes. When handling bank notes, most people quickly control their genuineness by verifying easy to detect security signs as, e.g., holograms, structured print, and special inks. This is seamlessly done by many people as they have got used to the bank notes and their security features. A person handling an unseen bank note will not know exactly what features to look for. This person might either just trust the source or look for features it might know from previously encountered bank notes: it might, e.g., look for a watermark or a metallic stripe inside the paper. Thus, even unknown bank notes can be checked for signs of forgery [14].

For less secured documents, a verification is only feasible if the characteristic features of the document are known. Assume the following scenario: a perpetrator has signed a contract that he wants to modify in order to gain advantages over the other contracting part. He therefore adds an additional clause in some white space area of the document. This will have as effect that the security features as the genuine signatures and the genuine paper are genuine. However, this type of forgery can detected by detecting verifying the consistency of the text-lines: small variations in the rotation and the alignment of the added text-lines compared to the previously printed, genuine text [14].

2.10.1 Exploiting Intrinsic Features

For checking the plausibility of a document, the following two features are used: the skew angle of text-lines and the alignment of them, Figure 4. Other features exist that could be used for detecting anomalies in text-lines as the word spacing between them or between their letters.

To measure the alignment of a text-line, the following approach is used: first, the left and right alignment lines are computed. These lines are defined as the left and right margin lines where justified, left- and right-aligned text start or end on. A visualization of the alignment lines is depicted in picture below [14]:

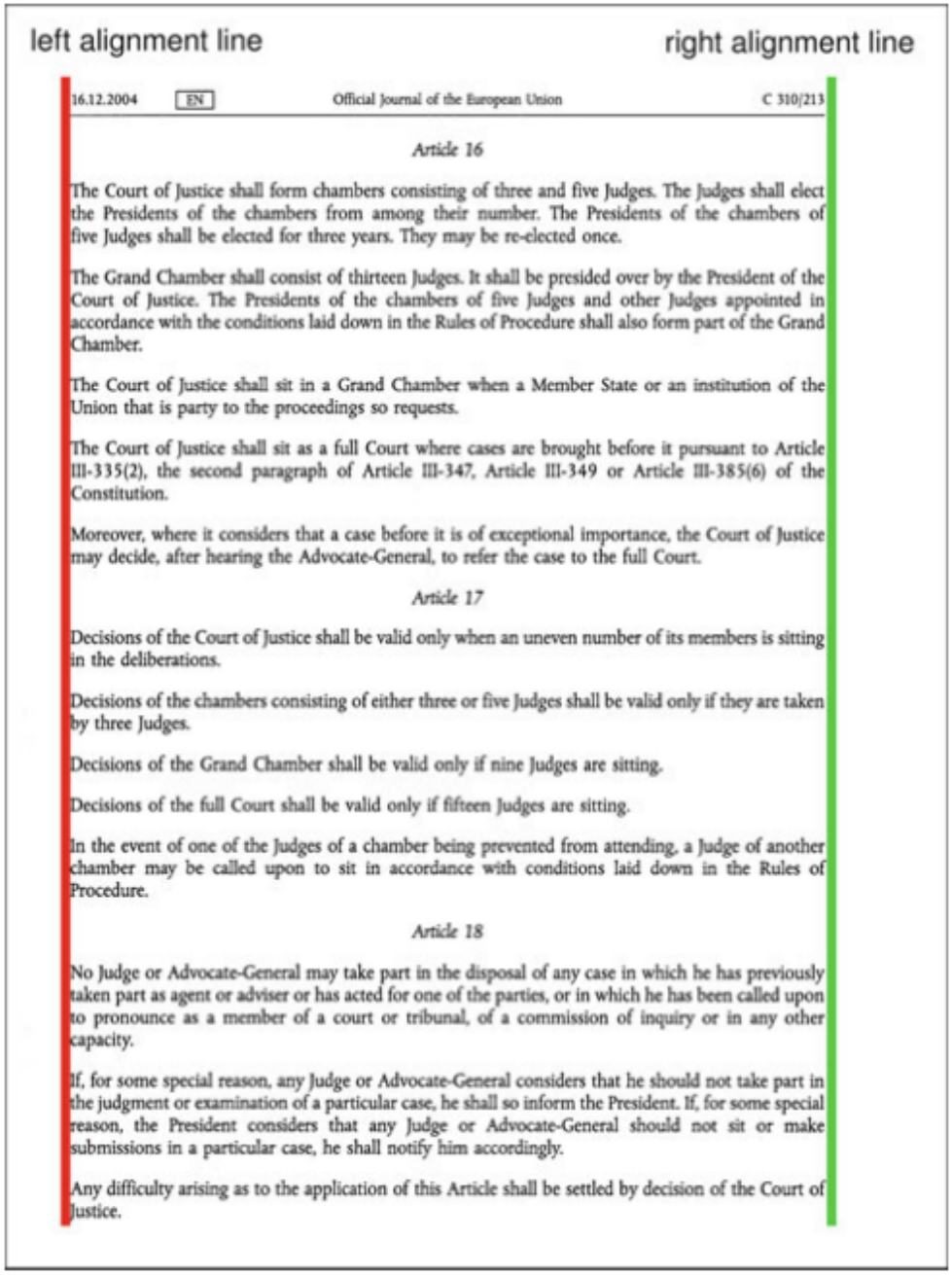


Figure 4: Example document with the left and the right alignment lines

After having extracted the alignment lines, the distance between the start and end point of a text-line to the respective alignment line is computed. These two distances are used as features to perform a plausibility check.

2.10.2 Plausibility check using skew angles

Using the skew angle of text-lines for identifying suspicious documents is a well-known technique in questioned document examination. Questioned document examiners do this step mostly manually using standard image manipulation software. The main idea of the process is sketched in next figure. The binarized document is correctly oriented and deskewed using the method presented in. Next, the text-line skew angles are extracted. These are checked if they are within the "natural" variation of text-line skew, a visual example can be seen on Figure 5. Then, the text-line is considered as valid. Else, the line is reported as an "implausible" line [14].

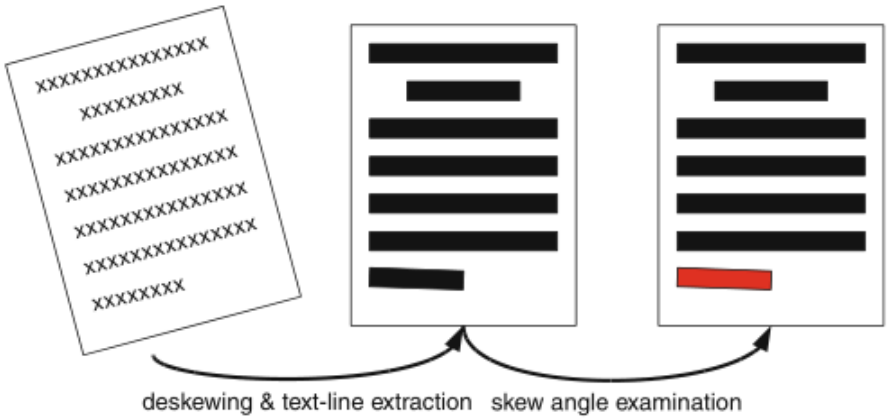


Figure 5: Visualization of the text-line skew examination: the binarized document is deskewed. The text-lines are examined if their skew angles are abnormally high or not

Using this model, a simple threshold-based method to decide whether a text-line skew angle is suspicious or not could be applied: using the confidence intervals, every skew angle outside

the 99.7% confidence interval will be reported.

2.10.3 Typographic enhancements

Note that on Figure 6 the characters “r,” “n,” “f,”and “x” are located slightly above the horizontal red line, which connects the bottom points of the characters with a round part at the base-line level, like “b” and “o”

Figure 6: Examples showing typographic enhancements.

2.10.4 Alignment line computation

After the text-line extraction, the alignment lines have to be detected. Four different alignment types are commonly distinguished in typesetting:

- Left aligned: text-lines start at the left margin.
- Right aligned: text-lines end at the right margin.
- Justified: text-lines start at the left margin and end at the right margin.
- Centered: text-lines do neither touch the right nor the left margin. The gaps between both margins are equal in size.

3 Objectives

As mentioned earlier in this document, the main purpose of the application is to evaluate any given document or image and inform the user if it has indicatives of a fraud or not, given the complete data that was extracted in an effective and fast manner. It is important to highlight the speed here, because there is no space for analysis that can take days to process in the real world, it just don't fit the current global dynamic of 'readiness'.

In the real world, documents are processed by humans and they evaluate if they fit the purpose of the entity 'on the fly'. Rare are the cases and institutions that have experts to seek if the given documents are real or not, and even then, those experts can take up to hours to give a response. And even if those experts were able to scan all documents, the data mentioned before shows that humans are no better than luck at the job. Just to point out once again the words 'instantly' and 'fast', they must be part of the application.

Also, many companies have standard contracts, files that have little to no change. If the system can validate a contract signed against it's blue print - the standard contract issued by the company -, it would be much easier to seek forgeries. The system must comport this feature, being able to compare images pixel by pixel. Showing the differences beetween them.

To accomplish the task, and be able to develop an application that will outlive the current technologies by all means, serverless must be used. Serverless will guarantee that main concerns as scalability, speed, security and mainly cost benefit are feasible.

Because of the broad nature of the scanning process, Machine Learning has to play it's role. Without it, the entire app is doomed to repeat all processes to every single document, increasing costs and velocity. Also, there is little importance to scan signature frauds on an uploaded landscape photo for instance.

Those will be the four main objectives of this work, in the next sections the methods will be covered to accomplish it.

4 Proposed Method

To be able to fulfill all the possible steps presented, this work will aim to run a series of steps concurrently, each one of them will be specific on its own. As seen earlier, the same algorithm is just not enough for all kinds of forgeries that exist, not scalable, not easily maintainable. Also, it seems unfeasible to train a Machine Learning model to look for every single aspect of forgery as each forgery requires an entire dataset to be trained.

Simply put, machine learning will be used on steps that have a generic nature, e.g. unsupervised classification of document class. Whilst custom algorithms will be used to detect anomalies on document class, after they are known. Consider this, once the application infers that the document is in fact a Driver's License, it will look more actively for frauds on the owner's photo and signature, fields that wouldn't be necessary if it was dealing with a Lease Contract for instance.

4.1 Retrieving metadata and pipeline the image

In retrospective, it seems that the logical steps to proceed with the Fraud Detection System are to first determine the metadata of the document, which is simply data about data. Retrieving metadata does not reflect or add any sort of entropy to the file itself, on contrary, it aids to the correct flow of the file though the pipeline.

Different from metadata, the steps below will introduce a tremendous quantity of entropy in the file, changing drastically its origin, thus making it improper for seeking forgeries. Therefore, after storing all the meta-data the original image will be stored. A copy from the original picture will be made and placed available in further steps.

Be aware that each of the following instructions will carry out a copy of the original file itself as they do not follow a logical order, e.g. the output of the grid image will not follow to the next pipeline. Though the system was created that way, there is room for improvement as a huge collection of methods were reused to do the same thing in different steps.

The instructions intended to be ran after the correctly retrieval of metadata and copy of the original file are:

- RGB to grayscale: In layman's terms, and RGB image is represented as matrix X, Y dimensions and depth of 3 planes, Red, Green and blue respectively, its values ranging from 0 to 255. Whereas gray image is represented as a matrix X,Y of 1 plane. DIP (Digital Image Processing) By doing the conversion, DIP decreases drastically [21].
- Noise Removal: Noise is the result of errors which is caused due to various types of acquisition process which result in pixel values that do not reflect to the true intensities. May be caused by the scanner itself, film grains in the source of noise, electronic transmission of image. To remove the noise, image averaging and mean filters are used [21].
- Resizing: The image matrix is resized to an optimal proportion as the system should be insensitive enough for the correction in the image [21].
- Crop the images to get only the desired area of the document.
- Split the image into a grid. It is usually a better bet to scan small chunks of image through parallelism - or at least faster -, than to scan the entire image.
- Seek inside each grid for blurred or sharpened portions.
- Determine copy-move forgery across the document. Just to infer if there are raw group of pixels that were copied from one part to another.
- OCR the image, extract all words.
- From the words extracted from OCR infer a bag of words using a Machine Learning tf-idf classifier in order to correct classify its class.
- Determine the document's class from the words extracted. Usually, documents have small portions that denote it's group, e.g. PASSPORT, NATIONAL DRIVER'S LICENSE, and so on.

- Check if the system has any blue print for that kind of class, namely a template. This fit one of the core objectives of the application, to compare two files, one is the organization contract that was issued to a customer, the other is the customer signed response, agreeing or not to the contract. Comparing both, pixel by pixel will denote if there are unintended portions that were adulterated.
- Determine the main chunks of the image that will be processed for forgery found on the class. e.g signatures, pictures, labels, typography.
- Types of typography used, and it's consistency throughout the file.
- Font sizes used and it's consistency throughout the file.
- Check text alignment, because usually if someone is tainting a document it won't be 'pixel-aware' of the text alignment, also, it is extremely difficult to position new text in the same alignment both horizontally and vertically on images or PDF's that no now allow change.
- Check ELA - Error Level Analysis, with it and the rainbow pixel quantity, it is easy to inform at least if the image is being 're-saved' and if it was adulterated by any forms, great for JPEG files.
- Check spaces between the lines, and with it seek if there are any spaces that escape the pattern of the file.
- Check for typography enhancements, this will show if the document has more than one type of typography, if the font changes throughout the document, or, if there are text portions that just don't fit the document pattern.

4.2 The non-intervention paradigm

Although the proposed method tries the best effort to determine what sort of forgery has been made, the place and the percentage of accuracy, it will not be focused on telling that the document

is fraudulent or not *per se*. Rather, it will infer upon high percentage that the document might be fraudulent. And that plays a huge difference on the application scope.

Analogously, there are Clinical Analysis that are made on blood samples and other fluids, and usually, the report that is given to the patient has little to no info on the patient's disease. A doctor however can read the report and usually give a prognostic to the patient. Here, the application will give the data in a structured manner, as a Clinical Analysis would. The responsibility of the accepting the analysis will be from the 'doctor', or in this case, the decision of accepting the systems output it entirely up to the system's operator.

Simply put, in order to accomplish the above mentioned steps, different technologies and frameworks will be used, the next section focus on explaining the architectural choices, and what each one of of those choices will properly do.

5 The intended result

For sure the complexity cited here may entrench into the public's mind the feeling that all of this may not work properly. And even that should be seen as a good and positive reason for the developing of the whole. The algorithm will be trained in a set of previously forged documents, for the most common causes that may occur, and will be not created or intended to be so a one-solution fits-all kind of application.

The great challenges that still lies on the table is how to reduce the already known main flaws of the serverless architecture, namely: function event-data injection, broken authentication, insecure serverless deployment configuration, over-privileged function permissions and roles, inadequate function monitoring and logging, insecure third-party dependencies, insecure application secrets storage, denial of service and financial resource exhaustion, serverless business logic manipulation, improper exception handling and verbose error messages, legacy / unused functions & cloud resources, cross-execution data persistency. [27]

But the concerns, problematic as may be seen, are all preventable, if not neglected. In the other hand, the positive outcomes [28] are way greater, namely cost, infinite elasticity and reduced security overhead.

The application herein refered will be created to right a wrong [29], but also to add an additional layer of security to business transactions, and that airports, hospitals, banks or any other related market sector can benefit from. The idea behind it is that with minimal effort, anyone can send a picture, image or document to a Web API and receive in seconds if that particular document is fraudulent or not.

5.1 Innovative contributions

To reach out the speed and availability necessary for this kind of project, a 100% serverless application will be developed, guaranteeing 99.99999% availability that will make beyond the storage of the probable images of fraud, the detection of the location of the fraud in the document, of what type it is characterized and what percentage of the algorithm has agreed that the document is indeed a fraud. At the same time, providing the analysis in near real-time, customized to each class of document by it's use of Machine Learning.

Serverless itself has many benefits as compared to more traditional, server-based approaches. Lambda handlers from different customers share common pools of servers managed by the cloud provider, so developers need not worry about server management. Handlers are typically written in languages such as JavaScript or Python; by sharing the runtime environment across functions, the code specific to a particular application will typically be small, and hence it is inexpensive to send the handler code to any worker in a cluster. Finally, applications can scale up rapidly without needing to start new servers. In this manner, the Lambda model represents the logical conclusion of the evolution of sharing between applications, from hardware to operating systems to (finally) the runtime environments themselves. [30]

Not only that, but by leveraging the use of serverless, you reach a new level of distributed services. There ain't a single worry about scalability, fault-tolerance and high availability. All of those items come free, out-of-the-box. Plus, all the traffic in and out of the applications goes through HTTPS. Comparing to old models where the communication between all the parts of the same system had to implement security, this is for sure a bonus.

Of course, the downsides of the framework exist, and although they will be discussed in more details later on, there are a few to mention:

- It's all about numbers, because the code and all libraries used must account for less than 50mb you have to develop a clean app, with few third party dependencies in order to work. True, there is the possibility of using handlers that will load your code on the fly from an

external source, but that just add even more latency to your start-up.

- There is a storage, but it is limited to 500mb and it will disappear as soon as your code stop.
- Applications will have a higher latency, period. But that latency won't go higher than 200ms.
- Costs are inexistent if there is no one using the application. And when there is someone using, payment is only by its use.
- The complexity of the code increases drastically.
- The footprint and debugging becomes messy if not careful.

The following picture (Figure 7) demonstrates at a higher level what a serverless application looks like:

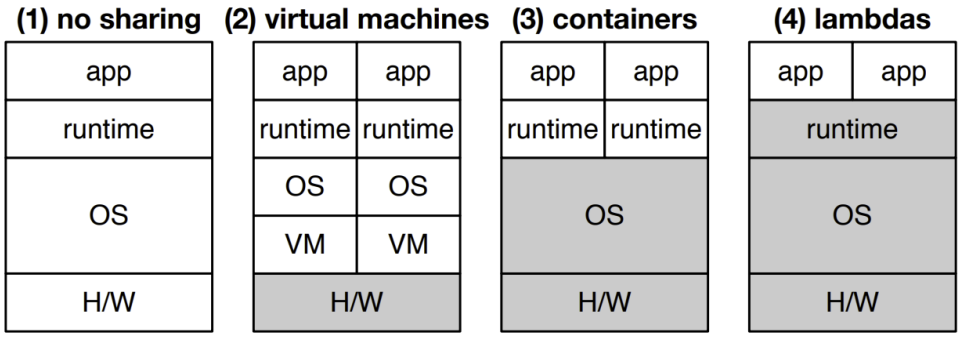


Figure 7: Shared layer model

5.2 Technical overview

As mentioned earlier several frameworks will be used to assist the development, among them there are mentions to Django and Zappa. The main development language will be Python and Javascript. For development of the machine-learning algorithm SciPy, TensorFlow and Matplotlib. For hosting and storage solution, S3, Glacier and DynamoDB.

This particular Technological Stack was not choose blindly as the following sections will cover. The core business here is fraud detection, so there is no guarantee in what time the peaks will occur, nor what size of queries it will attend. In this particular scenario there is no need for guessing if one can manage to provide a tool that auto-scale itself when needed. If needed.

Plus, when dealing with prototypes and proof of concepts the language barrier represents time. Choosing a high level language as Python just solves the problem of not having to deal with languages particularities. Also, its huge community gives access to a higher number of libraries and wrappers, generally involved in data science, machine learning and automation.

There is at least three big players at the present moment on Cloud business, AWS, Azure and Google. All of them possess the serverless paradigm discussed previously. The choice between them is also trivial an unimportant as the frameworks that will be used will abstract all possible third party integrations needed.

To exemplify the desired architecture, see the following schema on Figure 8.

As one can see, the entire traffic will come and be server through an API Gateway, that will direct, based on URI path to the correct micro-service. The flow is:

- A user access the main page.
- After registration, and e-mail confirmation it became authorized to start using both website and API.
- Accessing the website, it can upload a single document (.pdf, .jpg, .png), while using the API it can make bulk uploads as needed.

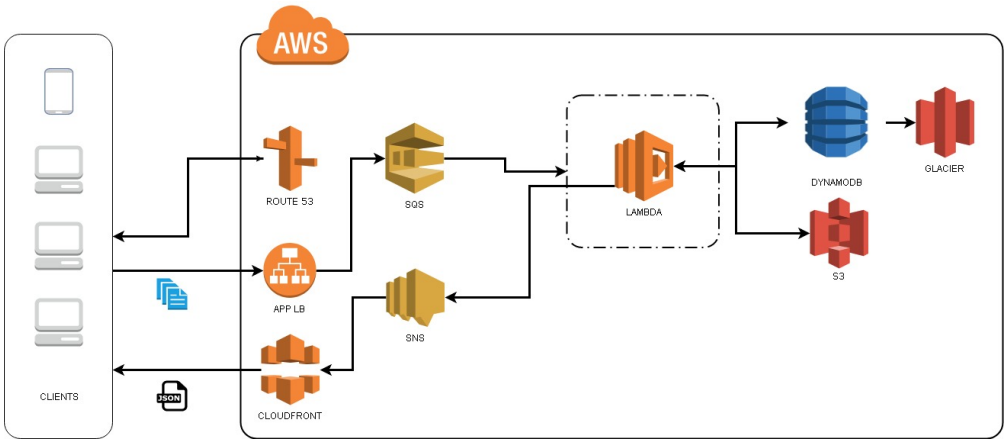


Figure 8: Serverless architecture overview

- Each document uploaded will be retained in a blob storage, where a uuid named folder will be assigned as well as an upload id. Also, a message will be created in a topic 'uploaded_files' containing its own metadata. This will allow for micro-service orchestration as it detaches the business logic into event queues.

- One service specialized in reading the topic 'uploaded_files' will query the topic and split the message into all the possible pipelines existent. If more steps are needed it can be simply improved by creating a new topic in this service.

- From here on, there is a fleet of micro-services. Namely:

  - apply_filters
  - create_histogram
  - create_hue
  - create_rags
  - crop
  - detect_cm
  - detect_ela
  - detect_faces

  - detect_superpixel
  - extract_signature
  - line_detection
  - ocr_extraction
  - remove_transparency
  - resize_image
  - ret_digest
  - ret_jfif
  - rexif
  - rotate_image
  - slice_images
  - tf_idf

- There are jobs, not described above that are responsible for feeding the initial stratum for Machine Learning jobs. Those will be supervised algorithms which will be fed and maintained manually, independent of the document upload process.

- Each micro-service contains solely a django app, its own needed libraries and a connection string to a single table in a given database. They query their own topic, deleting the message after a successful run.

- There is a final service that queries the topics seeking if the uuid generated upon upload was successful, if it was, it will trigger an e-mail or sms, depending of user registration about the task completion. The user will be then prompted to download the generated report.

Micro-services were the only possible and feasible choice, as each one of them requires specific libraries and the overall size has to be below 50mb. While a traditional development would allow all the code to be deployed as it is, here common files will repeat itself, namely the django setting file and boto3 SDK.

6 Architecture

6.1 Django

Django is a high-level open-source Python Web framework that encourages rapid development and clean, pragmatic design. Django was open-sourced in summer 2005. Project was supported by other open-source projects such as Apache, Python and PostgreSQL [31].

This basic Django way to create web-services uses Model Template View (MTV) paradigm, which is essentially Model View Controller [32].

In the MVC paradigm, the model represents the data, the view shows it and the controller controls what is done to the data. The user sees the view and uses the controller to modify the model, which is then updated in the view again. The model acts as the data storage [33].

In a regular Django application, requests are directed by the urls.py config file. Django parses the Uniform Resource Locator (URL) and redirects the data to different modules. These modules include the view (controller), data models and templates. When creating a new Django module (app) it creates a folder with app's name and the file structure. Template files are not created because the view.py can be used to render HTML as well. Making the app easier to manage. This is particular usefull as one can use it to serve the system API through it.

When choosing a framework to work with, Django was chosen for its easy setup and easy approach. Django encourages developers to separate different concerns of the system in to small parts, and thus generalizes everything in such a way that the user only needs to create models with the desired structure and functionality. Plus, being made with Python turns easy the adoption of python scripts to automate the workflow [34].

Django's functionality is based on small apps and their interaction with each other's. Each app has own models and URL configurations.

Unlike in some other frameworks, in Django not only the data type is defined, but things such as relation to other data models and restrictions of the data. This means that Django models

contain some of the validation process already inside the model declaration.

The doubts on the current project were amongst which libraries to use and what would be the most suitable framework to develop in.

In that sense, the next topics will focus on which libraries were chosen as well as a gentle introduction to why Python.

6.2 Python

The Python programming language is establishing itself as one of the most popular languages for scientific computing. Thanks to its high-level interactive nature and its maturing ecosystem of scientific libraries, it is an appealing choice for algorithmic development and exploratory data analysis. Yet, as a general-purpose language, it is increasingly used not only in academic settings but also in industry [35].

Python is a cross-platform, general-purpose, object-oriented programming language, increasing popular in sectors of mathematics, biophysics, machine learning and web development. Python programs are usually interpreted instead of compiled to platform-specific binaries which allows a fast development cycle [31].

Python is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++. The language founded in the year 1991 by the developer Guido Van Rossum has the programming easy and fun to do. The Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features [36].

As mentioned before, the main idea was to add speed to development, taking advantage of it's ease of use, general-purpose and cross-platform nature. Also, to leverage it's full potential, Pillow as used as one of the main libraries to deal with image binaries.

6.3 Pillow

The Python Imaging Library adds image processing capabilities to the Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities [37].

The core image library is designed for fast access to data stored in a few basic pixel formats. It provides a solid foundation for a general image processing tool.

Example of a small application in Python for metadata extraction using Pillow.

```
import os
import sys
from PIL import Image
from PIL.ExifTags import TAGS

image = sys.argv[1]

for (tag,value) in Image.open(image)._getexif().iteritems():
        print '%s = %s' % (TAGS.get(tag), value)
```

The Python Imaging Library is ideal for image archival and batch processing applications. The library contains basic image processing functionality, including point operations, filtering with a set of built-in convolution kernels, and colour space conversions [38].

The library also supports image resizing, rotation and arbitrary affine transforms. Although simple, Python and the ability to handle images is just not enough. The application has to be able to iterate over the image bits and extract from it useful information, to add that capability, Numpy will be used.

6.4 OpenCV and its algorithms

OpenCV is a very well-known and widely used open source library for computer vision. It has tools for digital image processing and includes a set of algorithms for pattern detection and image comparison,briefly explained below. The Harris Corner Detection algorithm was developed by Chris Harris and Mike Stephens. The underpinning mathematic model used to detect corners and edges considers window in the image and then determine the average changes of image intensity. The result obtained is achieved by shifting the window by a small number of pixels in various directions. The detection of corners in digital images is made by comparing the same area in both documents. Lowe's Scale-Invariant Feature Transform algorithm (SIFT) is another algorithm to detect corners. SIFT is meant to be invariant to image scale and rotation, that is invariant when the image is zoomed out or zoomed in. Rosten and Drummond developed the Fast Algorithm for Corner Detection (FAST) that may have better performance in real time applications.

OpenCV framework has a wide set of interesting functionalities for pattern detection in digital images, besides the core implementation. Homography is one of such functionalities that refers to the detection of an image inside another, by a matching templates [9].

6.5 Numpy

NumPy is the fundamental package for scientific computing with Python. It contains among other things [39]:

- A powerful N-dimensional array object;
- Sophisticated (broadcasting) functions;
- Tools for integrating C/C++ and Fortran code;
- Useful linear algebra, Fourier transform, and random number capabilities;

Besides its obvious scientific uses [40] [41], NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

6.6 SciKit-Image

Scikit-image is an image processing Python package that works with numpy arrays. The package is imported as skimage [42]:

Within scikit-image, images are represented as NumPy arrays. The skimage.data sub-module provides a set of functions returning example images, that can be used to get started quickly on using scikit-image's functions:

```
import skimage
from skimage import data
camera = data.camera()
type(camera)
<type 'numpy.ndarray'>

# An image with 512 rows and 512 columns
camera.shape
(512, 512)

coins = data.coins()
from skimage import filters
threshold_value = filters.threshold_otsu(coins)
threshold_value
107
```

```
import os
filename = os.path.join(skimage.data_dir, 'moon.png')
from skimage import io
moon = io.imread(filename)
```

Images in scikit-image are represented by NumPy ndarrays. Hence, many common operations can be achieved using standard NumPy methods for manipulating arrays, in NumPy indexing, the first dimension (camera.shape[0]) corresponds to rows, while the second (camera.shape[1]) corresponds to columns, with the origin (camera[0, 0]) at the top-left corner. This matches matrix/linear algebra notation, but is in contrast to Cartesian (x, y) coordinates.

All of the above remains true for color images. A color image is a NumPy array with an additional trailing dimension for the channels:

```
cat = data.chelsea()
type(cat)
<type 'numpy.ndarray'>

cat.shape
(300, 451, 3)
```

This shows that cat is a 300-by-451 pixel image with three channels (red, green, and blue).

Till now, image is covered, the previous libraries are more than enough to load an image, extract information, iterate over pixels. But what about text? The system will handle documents and plus the image side, it must access text easily. To do so, Tesseract will be used.

6.7 Tesseract

Optical character Recognition (OCR) allows machine to recognize the text automatically [43] [44].

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and "read" the text embedded in images.

Python-tesseract is a wrapper for Google's Tesseract-OCR Engine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others [45].

OCR is not a solution that solves a hundred percent of the problems. Usually images have distortions or are so beyond legible that recognizing characters become difficult. To improve Pytesseract recognition [45]:

- Clean the image arrays so there is only text(font generated, not handwritten). The edges of letters should be without distortion. Apply threshold. Also apply smoothing filters.
- Resize the image with text you want to recognize to higher resolution.
- Pytesseract should generally recognize letters of any kind, but by installing font in which the text is written superbly increase accuracy.

6.8 Machine Learning

That said, covering the basics, the system is currently ready for being accessed for a higher level API, one that can detect desired patterns on documents. To accomplish that, Machine Learning tasks will be used.

Machine learning is the science of computers acting without being explicitly programmed. Data plays a vital role in this, with the given data knowledge is derived to solve a problem. Machines are trained on the basis of past experiences, user behaviour and data to take decisions in future [46]. Machine learning has given us self driving cars, practical speech recognition , effective web search and an understanding of human genome. Deep learning is a subfield of machine learning whose algorithms are based on functionalities, structure of brain [47].

Machine learning contains a set of methods, which allow a machine to learn meaningful patterns from data directly with minimal human interaction. The strength of a machine learning technique is, in part, dependent on human knowledge. Such knowledge can help a machine to learn more efficiently through techniques like appropriate feature selection, transfer learning, and multitask learning [48].

Learning problems fall into a few categories, but the most used one, for the job herein referred will be classification and unsupervised learning.

An example of a classification problem would be handwritten digit recognition, in which the aim is to assign each input vector to one of a finite number of discrete categories. Another way to think of classification is as a discrete (as opposed to continuous) form of supervised learning where one has a limited number of categories and for each of the n samples provided, one is to try to label them with the correct category or class [49].

unsupervised learning, consists of a set of input vectors x without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization [49].

For development of the Machine Learning tasks, Scikit-learn library will be used, because of it's high level API and it's real life use scenarios [50] [51].

It's main use will be to classify documents through all of possible classes, in an effort of unsupervised classification learning.

Answering the growing need for statistical data analysis by non-specialists in the software and web industries, as well as in fields outside of computer-science, such as biology or physics. Scikit-learn differs from other machine learning toolboxes in Python for various reasons [35]:

- It is distributed under the BSD license.

- It incorporates compiled code for efficiency, unlike MDP and pybrain.
- It depends only on numpy and scipy to facilitate easy distribution, unlike pymvpa that has optional dependencies such as R and shogun.
- It focuses on imperative programming, unlike pybrain which uses a data-flow framework.

6.8.1 NLP

What is Document Classifier? Document classification is an example of Machine Learning (ML) in the form of Natural Language Processing (NLP).

Steps for Document Classification 1. The dataset The quality of the tagged dataset is by far the most important component of a statistical NLP classifier. The dataset needs to be large enough to have an adequate number of documents in each class. For 500 possible document categories, you may require 100 documents per category so a total of 50,000 documents may be required. The dataset also needs to be of a high enough quality in terms of how distinct the documents in the different categories are from each other to allow clear delineation between the categories. 2. Preprocessing In our dataset we should given equal importance to each and every word when creating document vectors. We could do some preprocessing and decide to give different weighting to words based on their importance to the document in question. A common methodology used to do this is TF-IDF (term frequency — inverse document frequency). The TF-IDF weighting for a word increases with the number of times the word appears in the document but decreases based on how frequently the word appears in the entire document set. 3. Classification Algorithm and Strategy We classify the document by comparing the number of matching terms in the document vectors. In the real world numerous more complex algorithms exist for classification such as Support Vector Machines (SVMs), Naive Bayes and Decision Trees.

https://medium.com/mlrecipies/document-classification-using-machine-learning-f1dfb1171935

6.8.2 TF-IDF

Text classification is a problem where we have fixed set of classes/categories and any given text is assigned to one of these categories. In contrast, Text clustering is the task of grouping a set of unlabeled texts in such a way that texts in the same group (called a cluster) are more similar to each other than to those in other clusters.

In information retrieval or text mining, the term frequency-inverse document frequency also called tf-idf, is a well known method to evaluate how important is a word in a document. tf-idf are also a very interesting way to convert the textual representation of information into a Vector Space Model (VSM). Google has already been using TF*IDF (or TF-IDF, TFIDF, TF.IDF, Artist formerly known as Prince) as a ranking factor for your content for a long time, as the search engine seems to focus more on term frequency rather than on counting keywords. TF*IDF is an information retrieval technique that weighs a term's frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. The product of the TF and IDF scores of a term is called the TF*IDF weight of that term. The TF*IDF algorithm is used to weigh a keyword in any content and assign the importance to that keyword based on the number of times it appears in the document. More importantly, it checks how relevant the keyword is throughout the web, which is referred to as corpus.

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids shoud be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid.

Nevertheless all that was discussed has to be deployed and used somewhere. As said earlier, the serverless paradigm will be used, and so the next section tries to cover it's main benefits, as well it's challenges. Also, it's intended to demonstrate the real case scenarios.

6.9 Serverless

The scale and complexity of ML workflows makes it hard to provision and manage resources. A burden for ML practitioners that hinders both their productivity and effectiveness. Encouragingly, however, serverless computing has recently emerged as a compelling solution to address the general problem of data center resource management [52].

Serverless architectures (also referred to as "FaaS," or Function as a Service) enable organizations to build and deploy software and services without maintaining or provisioning any physical or virtual servers [53].

Serverless computing simplifies deployment of computational resources, and abstract the user of infrastructure knowledge. Problems that are frequent in users that code Machine Learning jobs.

First, users of Machine Learning are often required to manually configure numerous system-level parameters, such as number of workers/parameter servers, memory allocation, number of CPUs, physical topology, etc. Second, users need to specify numerous ML-specific parameters, such as learning rate, learning algorithms, neural network structure, that interact in non-obvious ways with the system-level parameters. Third, ML workflows are increasingly comprised of multiple stages, including pre-processing, training, and hyper parameter search, each of which has different computational requirements that ML users have to account for [52].

The problem is, Machine Learning frameworks as Scikit-learn, Tensorflow and Keras consumes huge amounts of CPU, Memory and Disk IO. High usage of any of those items simply goes in the oppose direction of what serverless computing represents.

The component that enables running serverless routines on AWS is called a Lambda.

These can be thought of as processes responding to certain events from other services or mandated by user-supplied code that is written in one of the languages AWS Lambda supports. As of now, application code for Lambda can be written in NodeJS, Java, C#, Go or Python [54].

Lambda functions, by design have very limited amount of memory and local disk. For instance, AWS lambdas can only access at most 3GB of local RAM and 512MB of local disk [52]. Those limits make unable to run Tensorflow or Spark on AWS lambdas nor, in fact, on VMs with such resource-constrained configurations.

Also Lambda functions have little available bandwidth. The largest AWS Lambda can only sustain 40MB/s of bandwidth, in contrast with > 1GB/s in the case of medium-sized VMs [52]. Lambda functions are short-lived and their launch times are highly variable – AWS lambdas can take up to 2 minutes to start.

There are design patterns that can help solving that problem, as will be shown on next sections. Splitting the code into micro-services help to account for the need of said items. Also publish/subscriber event systems in a document analysis solution make the launch time pointless.

Lambdas start at unpredictable times and can finish in the middle of training. This requires ML runtimes for lambdas to tolerate the frequent departure and arrival of workers. Lack of Fast Shared Storage. Because lambda functions cannot connect between themselves,shared storage needs to be used. To achieve good performance, this shared storage needs to be low latency, high-throughput and optimized for the type of communications in ML workloads. However,as of today there is no fast serverless storage for the cloud that provides all these properties.

To overcome the storage problem, Django will abstract the storage using django-storages, yet another library that trespass the 500mb barrier imposed by serverless vendors by using blob storages on any given Cloud, for the purposes of this system, S3.

A framework for serverless machine learning needs to meet three critical goals. First, its API needs to support a wide range of ML tasks: dataset preprocessing, training, and hyperparameter optimization. In order to ease the transition from existing ML systems, such API should be developed in a high-level language such as Python [52].

Using serverless algorithms inside serverless requires a high-level API indeed, Django will wrap those jobs and provide them through an API Gateway. It can get overwhelming to deal with all those moving parts. Deploying them can get even harded. Imagine them joining more than thirty pieces of code, with different endpoints and different start times.

A noteworthy aspect of the AWS API Gateway [55] is that it forces a separation of the API's implementation from the back-end services. To 'glue' together code and API gateway, Zappa will be used.

The next section covers the specifics of serverless, security, authentication, authorization and common patterns.

6.9.1 The core responsabilities

Applications made using serverless architectures are suitable for a wide range of services and can scale elastically as cloud workloads grow. From a software development perspective, organizations adopting serverless architectures can focus on core product functionality, and completely disregard the underlying operating system, application server or software runtime environment [53].

In serverless architectures, the serverless provider is responsible for securing the data center, network, servers, operating systems, and their configurations. However, application logic, code, data,and application-layer configurations still need to be robust and resilient to attacks. These are the responsibility of application owners. Serverless architectures introduce a new set of issues that must be considered when securing such applications:

6.9.2 Injection flaws

Injection flaws in applications are one of the most common risks to date and have been thoroughly covered in many secure coding best practice guides (as well as in the "Open Web Application Security Project (OWASP) Top 10"project). At a high level, injection flaws occur when untrusted input is passed directly to an interpreter before being executed or evaluated.

6.9.3 Broken authentication

Imagine a serverless application which exposes a set of public APIs, all of which enforce proper authentication. At the other end of the system, the application reads files from a cloudstorage service where file contents are consumed as input to specific serverless functions. If proper authentication is not applied to the cloud storage service, the system is exposing an unauthenticated rogue entry point—an element not considered during system design.

6.9.4 Insecure serverless deployment

Certain configuration parameters have critical implications for overall security postures of applications and should be given attention, and settings provided by serverless architecture vendors may not be suitable for a developer's needs. Misconfigured authentication/authorization is a widespread weakness affecting applications that usecloud-based storage. Since one of the recommended best practice designs for serverless architectures is to make functions stateless, many applications built for serverless architectures rely on cloud storage infrastructure to store and persist data between executions.

6.9.5 Over-privileged function permissions

Since serverless functions usually follow microservices concepts, many serverless applications contain dozens, hundreds or even thousands of functions. Resultantly, managing function permissions and roles quickly becomes a tedious task. In such scenarios, organizations may be forced to use a single permission model or security role for all functions—essentially granting each function full access to all other system components. When all functions share the same set of over-privileged permissions, a vulnerability in a single function can eventually escalate into a system-wide security catastrophe.

6.9.6 Inadequate monitoring and logging

Looking back at major successful cyber breaches, one key element that was always an advantage for the attackers, was the lack of real-time incident response, which was caused by failure to detect early signals of an attack.

6.9.7 Insecure third-party dependencies

Numerous white papers and surveys have addressed the prevalence of insecurethird-party packages, evidenced by a quick search in the MITRE CVE (Common Vulnerabilities andExposures) database. Furthermore, packages and modules—often used when developing serverlessfunctions—frequently contain vulnerabilities.

6.9.8 Insecure application secretes storage

As applications grow in size and complexity, there is a need to store and maintain "application secrets. API keys, Database credentials, Encryption keys, Sensitive configuration settings recurring mistakes related to application secrets storage, is to simply store these secrets in a plain text configuration file that is a part of the software project.

6.9.9 Denial of service and financial resources

In 2016, a distributed denial-of-service (DDoS) attack reached a peak of one terabit per second (1 Tbps). The attack supposedly originated from a botnet comprised of millions of infected IoT devices. While serverless architectures bring promises of automated scalability and high availability, they also come with limitations and issues which require attention, namely the cost involved with infinity scalability.

6.9.10 Serverless function execution flow manipulation

Business logic manipulation is a common problem in many types of software and serverless architectures. However, serverless applications are unique, as they often follow the microservices design paradigm and contain many discrete functions. These functions are chained together in a specific order, which implements the overall application logic. In a system where multiple functions exist - and each function may invoke another function - the order of invocation may be critical for achieving the desired logic. Moreover, the design might assume that certain functions are only invoked under specific scenarios and only by authorized invokers.

6.9.11 Improper exception handling

At the time of this writing, line-by-line debugging options for serverless-based applications are limited(and more complex) when compared to debugging capabilities for standard applications. This reality is especially true when serverless function utilizes cloud-based services not available when debugging the code locally.

6.9.12 Obsolete functions

Similar to other types of modern software applications, over time some serverless functions and related cloud resources might become obsolete and should be decommissioned. Pruning obsolete application components should be done periodically both for reducing unnecessary costs, and for reducing avoidable attack surfaces. Obsolete serverless application components may include: Deprecated serverless functions versions, Serverless functions that are not relevant anymore, Unused cloud resources (e.g. storage buckets, databases, message queues, etc.), Unnecessary serverless event source triggers, Unused users, roles or identities, Unused software dependencies

6.9.13 Cross-execution data persistency

Serverless platforms offer application developers local disk storage, environment variables and RAM memory in order to perform compute tasks in a similar fashion to any modern software stacks. In order to make serverless platforms efficient in handling new invocations and avoid cold-starts, cloud providers might reuse the execution environment (e.g. container) for subsequent function invocations. In a scenario where the serverless execution environment is reused for subsequent invocations, which may belong to different end users or session contexts, it is possible that sensitive data will be left behind and might be exposed.

6.10 Zappa

Zappa is a tool aimed to help deployment of Serverless Python Web Services. It is written on top of Boto3 and provides bindings for both the AWS Lambda and API Gateway and is capable of handling applications written in Django and Flask [56]. And although it really simplifies usage of serverless deployment, some of the code had to be designed using boto3, namely the bindings to SQS services.

With Zappa, it's possible to omit predefined API Gateway bindings from the uploading phase of cloud environment and defer them to be established when the first deployment of application logic is done.

The semantics of Zappa's behaviour can be divided into two phases. First the application is packaged into an archive that contains all the required code for the application to run. During this process Zappa will replace any local dependencies with AWS compatible versions and skip unnecessary files.

Second, the package is then uploaded to S3, depending on it's size, a small handler will be deployed to Lambda which will be responsible for retrieving the bigger package from S3 later on, upon usage.

This is particularly useful as libraries used on the Fraud detection system tends to be bigger than the maximum allowed size for AWS Lambdas. Bear in mind that dealing with handlers to load an application on-the-fly brings a hidden cost attached of latency. So, to use it wisely, the only tasks where this strategy was used were on the asynchronous jobs.

6.11 Microservices

Microservices architecture is increasingly being used to develop application systems since its smaller codebase facilitates faster code development, testing, and deployment as well as optimization of the platform based on the type of microservice, support for independent development teams, and the ability to scale each component independently [57].

In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API [58].

For instance, at the beginning of the process the client needs to upload a photo to the database and create a unique hash-key to be used both by the client application and the S3 buckets. This application is a microservice, it's purpose is only to allow the upload of a photo to a S3 bucket.

Another example is if the upload succeeds another microservice will read it's content and create new tasks, for a fleet of other microservices. The purpose of this service is just this, create new tasks upon a successful upload.

This design pattern decouples the logic and follows the vision of serverless applications which are in it's core, small chunks of code.

7 Results

All that was promised and intended for this particular work was successfully done. Although the code had to be rewritten to fit serverless scenarios at least two times, which led to double code base to maintain.

With the discussed method, see Figure 7, picture the following example where a supposed original image enters the pipeline and first, the algorithm rotates it to the optimal angle, it is visible the blue line indication that the algorithm used as base to rotate the whole document.

Also, the image has white disproportions along the horizontal and vertical lines, so the document is cropped. The next image shows signature removal, which is nothing more than the discovery of ink on paper. The last image, although looking entirely black, reveals what our eyes already know, that the document is doctored. For that, it uses Error Level Analysis.

The score given for ELA was 22, whilst the number of rainbown pixels was 1397.

To explain ELA levels, take for instance the next picture (Figure 7), entirely original, no modifications whatsoever.

The first, is the original, whilst the second is the output for the ELA pipeline mentioned before. Notice that this time, there are no huge amounts of rainbow pixels, 32 only, and the score given is 7.

That indicates first, no modification in the image, only that it was probably re-saved.

The same image, if altered, as seen on Figure 7 will give us a different output:

Three spots were purposely changed in the image, the hair was blurred, as well as different spots on the image, and the chip was copied to the bottom with the copy move technique. This is what was obtained, the white dots are gone, rainbown came present in the modification areas and the ELA scores rose from 7 in the previous unmodified to 26 in the modified. Whilst the rainbow pixel quantity rose from 7 in the original to 61.

Notice the correlation between those two points, as they are one of the many indicators

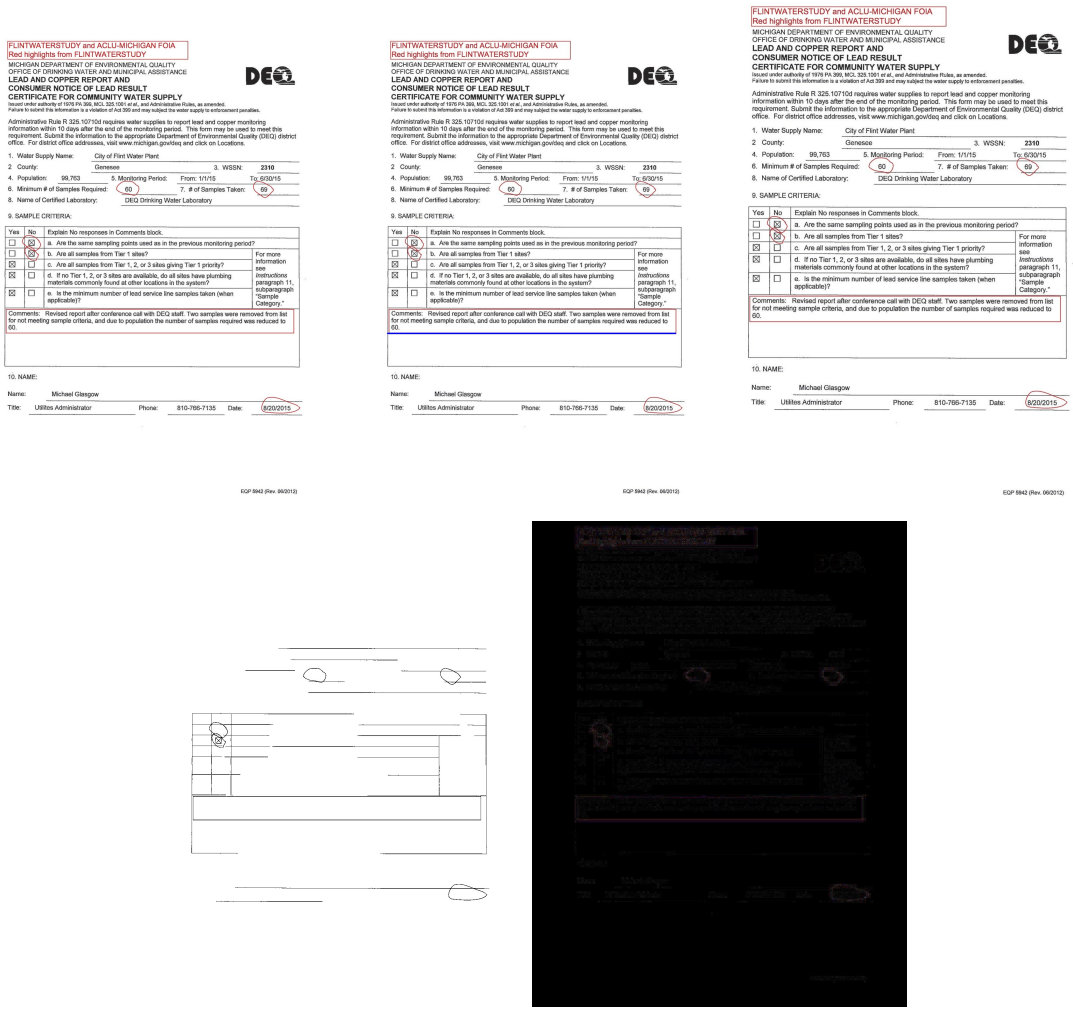


Figure 9: Input and output of a single document analysis

Figure 10: Input and output of a original, unaltered image

Figure 11: Input and output of a doctored image

that the image was in fact doctored.

Another topic covered by application is the classification of the document in it's own subclass. Although relatively simple, the process can be described as:

- The document has it's characters removed with OCR using Textract, which gives the percentage of confidence for every word extracted.
- Once the OCR is retrieved, all words are just a part of a unordered array, with no semantic correlation.
- Using the library scikit-learn, and it's module TfidfVectorizer, the importance of the words are given to the appropriate context.
- The words with higher classification are then saved, as they will be used to determine the Document class. Using the pre built data.

Although used, the retrieval of EXIF metadata is inconclusive, in simple words, it can't be used, it's a broken standard and because of it's ephemeral nature everyone could simply change its metadata.

There is a pipeline which specifically looks for copy move spots all over the document. More about the analysis that were made can be found later on, more precisely on the Attachments section.

The speed of the application is worth-mentioning. Usually it is expected a higher latency when dealing with serverless applications. But, because the deployment was using the 'keep-

warm' feature, the portal could be accessed using 200ms latency usually. Of course that those numbers turn it impossible to use in particular industries, but for the purpose of this work, it was just more than enough. For instance, latency while developing locally was around 50ms to reach portal.

Because of it decoupled nature, the only speed that actually matters for the usage of the app is while accessing the portal. All the other jobs are asynchronous and thus, retrieve info from the queue on any given time.

It is worth mentioning that although Celery couldn't be used in serverless, a boto3 implementation had to be used to integrate the app with Amazon SQS. Take for example the sqs.py used for the ELA detection:

```
import time
from botocore.client import Config
import boto3

config = Config(connect_timeout=15, retries={'max_attempts': 2})

sqs = boto3.client('sqs', config=config)

def receive_sqs(request):
    queue_name='detect_ela'
    rebuilt_url = get_queue_url(queue_name)

    # Receive message from SQS queue
    response = sqs.receive_message(
        QueueUrl=rebuilt_url,
        MaxNumberOfMessages=1,
        MessageAttributeNames=[
```

```
        'All'
    ],
    VisibilityTimeout=20,
    WaitTimeSeconds=20
)
message = response['Messages'][0]
receipt_handle = message['ReceiptHandle']
message_attributes = message['MessageAttributes']
msg_id = message_attributes['id']['StringValue']
msg_unique = message_attributes['unique']['StringValue']
msg_uploaded = message_attributes['upload_url']['StringValue']
detect_ela(msg_uploaded, msg_id, msg_unique)

# Delete received message from queue
sqs.delete_message(
    QueueUrl=rebuilt_url,
    ReceiptHandle=receipt_handle
)
if response['Messages']:
    return HttpResponse(response['Messages'], status=201)
else:
    return HttpResponse(status=500)
```

This was particular useful when integrated with Zappa settings for the current job:

```
{
"dev": {
    "aws_region": "eu-west-1",
    "django_settings": "p_detect_ela.settings",
```

```
        "profile_name": "default",
        "project_name": "p-detect-ela",
        "runtime": "python3.7",
        "apigateway_enabled": false,
        "s3_bucket": "portal-2-detect-ela",
        "use_precompiled_packages": true,
        "events": [
            {
                "function": "p_detect_ela.sqs.receive_sqs",
            }
        ]
    }
}
```

Notice that on events, multiple functions can be summoned. In that particular scenario, it calls the receive_sqs method, which will trigger Amazon SQS for new messages, if there are none, nothing happens and the functions dies. It there are any messages, the method detect_ela will be called, with the following arguments msg_uploaded, msg_id, msg_unique.

The same logic was applied to all pipelines, and it made possible for the correct decoupling of the application. This method makes easier to extend it to support hundreds of new algorithms, because what actually matters is what the code does with the received event. And that can be anything. It is truly a solid foundation to support newcomers to the area, creating a easily extensible playground.

8 Difficulties

The absence of publicly available identity document datasets is a serious problem which directly affects the state of research in this field. It raises the entry barrier, putting off those who don't have the resources to create their own datasets and slowing down those who haven't collected the data yet. Furthermore, it becomes impossible to evaluate and compare various identity document analysis methods to each other, since they have been tested on completely different and locked down data. [2]

Not only the dataset imposes barriers, as different methods have to be used to normalize the input documents, but, different documents have so many intrinsicalities that there is almost a group of algorithms per document class. That said, to correctly accomplish and determine fraud, the correct classification must be done. There should be a humongous amount of documents from everywhere to accomplish that, MIDV-500 dataset almost did the job.

Serverless is also a huge buzz-world right now, what it implies shall be defined still. It simplifies deployment and abstract development from infrastructure, in that sense, it's usage should be saught even more often than nowadays. The problem it imposes is the lack of good real world scenarios, low to none academic usage and costs intrinsic to the method. Not only, security is critical if you consider it thoroughly.

Dealing with serverless comes with different security risks. Serverless functions consume data from a wide range of event sources, such as HyperText Transfer Protocol (HTTP) application program interfaces (APIs), message queues, cloud storage, internet of things (IoT) device communications, and so forth [53]. Cloud providers also have problems whilst dealing with the correct flow of serverless applications, most of them have the default policy of all the traffic between serverless applications happen through the internet. This capability is undesired as generally what is wanted is to connect microservices internally, as it was inside a LAN.

This diversity increases the potential surface dramatically, especially when messages use protocols and complex message structures. Many of these messages cannot be inspected by

standard application layer protections, such as web application firewalls (WAFs).

Also the attack surface in serverless architectures can be difficult for some to understand given that such architectures are still somewhat new. Many software developers and architects have yet to gain enough experience with the security risks and appropriate security protections required to secure such applications [53].

Visualizing and monitoring serverless architectures is still more complicated than standard software environments. Also, by using frameworks to abstract the deployment process a huge amount of detail is hidden. Libraries are compiled under the hood, which may lead to even bigger security problems.

Python was absolutely great for the task, gave a huge speed to the whole process. Though it can be used for everything, with serverless and tesseract it just do not work. Generally speaking, tesseract has to be installed in the environment to be used by Python as a wrapper. In serverless you can't install it, and although cloud providers are issuing a new capability, called layers, the correct method of usage is just not clear yet. So to overcome this step, Textract was used.

The development process is also not linear. The POC here was done using containers, so it was simple to use Django, Celery, PostgreSQL, TensorFlow, Keras, Scikit-learn and so on. All that, working on a single environment it quite easy to maintain. The network is only one, all is internally connected by Docker overlay network. It works flawlessly. You can't deploy it through serverless as it is. Celery requires daemonization to do so, so had to be replaced by boto3 and zappa capabilities. Also, the communication between apps have to go through APIs which demands the code sometimes to be refactored to work on serverless.

Performing security testing for serverless architectures is more complex than testing standard applications, especially when such applications interact with remote third-party services or with backend cloud services, such as Non-Structured Query Language (NoSQL) databases, cloud storage, or stream processing services. Additionally,automated scanning tools are currently not adapted to examining serverless applications [53]. Which leads generally to the use of lock-in vendor tools to correct debug of the application.

9 Conclusion

All that was promised in the objectives section and intended for this particular work was successfully done, even though not all the possibilities discussed here were covered. This was intended as the main focus remained on general fraud detection, so specific forgeries - mechanical for instance - couldn't be covered.

Also, the absence of publicly available identity document datasets is a serious problem which directly affects the state of research in this field. It raises the entry barrier, putting off those who don't have the resources to create their own datasets and slowing down those who haven't collected the data yet. Also, because not everyone has the same data, the solutions offered can't be compared.

Technically speaking, Serverless is a huge buzz-world right now, what it implies shall be defined still. One fact is clear, the market truly needs better and faster ways of doing software, Serverless is one, it abstracts the developer from dealing with servers, networking, security and so on. Simply put, it simplifies deployment and abstract development from infrastructure, in that sense, it's usage should be sought even more often than nowadays as it speed up the process drastically. The problem it imposes is the lack of good real world scenarios, low to none academic usage and costs intrinsic to the method. Also, dealing with different cloud vendors and serverless approaches most certainly requires a usage of a third party framework which generally imposes new methods and new APIs to learn.

Security also plays a critical role if you consider it thoroughly, as the shared layer abstracts your current infrastructure and there are no guarantees of exactly what physical resources are being used and if they are leaking information, if any. Visualizing and monitoring serverless architectures is still more complicated than standard software environments and it depends hugely on the cloud vendor. Also, by using frameworks to abstract the deployment process a huge mount of detail is hidden. Libraries are compiled under the hood, which may lead to even bigger security problems.

One fact is clear, today's technology allows digital media to be altered and manipulated in ways that were simply impossible 20 years ago. Tomorrow's technology will almost certainly allow us to manipulate digital media in ways that today seem unimaginable. Simply put, the methods here described will be used to leverage the entire subsequent project. While the main objective is to proper define what is fraud and how can one prevent and outcome it, this particular project focused in a practical, extensible, cost-effective manner to combat an increasing threat.

The tools herein referenced can easily be scaled to reach any size of company, team or logic - from small markets to multinational companies, it's use can be found in any sort of market sector. Although a tremendous work lies ahead, there is not a single doubt that the concept explained can and will be easily improved in the subsequent work.

As the field of forensics continues to expand, new methodologies may appear, and given that any piece of code can be extended to receive an event and process an image, this work will still be valid.

So to achieve that, the future of this application is to become open source, and with it the community will be able to develop over it. It is not hard to add image processing steps after all, basically what is needed is just a any-code script that reads an image based on a dynamic path, and over it, can perform it's algorithm over a correct SQS/Task binding.

As it is, any sort of binding can be used. Any framework, any cloud vendor or on-premises equipment, and basically any language. So to speak, it imposes almost no entry barriers to any person that seek to develop over it.

In general terms, a lot has to be done. After the correct classification of document class, the extraction of main features, detection of the main fraud methods, one can always out-perform the application adding any desired method. It is not hard to picture a scenario where a company wants to add this very application to verify all signed contracts, digitally or not to seek for clause changes.

Even better, the system can be easily extended to verify frames into digital streams. Making it useful to a whole new spectrum of companies worldwide.

Digital forensics needs a faster way of dealing with endless doctored images, this project is one.

10 Attachments

The section will show two documents that were uploaded to the application, for brevity purposes, only two will be shown here.

10.1 Analysis 52

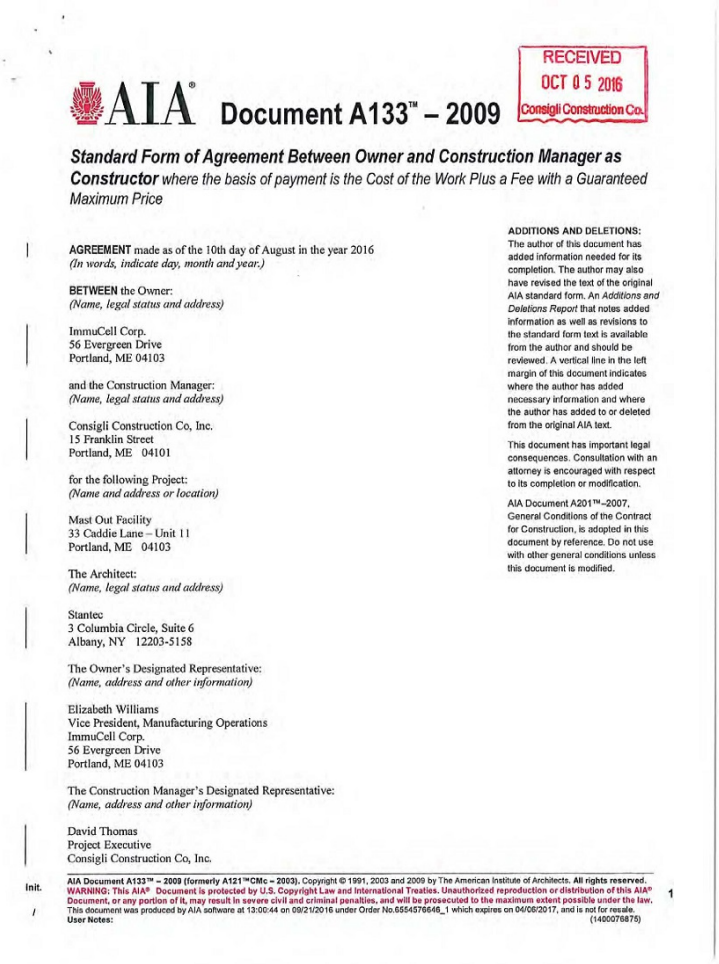
The next pages retract the output as is from the application for the document number 52. No adjustments were made to the image. A border was added to illustrate how the output would be.

Document Analysis 52

Wed Jun 24 13:08:40 2020
Created by cchostak on 2020-05-31 16:25:03.866318+00:00
Description: 133
Unique identifier: XBPemmJmTNqwXg-digjbYwhmsvGEgJRFGM6CVd4mc1yg
Text extracted from image: "b'RECEIVED OCT 0 15 2016 AIA Document A133 - 2009 Consigl Construction Co. Standard Form of Agreement Between Owner and Construction Manager as Constructor where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price ADDITIONS AND DELETIONS: The author of this document has AGREEMENT made as of the 10th day of August in the year 2016 added information needed for its (In words, indicate day, month and year.) completion. The author may also have revised the text of the original BETWEEN the Owner: AIA standard form. An Additions and (Name, legal status and address) Deletions Report that notes added information as well as revisions to ImmuCel Corp. the standard form text is available 56 Evergreen Drive from the author and should be Portland, ME 04103 reviewed. A vertical line in the left margin of this document indicates and the Construction Manager: where the author has added (Name, legal status and address) necessary information and where the author has added to or deleted Consigli Construction Co, Inc. from the original AIA text. 15 Franklin Street This document has important legal Portland, ME 04101 consequences. Consultation with an attorney is encouraged with respect for the following Project: to its completion or modification. Name and address or location) AIA Document A4201TM-2007 Mast Out Facility General Conditions of the Contract 33 Caddie Lane - Unit I1 for Construction. is adopted in this Portland, ME 04103 document by reference. Do not use with other general conditions unless this document is modified The Architect: (Name, legal status and address) Stantec 3 Columbia Circle, Suite 6 Albany, NY 12203-5158 The Owner\'s Designated Representative: (Name, address and other information) Elizabeth Williams Vice President, Manufacturing Operations ImmuCell Corp. 56 Evergreen Drive Portland, ME 04103 The Construction Manager\'s Designated Representative: (Name, address and other information) David Thomas Project Executive Consigli Construction Co, Inc AIA Document A133T - 2009 (formerly 121"MCMc 2003). Copyright 1991 2003 and 2009 by The American Institute of Architects. Al rights reserved Init. WARNING: This AIA Document is protected by U.S. Copyright Law and International Treaties. Unauthorized reproduction or distribution of this AIAR 1 Document or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law, / This document was produced by AIA software at 13:00:44 on 09/21/2016 under Order No.6554576646_ which expires on 04/06/2017, and is not for resale. User Notes: (1400076875) RECEIVED OCT 0 15 2016 AIA Document A133 - 2009 Consigl Construction Co. Standard Form of Agreement Between Owner and Construction Manager as Constructor where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price ADDITIONS AND DELETIONS: The author of this document has AGREEMENT made as of the 10th day of August in the year 2016 added information needed for its (In words, indicate day, month and year.) completion. The author may also have revised the text of the original BETWEEN the Owner: AIA standard form. An Additions and (Name, legal status and address) Deletions Report that notes added information as well as revisions to ImmuCel Corp. the standard form text is available 56 Evergreen Drive from the author and should be Portland, ME 04103 reviewed. A vertical line in the left margin of this document indicates and the Construction Manager: where the author has added (Name, legal status and address) necessary information and where the author has added to or deleted Consigli Construction Co, Inc. from the original AIA text. 15 Franklin Street This document has important legal Portland, ME 04101 consequences. Consultation with an attorney is encouraged with respect for the following Project: to its completion or modification. Name and address

or location) AIA Document A4201TM-2007 Mast Out Facility General Conditions of the Contract 33 Caddie Lane - Unit I1 for Construction. is adopted in this Portland, ME 04103 document by reference. Do not use with other general conditions unless this document is modified The Architect: (Name, legal status and address) Stantec 3 Columbia Circle, Suite 6 Albany, NY 12203-5158 The Owner\'s Designated Representative: (Name, address and other information) Elizabeth Williams Vice President, Manufacturing Operations ImmuCell Corp. 56 Evergreen Drive Portland, ME 04103 The Construction Manager\'s Designated Representative: (Name, address and other information) David Thomas Project Executive Consigli Construction Co, Inc AIA Document A133T - 2009 (formerly 121"MCMc 2003).

Original image:



AIA® Document A133™ – 2009

RECEIVED
OCT 0 5 2016
Consigli Construction Co.

Standard Form of Agreement Between Owner and Construction Manager as Constructor *where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price*

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Out Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

David Thomas
Project Executive
Consigli Construction Co, Inc.

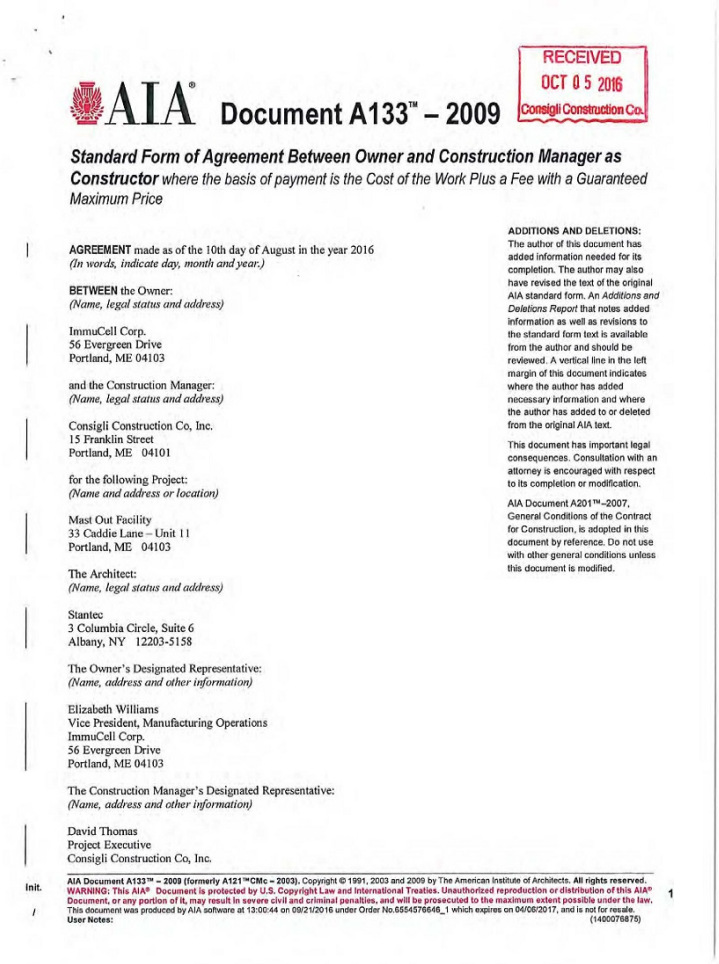
ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An *Additions and Deletions Report* that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adopted in this document by reference. Do not use with other general conditions unless this document is modified.

Init.
/

Converted image:



AIA® Document A133™ – 2009

RECEIVED
OCT 0 5 2016
Consigli Construction Co.

Standard Form of Agreement Between Owner and Construction Manager as Constructor *where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price*

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Out Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

David Thomas
Project Executive
Consigli Construction Co, Inc.

ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An *Additions and Deletions Report* that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adopted in this document by reference. Do not use with other general conditions unless this document is modified.

Init.
/

HUE image levels:

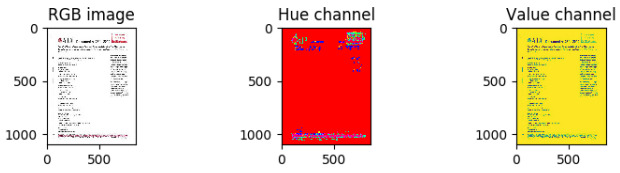
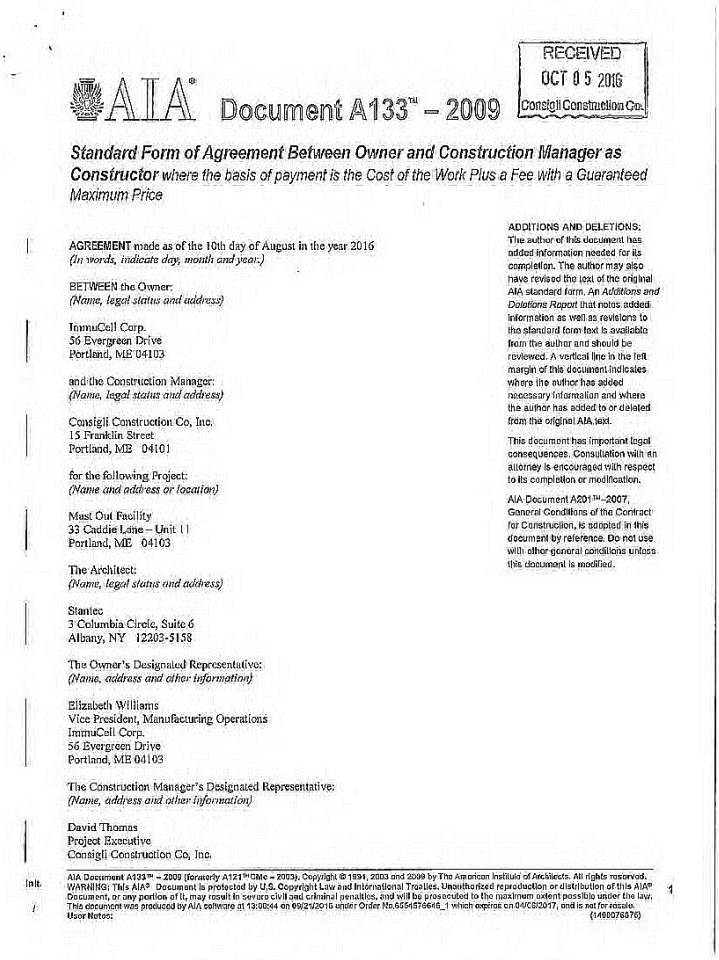


Image Contours:



RECEIVED
OCT 05 2016
Consigli Construction Co.

AIA® Document A133™ – 2009

Standard Form of Agreement Between Owner and Construction Manager as Constructor *where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price*

ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An *Additions and Deletions Report* that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adopted in this document by reference. Do not use with other general conditions unless this document is modified.

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Out Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

David Thomas
Project Executive
Consigli Construction Co, Inc.

Image Edges:

RECEIVED
OCT 05 2016
Consigli Construction Co.

AIA Document A133™ – 2009

Standard Form of Agreement Between Owner and Construction Manager as Constructor where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Out Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

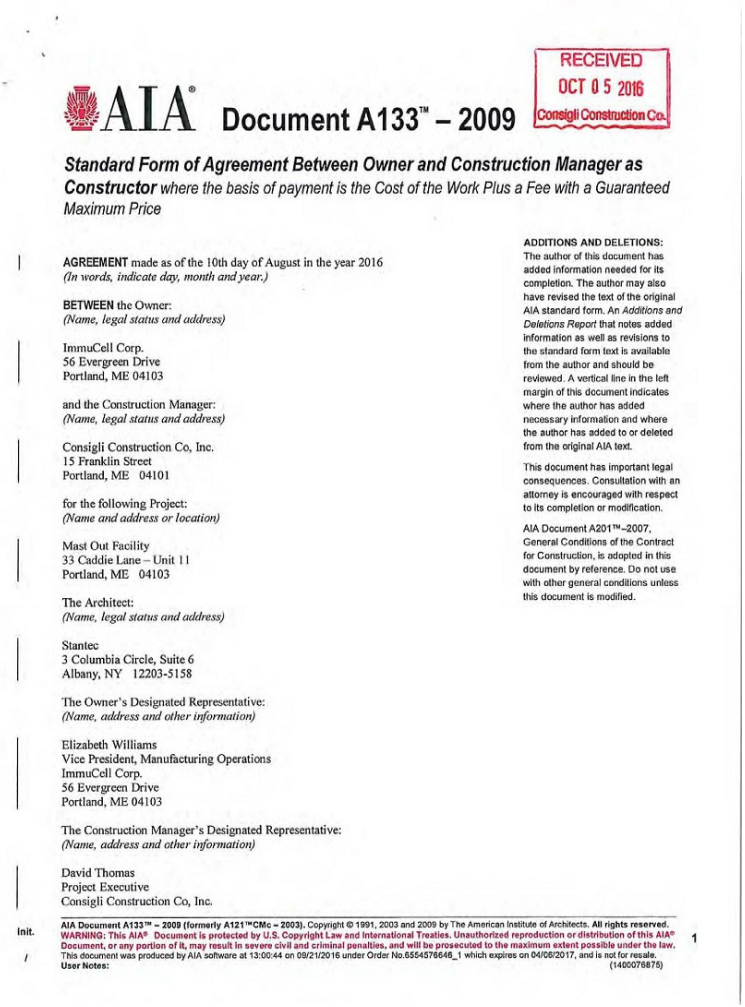
David Thomas
Project Executive
Consigli Construction Co, Inc.

ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An *Additions and Deletions Report* that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adopted in this document by reference. Do not use with other general conditions unless this document is modified.

Image Cropped:



RECEIVED
OCT 05 2016
Consigli Construction Co.

AIA® Document A133™ – 2009

Standard Form of Agreement Between Owner and Construction Manager as Constructor *where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price*

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Out Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

David Thomas
Project Executive
Consigli Construction Co, Inc.

ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An *Additions and Deletions Report* that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adopted in this document by reference. Do not use with other general conditions unless this document is modified.

Init.
/

Image Rotated:

AIA® Document A133™ – 2009

RECEIVED
OCT 05 2016
Consigli Construction Co.

Standard Form of Agreement Between Owner and Construction Manager as Constructor *where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price*

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Out Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

David Thomas
Project Executive
Consigli Construction Co, Inc.

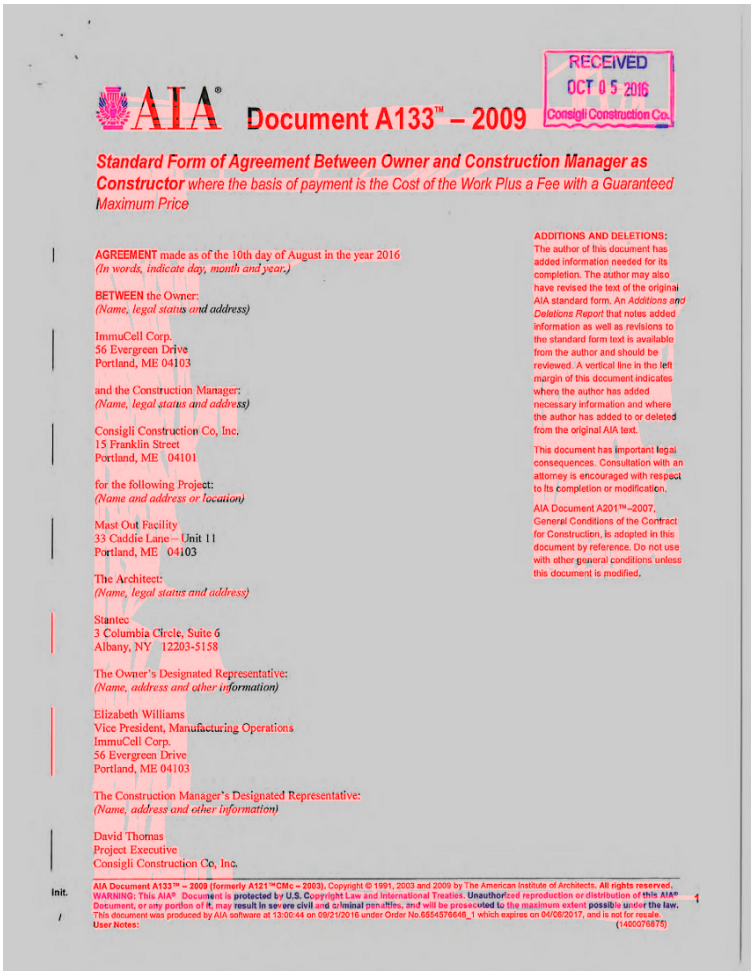
ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An *Additions and Deletions Report* that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adopted in this document by reference. Do not use with other general conditions unless this document is modified.

Init.
/

Image Line Detection:



AIA® Document A133™ – 2009

RECEIVED
OCT 05 2016
Consigli Construction Co.

Standard Form of Agreement Between Owner and Construction Manager as Constructor *where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price*

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Out Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

David Thomas
Project Executive
Consigli Construction Co, Inc.

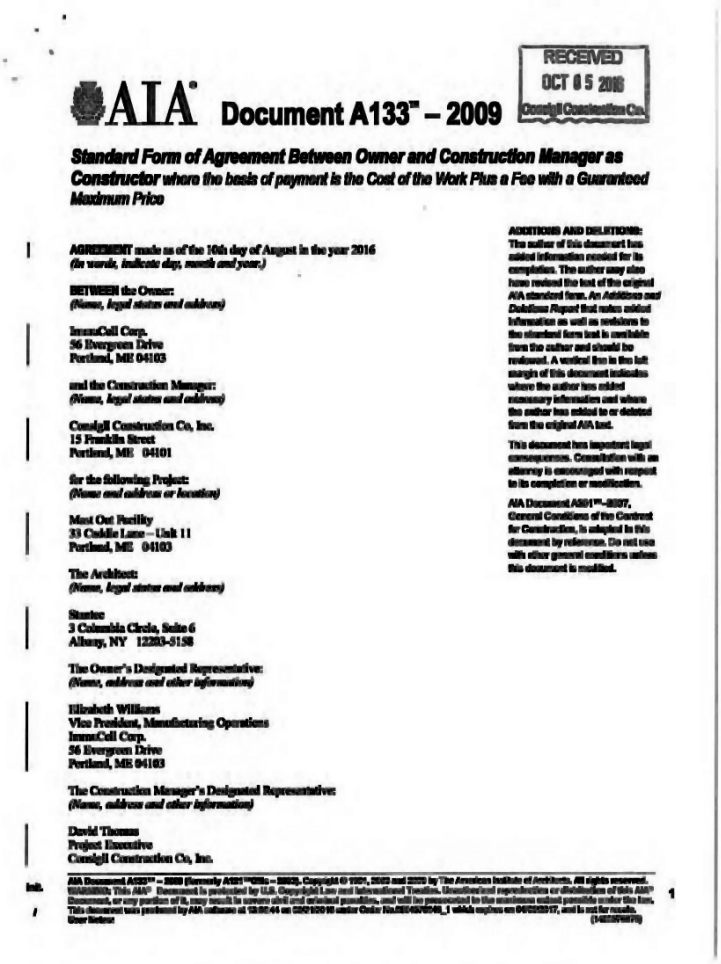
ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An *Additions and Deletions Report* that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adopted in this document by reference. Do not use with other general conditions unless this document is modified.

Init.
/

Image Minima:



RECEIVED
OCT 05 2016
Consigli Construction Co.

AIA Document A133™ – 2009

Standard Form of Agreement Between Owner and Construction Manager as Constructor where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Cot Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

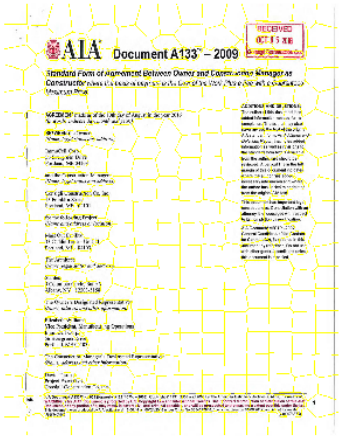
David Thomas
Project Executive
Consigli Construction Co, Inc.

ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An Additions and Deletions Report that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adapted in this document by reference. Do not use with other general conditions unless this document is modified.

Image Superpixels:

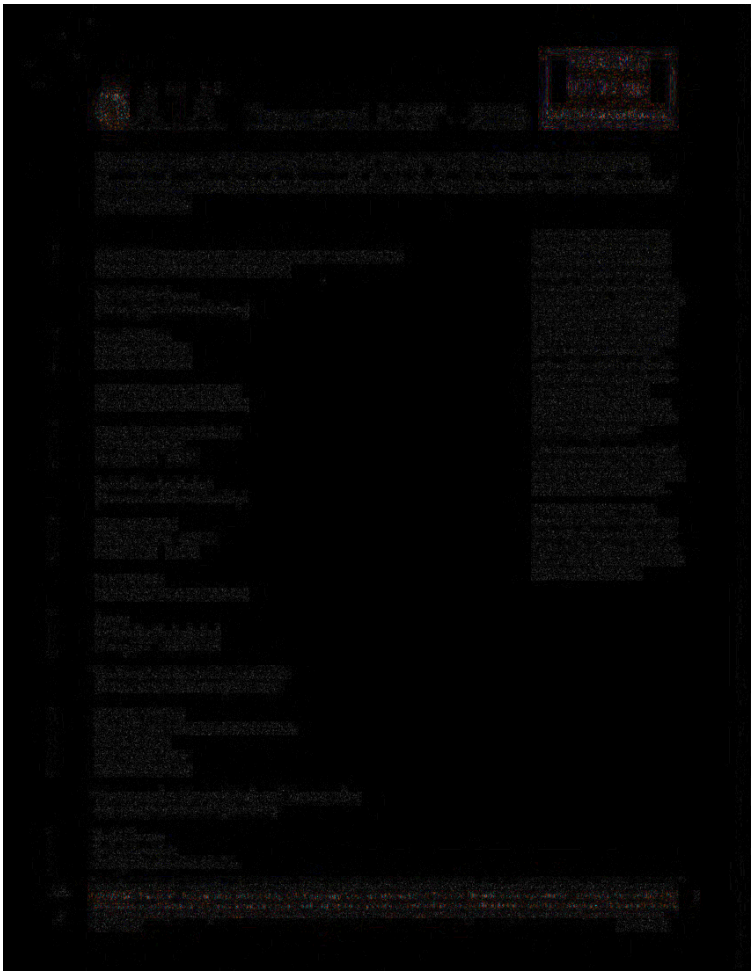


AIA Document A133™ – 2009

Standard Form of Agreement Between Owner and Construction Manager as Constructor where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price

RECEIVED
OCT 15 XX

ELA image levels:



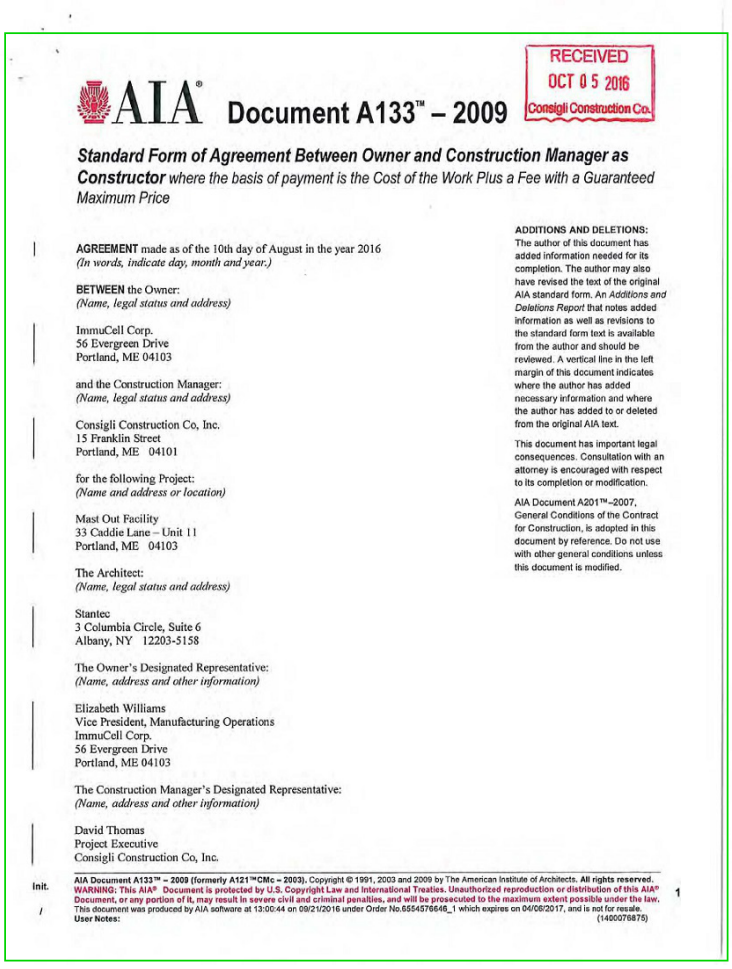
Labels infered from image:

Label Description: Page
Label Accuracy: 99.91699981689453

Label Description: Text
Label Accuracy: 99.91699981689453

Label Description: Label
Label Accuracy: 86.34650421142578

Label Description: Menu
Label Accuracy: 72.6274642944336



AIA® Document A133™ – 2009

RECEIVED
OCT 0 5 2016
Consigli Construction Co.

Standard Form of Agreement Between Owner and Construction Manager as Constructor *where the basis of payment is the Cost of the Work Plus a Fee with a Guaranteed Maximum Price*

AGREEMENT made as of the 10th day of August in the year 2016
(In words, indicate day, month and year.)

BETWEEN the Owner:
(Name, legal status and address)

ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

and the Construction Manager:
(Name, legal status and address)

Consigli Construction Co, Inc.
15 Franklin Street
Portland, ME 04101

for the following Project:
(Name and address or location)

Mast Out Facility
33 Caddie Lane – Unit 11
Portland, ME 04103

The Architect:
(Name, legal status and address)

Stantec
3 Columbia Circle, Suite 6
Albany, NY 12203-5158

The Owner's Designated Representative:
(Name, address and other information)

Elizabeth Williams
Vice President, Manufacturing Operations
ImmuCell Corp.
56 Evergreen Drive
Portland, ME 04103

The Construction Manager's Designated Representative:
(Name, address and other information)

David Thomas
Project Executive
Consigli Construction Co, Inc.

ADDITIONS AND DELETIONS:
The author of this document has added information needed for its completion. The author may also have revised the text of the original AIA standard form. An *Additions and Deletions Report* that notes added information as well as revisions to the standard form text is available from the author and should be reviewed. A vertical line in the left margin of this document indicates where the author has added necessary information and where the author has added to or deleted from the original AIA text.

This document has important legal consequences. Consultation with an attorney is encouraged with respect to its completion or modification.

AIA Document A201™–2007, General Conditions of the Contract for Construction, is adopted in this document by reference. Do not use with other general conditions unless this document is modified.

Init.
/

Label Description: Newspaper
Label Accuracy: 66.894775390625

Label Description: Paper
Label Accuracy: 60.948204040527344

Label Description: Number
Label Accuracy: 55.9510383605957

Label Description: Symbol
Label Accuracy: 55.9510383605957

Label Description: Word
Label Accuracy: 54.05191421508789

Label Description: Advertisement
Label Accuracy: 53.47906494140625

Label Description: Poster
Label Accuracy: 53.47906494140625

Digest Image Width: 850
Digest Image Height: 1100
Digest Image Perceptual Hash: 10666780694329732224
Digest Image MD5: cf71ff074fd3e27209d04c0fb8899a73
Digest Image SHA1: d9bc39fb0c641061574075a5a0f263ab0a48ea78
Digest Image SHA256:
50900cd21dd764ac6958ccba69bd8b02b33ff980be8b9f751b87927e5846d28b
Digest Image Seconds to Complete: -2.703378200531006

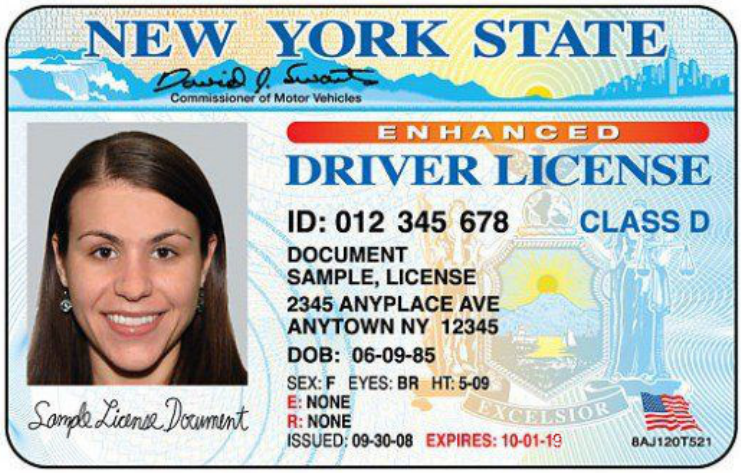
10.2 Analysis 58

The next pages retract the output as is from the application for the document 58. No adjustments were made to the image. A border was added to illustrate how the output would be.

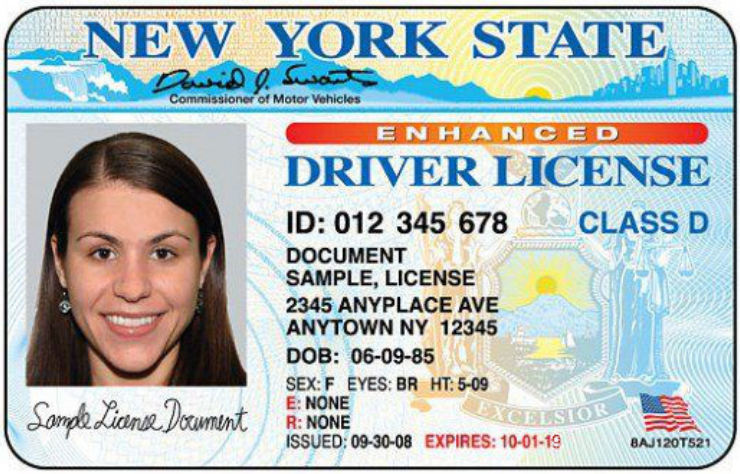
Document Analysis 58

Sun Jul 5 15:10:53 2020
Created by cchostak on 2020-07-05 15:10:09.755139+00:00
Description: 52800
Unique identifier: twl2dBggTR--uAMO9AdIIwm8vCKES_Qk-GZ2YFLlJJIg
Text extracted from image: "b'EW YORK STATE 240 9 So Commissioner of Motor Vehicles ENHAn N DRIVER LICENSE ID: 012 345 678 CLASS D DOCUMENT SAMPLE, LICENSE 2345 ANYPLACE AVE ANYTOWN NY 12345 DOB: 06-09-85 SEX: F EYES BR HT 5-09 SompbeLiane Doumint E NONE R: NONE ISSUED: 09-30-08 EXPIRES: 10-01-19 8A.J120T521 EW YORK STATE 240 9 So Commissioner of Motor Vehicles ENHAn N DRIVER LICENSE ID: 012 345 678 CLASS D DOCUMENT SAMPLE, LICENSE 2345 ANYPLACE AVE ANYTOWN NY 12345 DOB: 06-09-85 SEX: F EYES BR HT 5-09 SompbeLiane Doumint E NONE R: NONE ISSUED: 09-30-08 EXPIRES: 10-01-19 8A.J120T521'"

Original image:



Converted image:



HUE image levels:

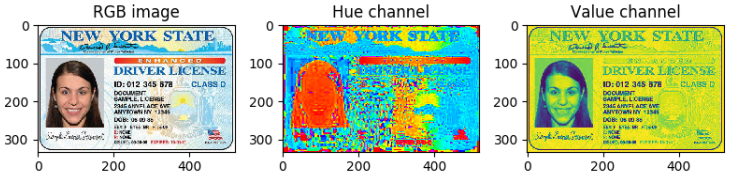


Image Contours:

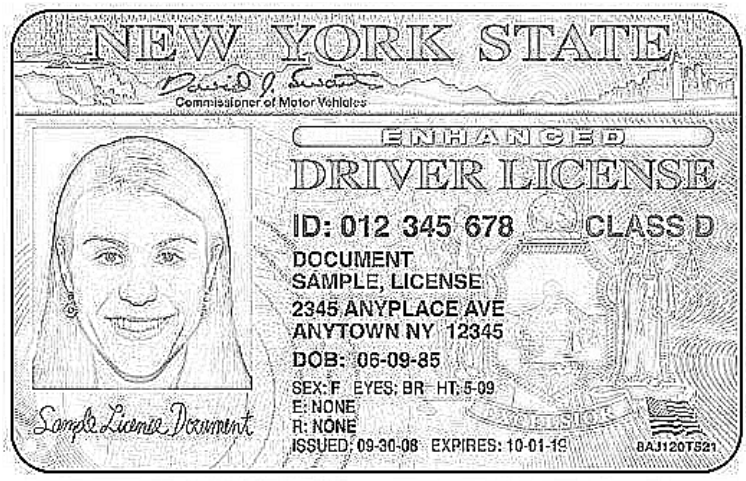


Image Edges:

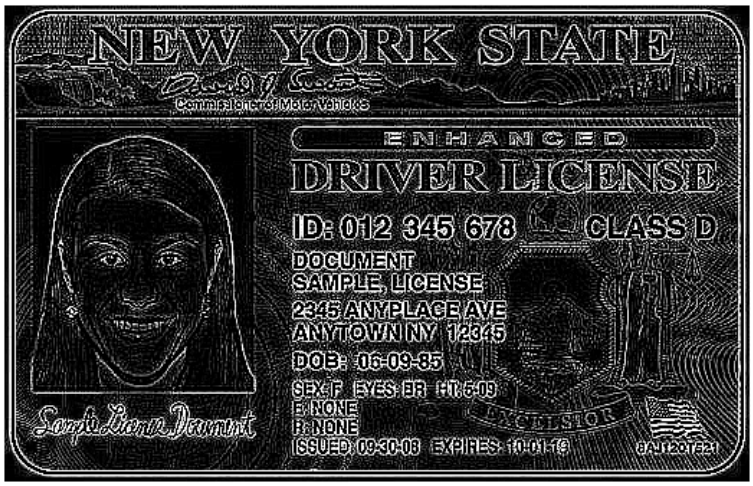


Image Cropped:

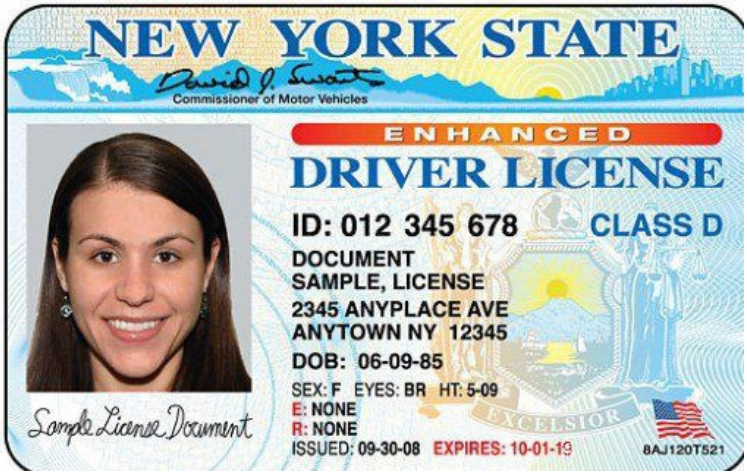


Image Rotated:

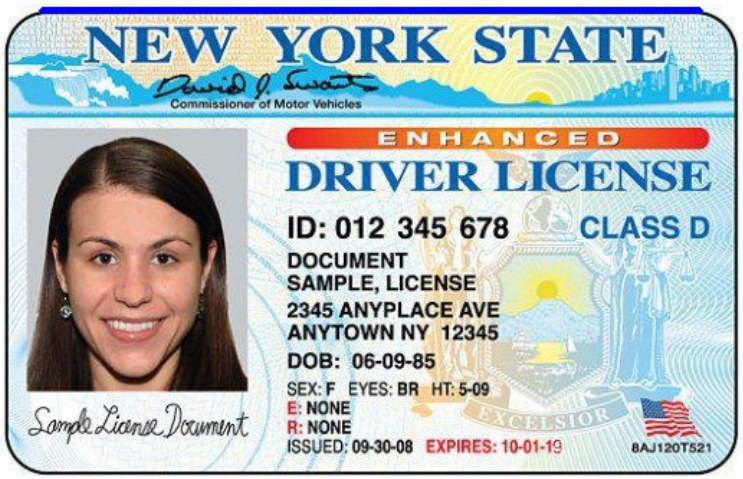


Image Line Detection:

Image Minima:

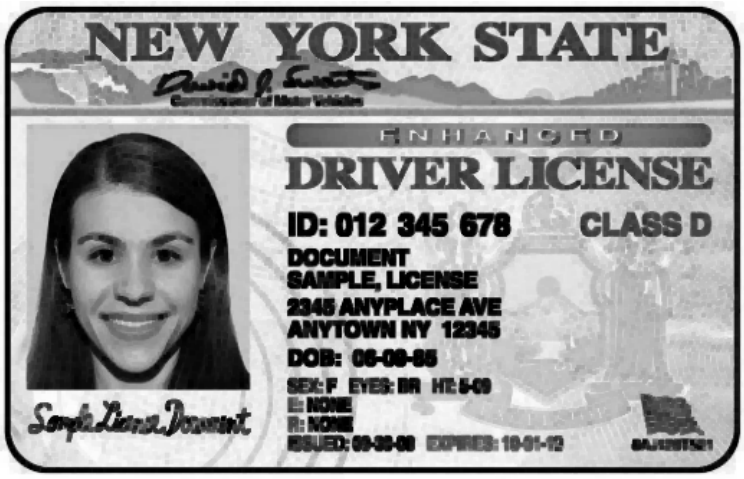


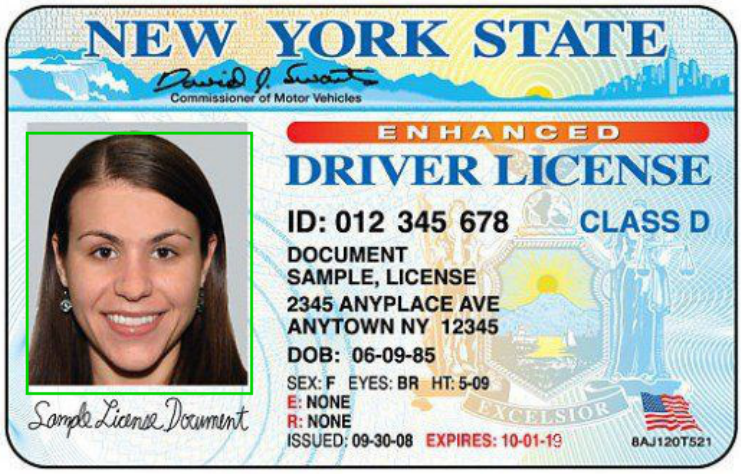
Image Superpixels:

ELA image levels:

Labels infered from image:

Label Description: Text
Label Accuracy: 99.97374725341797

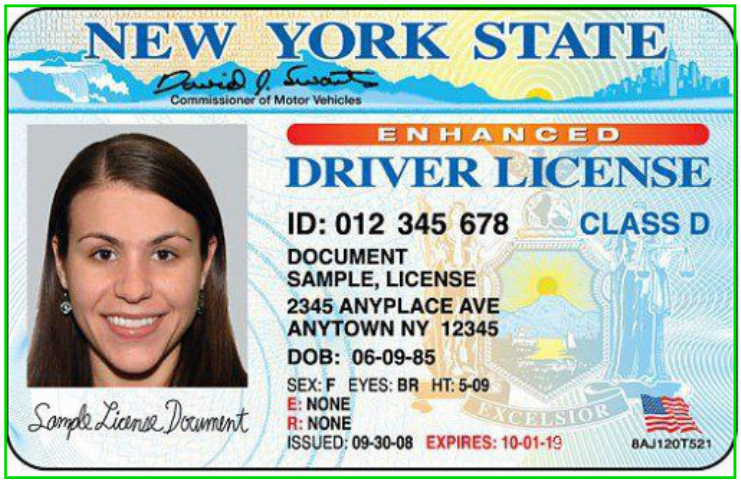
Label Description: Person
Label Accuracy: 99.5541763305664



Label Description: Human
Label Accuracy: 99.5541763305664

Label Description: License
Label Accuracy: 96.62488555908203

Label Description: Driving License
Label Accuracy: 96.62488555908203



Label Description: Document
Label Accuracy: 96.62488555908203

Label Description: Id Cards
Label Accuracy: 77.19425201416016

Digest Image Width: 520
Digest Image Height: 333
Digest Image Perceptual Hash: 5292527319121307062
Digest Image MD5: 5e134ae1f909222dabaa2537b6686024
Digest Image SHA1: b56e52515a242b55ecc7733f9a79c62b78b60054
Digest Image SHA256:
4a1f86724728c09407bcb45fce9913ee0455bb2630e233544663457269f5b992
Digest Image Seconds to Complete: -1.7191834449768066

References

[1] G. Stevenson, "Computer fraud: Detection and prevention," *Computer Fraud Security*, vol. 2000, no. 11, pp. 13 – 15, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1361372300110188

[2] V. V. Arlazarov, K. Bulatov, T. Chernov, and V. L. Arlazarov, "MIDV-500: A Dataset for Identity Documents Analysis and Recognition on Mobile Devices in Video Stream," 2018. [Online]. Available: http://arxiv.org/abs/1807.05786

[3] Statista, "U.s. payment card fraud losses by type 2018 | statistic," *Statista*, 2018. [Online]. Available: https://www.statista.com/statistics/419628/payment-card-fraud-losses-usa-by-type/

[4] L. Delamaire, H. Abdou, J. Pointon *et al.*, "Credit card fraud and detection techniques: a review," *Banks and Bank systems*, vol. 4, no. 2, pp. 57–68, 2009.

[5] F. I. Corporation, "Evolution of card fraud in europe 2018," *Fair Isaac Corporation*, 2018. [Online]. Available: https://www.fico.com/europeanfraud/

[6] J. Fridrich, "Methods for detecting changes in digital images," in *IEEE Workshop on Intelligent Signal Processing and Communication Systems, Melbourne, Australia*, 1998.

[7] Z. Zhou, C.-N. Yang, B. Chen, X. Sun, Q. Liu, and J. QM, "Effective and efficient image copy detection with resistance to arbitrary rotation," *IEICE Transactions on information and systems*, vol. 99, no. 6, pp. 1531–1540, 2016.

[8] C. Artaud, A. Doucet, J.-M. Ogier, and V. Poulain D'andecy, "Receipt Dataset for Fraud Detection," Tech. Rep.

[9] M. A. A. A. Rafael Vieira, Catarina Silva, "Pattern recognition in images of counterfeited documents rafael."

[10] H. Farid, "Image forgery detection," *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.461.798&rep=rep1&type=pdf*, 2009.

[11] H. H. Harralson, "Forensic document examination of electronically captured signatures," *Digital Evidence and Electronic Signature Law Review*, vol. 9, 2014.

[12] H. Farid, "A Survey of Image Forgery Detection," *IEEE Signal Processing Magazine*, 2009.

[13] A. Ghosh, D. Zou, M. Singh, and V. Analytics, "Detection and Localization of Image and Document Forgery: Survey and Benchmarking," Tech. Rep.

[14] J. van Beusekom, F. Shafait, and T. M. Breuel, "Text-line examination for document forgery detection," *International Journal on Document Analysis and Recognition*, vol. 16, no. 2, pp. 189–207, 2013.

[15] N. Razumnuy, A. Kozharinov, V. Arlazarov, D. Nikolaev, and T. Chernov, "Image quality assessment for video stream recognition systems," 04 2018, p. 39.

[16] N. Krawetz, "Digital Image Analysis and Forensic," pp. 1–43, 2008.

[17] W. Li, Y. Yuan, and N. Yu, "DETECTING COPY-PASTE FORGERY OF JPEG IMAGE MOE-Microsoft Key Laboratory of Multimedia Computing and Communication , School of Engineering and Applied Science , Aston University ,," *Science And Technology*.

[18] A. Stojkovic, I. Shopovska, H. Luong, J. Aelterman, L. Jovanov, and W. Philips, "The effect of the color filter array layout choice on state-of-the-art demosaicing," *Sensors*, vol. 19, p. 3215, 07 2019.

[19] I. Yerushalmy and H. Hel-Or, "Digital image forgery detection based on lens and sensor aberration," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 71–91, 2011.

[20] M. Gariup and J. Piskorski, "The challenge of detecting false documents at the border: Exploring the performance of humans, machines and their interaction," *International Journal of Critical Infrastructure Protection*, 2019.

[21] S. Jerome Gideon, A. Kandulna, A. A. Kujur, A. Diana, and K. Raimond, "Handwritten signature forgery detection using convolutional neural networks," *Procedia Computer Science*, vol. 143, pp. 978–987, 2018.

[22] H. H. Harralson, "Forensic document examination of electronically captured signatures," *Digital Evidence and Electronic Signature Law Review*, vol. 9, pp. 67–73, 2012.

[23] E. Sorio, A. Bartoli, G. Davanzo, and E. Medvet, "A Domain Knowledge-based Approach for Automatic Correction of Printed Invoices," Tech. Rep., 2012. [Online]. Available: http://bartoli.inginf.units.it

[24] C. A. Martins, M. C. Monard, E. T. Matsubara, C. A. Martins, and M. C. Monard, "PreTexT : uma ferramenta para pr ."

[25] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries," *Department of Computer Science, Rutgers University, 23515 BPO Way, Piscataway, NJ, 08855*, vol. 42, no. 4, pp. 40–51, 2003. [Online]. Available: https://www.researchgate.net/profile/Farshad_Madani/post/In_information_retrieval_tf-idf_calculation_why_we_dont_divide_tf_by_the_length_of_the_related_document/attachment/59d6446679197b807799fae0/AS%3A448525403201536%401483948197307/download/Using+TF-IDF

[26] A. Bartoli, G. Davanzo, E. Medvet, and E. Sorio, "Improving features extraction for supervised invoice classification," *Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Applications, AIA 2010*, pp. 401–405, 2010.

[27] CSA, "12 most critical risks for serverless applications," *https://www.puresec.io/hubfs/The-12-Most-Critical-Risks-for-Serverless-Applications.pdf*, 2019.

[28] T. Technologies, "Serverless architectures: Everything you need to know," *https://www.thorntech.com/2017/03/serverless-architectures-everything-need-know/*, 2017.

[29] G. Kawasaki, "El arte de empezar 2.0," 2015.

[30] S. Hendrickson, S. Sturdevant, T. Harter, V. Venkataramani, A. C. Arpaci-Dusseau, and R. H. Arpaci-Dusseau, "Serverless computation with openlambda," *Elastic*, vol. 60, p. 80, 2016.

[31] S. Overflow, "Stack overflow developer survey 2018," *Stack Overflow*, 2018. [Online]. Available: https://insights.stackoverflow.com/survey/2018/#technology-most-loved-dreaded-and-wanted-frameworks-libraries-and-tools

[32] S. Hansen, "Advantages and disadvantages of django," *Stevenn Hansen*, 2018. [Online]. Available: https://hackernoon.com/advantages-and-disadvantages-of-django-499b1e20a2c5?gi=6ba10294c619

[33] Kalpit, "When to use django (and when not to)," *Kalpit*, 2018. [Online]. Available: https://medium.com/crowdbotics/when-to-use-django-and-when-not-to-9f62f55f693b

[34] B. Agrawal, "How we broke up our monolithic django service into microservices," *Barkha Agrawal*, 2018. [Online]. Available: https://medium.com/greedygame-media/how-we-broke-up-our-monolithic-django-service-into-microservices-8ad6ff4db9d4

[35] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, "Scikit-learn," *GetMobile: Mobile Computing and Communications*, vol. 19, no. 1, pp. 29–33, 2015.

[36] M. Solutions, "Advantages and disadvantages of python programming language," *medium.com*, 2017. [Online]. Available: https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121

[37] P. Physics, "Image processing with python and scipy," *prancer.physics.louisville.edu*, 2018. [Online]. Available: http://prancer.physics.louisville.edu/astrowiki/index.php/Image_processing_with_Python_and_SciPy

[38] M. Byrne, "Hack this: Extract image metadata using python," *vice.com*, 2016. [Online]. Available: https://www.vice.com/en_us/article/aekn58/hack-this-extra-image-metadata-using-python

[39] NumPy, "Numpy," *NumPy*, 2018. [Online]. Available: https://numpy.org/

[40] D. Maclaurin, D. Duvenaud, and R. P. Adams, "Autograd: Effortless Gradients in Numpy," *ICML '15 AutoML Workshop*, p. 3, 2015. [Online]. Available: http://www.scipy.org/.{\%}0Ahttps://github.com/HIPS/autograd{\%}0Ahttp://www.cs.toronto.edu/{~}rgrosse/courses/csc321{_}2017/tutorials/tut4.pdf

[41] C. Bauckhage, "NumPy / SciPy Recipes for Data Science : k -Medoids Clustering," vol. 1, no. April, pp. 1–6, 2015. [Online]. Available: https://www.researchgate.net/publication/272351873

[42] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014. [Online]. Available: https://doi.org/10.7717/peerj.453

[43] C.-a. Boiangiu, M. Prodan, I. I. Bucur, and G. Vlasceanu, "Combining Tesseract and Asprise Results To Improve Ocr Text," no. May, 2019.

[44] N. Pawar, Z. Shaikh, P. Shinde, and P. Y. P. Warke, "Image to Text Conversion Using Tesseract," *International Research Journal of Engineering and Technology (IRJET)*, vol. 06, pp. 516–519, 2019.

[45] A. Rosebrock, "Using tesseract ocr with python," *pyimagesearch*, 2017. [Online]. Available: https://www.pyimagesearch.com/2017/07/10/using-tesseract-ocr-python/

[46] M. Awad and R. Khanna, *Machine Learning*. Berkeley, CA: Apress, 2015, pp. 1–18. [Online]. Available: https://doi.org/10.1007/978-1-4302-5990-9{_}1

[47] A. Balamurali and O. G. Assistant, "Deep Learning and Neural Networks," *Deep Learning and Neural Networks*, vol. 14, no. 8, pp. 1860–1864, 2019.

[48] G. S. Fu, Y. Levin-Schwartz, Q. H. Lin, and D. Zhang, "Machine Learning for Medical Imaging," *Journal of Healthcare Engineering*, vol. 2019, pp. 10–12, 2019.

[49] scikit learn, "An introduction to machine learning with scikit-learn," *scikit-learn*, 2018. [Online]. Available: https://scikit-learn.org/stable/tutorial/basic/tutorial.html

[50] S. Kiyohara, T. Miyata, and T. Mizoguchi, “Prediction of grain boundary structure and energy by machine learning,” vol. 18, pp. 1–5, 2015. [Online]. Available: http://arxiv.org/abs/1512.03502

[51] A. Agarwal and A. Saxena, “Malignant Tumor Detection Using Machine Learning through Scikit-learn,” vol. 119, no. 15, pp. 2863–2874, 2018. [Online]. Available: http://www.acadpubl.eu/hub/

[52] J. Carreira, P. Fonseca, A. Tumanov, A. Zhang, and R. Katz, “A Case for Serverless Machine Learning,” *Systems for ML*, 2018.

[53] Puresec, “The 12 Most Critical Risks for Serverless,” 2019. [Online]. Available: www.facebook.com/groups/789522244477928

[54] AWS, “Aws lambda,” *AWS*, 2019. [Online]. Available: https://aws.amazon.com/lambda/

[55] R. Baker, P. Macharrie, H. Phung, J. Hansford, J. Reddy, S. Causey, J. Sobanski, S. Walsh, R. Niemann, D. Beall, N. D. E. Proving, and G. Npg, “SSC19-IX-01 Amazon Web Services (AWS) Cloud Platform for Satellite Data Processing,” no. February, 2019.

[56] V. Technologies, “Deploy serverless, event-driven python applications using zappa,” *Medium*, 2019. [Online]. Available: https://medium.com/velotio-perspectives/deploy-serverless-event-driven-python-applications-using-zappa-8e215e7f2c5f

[57] R. Chandramouli, “Security Strategies for Microservices-based Application Systems NIST Special Publication 800-204 Security Strategies for Microservices-based Application Systems.”

[58] J. L. Martin Fowler, “A definition of this new architectural term,” *martinfowler*, 2014. [Online]. Available: https://martinfowler.com/articles/microservices.html

Glossary

API A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service. 44

Django High-level Python Web framework that encourages rapid development and clean, pragmatic design. 47

EMEA Europe, the Middle East and Africa. 11

Glacier Storage service optimized for infrequently used data, or "cold data". 47

S3 Simple Storage Service, object storage service that offers industry-leading scalability, data availability, security, and performance. 3, 4, 47

SNS Simple Notification Service, a highly available, durable, secure, fully managed pub/sub messaging service that enables you to decouple microservices, distributed systems, and serverless applications. 3, 4

SQS Simple Queue Service, fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications. 3, 4