Building Tools to Analyze the Files of the Uruguayan Dictatorship: Information Extraction From the Personal Records of Organización Coordinadora de Operaciones Antisubversivas (OCOA)

Mateo Nogueira
Instituto de Computación
Facultad de Ingeniería,
Universidad de la República,
Montevideo, Uruguay
mateo.nogueira@fing.edu.uy

Lorena Etcheverry
Instituto de Computación
Facultad de Ingeniería,
Universidad de la República,
Montevideo, Uruguay
lorenae@fing.edu.uy

Gregory Randall
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería,
Universidad de la República,
Montevideo, Uruguay
randall@fing.edu.uy

***Abstract*—An automated method to analyze personal record cards generated by Organismo Coordinador de Operaciones Antisubversivas (OCOA) during the civic-military dictatorship in Uruguay between 1973 and 1985 is presented. These personal records are part of *Archivo Berrutti*, a collection of digitized documents, partially processed by the Uruguayan government. The main goal of this study is to extract the maximum amount of information from the personal record cards to ease the analysis by specialized teams. To achieve the goal, a methodology which combines image processing and pattern recognition techniques has been developed. This methodology takes advantage of the known geometric structure of the cards to straighten them, identify and extract relevant pieces, classify them, extract relevant fields, and transcribe crucial information such as names and identification numbers.**

***Index Terms*—OCR, template matching, historical document analysis.**

I. Introducción

Entre los años 1973 y 1985, Uruguay fue gobernado por una dictadura cívico-militar precedida por terrorismo de Estado entre 1968 y 1973. Durante este período, ocurrieron numerosas violaciones a los derechos humanos, incluyendo secuestros, torturas, asesinatos y desapariciones. La investigación de estos crímenes ha sido difícil debido a complicidad de militares, políticos e instituciones involucradas en los hechos, que obstruyeron el acceso a las fuentes de información generadas durante ese período [1].

Una de las fuentes documentales a la que se tiene acceso es el *Archivo Berrutti*. Una colección de aproximadamente tres millones de imágenes escaneadas de microfilmaciones, agrupadas en rollos de aproximadamente 2000 imágenes cada uno [1]. Los rollos son heterogéneos, con documentos generados por diversos organismos represores y organizaciones del Estado a lo largo de muchos años. Su calidad es muy variada. Los rollos están compuestos de páginas (u hojas). Cada una corresponde a una imagen digitalizada de una foto de microfilm. Los rollos se encuentran numerados, dato que fue generado durante la microfilmación.

Los rollos 570, 584, 585, 587, 588, 594, 599, 603, 607, 608, 612, 613, 614 y 615 del *Archivo Berrutti* [2] contienen fichas personales (información personal y antecedentes) generadas por la OCOA, uno de los principales organismos represores, creado con el fin de perseguir y capturar opositores [3] [4]. Debido al rol de este organismo, explorar el contenido de estos rollos es de suma importancia para comprender algunos de los hechos ocurridos durante la dictadura. Esta tarea se ve dificultada por el hecho de que los documentos se encuentran almacenados como imágenes lo que imposibilita el realizar búsquedas por texto. Además, la estructura y tipografía dificultan el uso directo de las herramientas de reconocimiento de texto u Optical Character Recognition (OCR) .

Cada ficha individual se compone de una o más plantillas con campos preimpresos y espacios para ser llenados con máquina de escribir o a mano. Las fichas consisten en una parte frontal con información personal y una parte trasera con anotaciones, antecedentes y/o observaciones. A veces, para un mismo individuo, hay más de una parte frontal o trasera.

La principal contribución de este trabajo es el desarrollo de una metodología para la extracción y transcripción automática de los campos con información en las fichas generadas por la OCOA. El enfoque utilizado se basa en técnicas clásicas de procesamiento de imágenes y herramientas de OCR basadas en redes neuronales. El conjunto de herramientas así como la metodología desarrolladas son lo suficientemente flexibles como para ser utilizadas en ficheros similares en el futuro.

Este documento se organiza de la siguiente manera: en la próxima sección se plantean los objetivos del trabajo, a

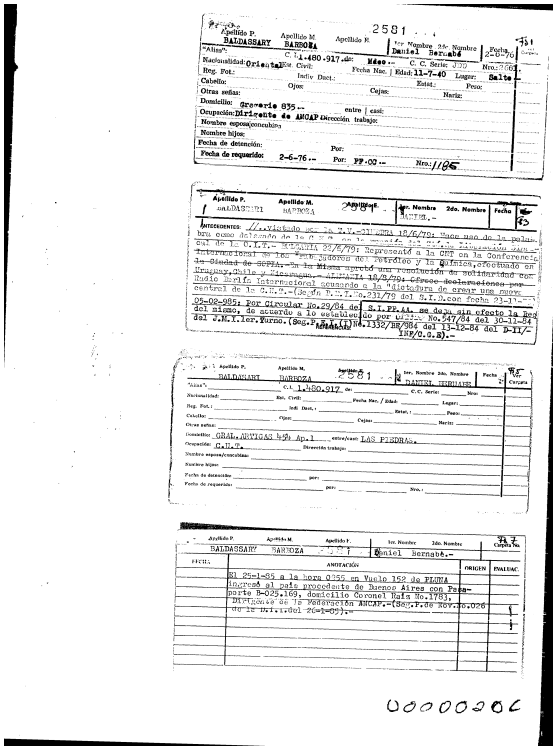


Fig. 1: Ejemplo de una hoja con cuatro fichas.

continuación se mencionan algunos antecedentes. Luego, se explica la metodología desarrollada junto con los resultados obtenidos. Finalmente, se concluye y se plantean posibilidades de trabajo futuro.

II. Objetivos

El objetivo principal del trabajo es desarrollar una metodología que permita extraer la mayor cantidad de información posible de las fichas de OCOA. En particular, extraer el contenido de los campos de las fichas y generar un sistema de búsqueda sobre los datos extraídos que permita a historiadores, abogados, periodistas, familiares y otras personas interesadas identificar y recuperar fichas según su contenido.

Previo al desarrollo de este proyecto, y dado que las fichas están en formato de imágenes, este proceso se realizaba por inspección manual de las fichas, lo cual es un proceso de búsqueda arduo y que demanda una considerable cantidad de tiempo.

Para dimensionar las dificultades del proceso de extracción de la información es necesario comprender el formato de las fichas originales y algunos detalles sobre el proceso de digitalización. Cabe señalar que este proceso fue realizado hace muchos años y que el presente proyecto sólo cuenta con las imágenes producidas por el proceso de digitalización.

Las fichas originales eran rectángulos de cartulina impresa en ambos lados, con un patrón determinado en su parte frontal y otro en su parte trasera. Para su digitalización Fueron colocadas sobre un plano y microfilmadas, de tal manera que cada imagen contiene entre una y cinco partes delanteras o traseras de fichas. Cada parte aparece en posición casi horizontal, con variaciones de inclinación entre ellas que pueden ir hasta una decena de grados. Dado que las fichas contienen información en ambos lados pero en cada imagen solo se observa uno de ellos, definimos la noción de "parte de ficha" para referirnos a la parte delantera o trasera de cada ficha. Además, el material de base con que contamos está formado por imágenes binarizadas producidas por el escaneo de esos rollos de microfilm. En la figura 1 se muestra un ejemplo de una hoja que contiene cuatro partes de ficha, todas con diferentes grados de rotación respecto a la horizontal. Hay dos partes frontales (la 1ra. y la 3ra. de arriba abajo) y dos partes traseras (las otras dos).

Todas las partes de ficha tienen la misma altura y ancho en todas las imágenes, determinados por las dimensiones físicas de las fichas originales. En algunos casos puede haber partes de ficha con otra información como fotos o huellas dactilares. En la Figura 2a se muestra un ejemplo de una parte de ficha frontal. Se observa que hay muchos campos vacíos.

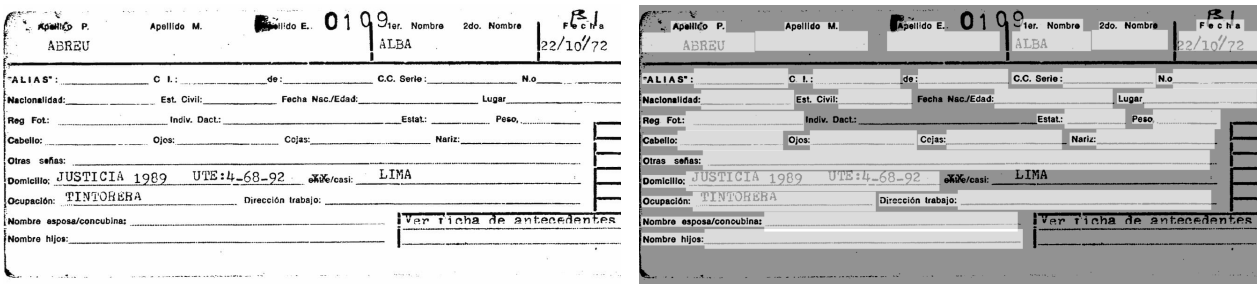
En resumen, podemos afirmar que las imágenes presentan varias partes de ficha rotadas, algunas tienen poca calidad y la tipografía no es uniforme. Si se aplica un OCR, por ejemplo Tesseract [5], se observa que lo que mejor se detecta son las etiquetas de la plantilla (Nombre, Ocupación, etc.). En algunos casos se detecta también el contenido de los campos. Sin embargo, es muy difícil saber qué texto pertenece a cada una de las partes de ficha así como también es complicado asociar el texto detectado a un campo.

Ese punto de partida implica las siguientes tareas a nivel de cada imagen (hoja) de un rollo:

1) Determinar cuántas partes de ficha contiene.
2) Recortar cada una. En esta etapa se debe además corregir la orientación (rotación) de cada parte de ficha para que queden horizontales. Esta tarea debe realizarse previo al recorte para no perder información.
3) Identificar a qué tipo corresponde cada parte (frontal o trasera). También se identificaron subclases de las partes frontales y traseras de fichas, debido en general a variaciones de la tipografía utilizada en los campos.
4) Identificar en cada parte de ficha el lugar geométrico del conjunto de etiquetas de la plantilla. Estas etiquetas están impresas, con una tipografía y posición definidas en cada subclase, configurando una plantilla conocida.
5) Identificar los campos contiguos a cada etiqueta, lugares eventualmente llenados a máquina o a mano con información específica sobre el individuo a quien corresponde la ficha.
6) Transcribir a lenguaje simbólico (texto o números según corresponda) el contenido de los campos de interés.

La Figura 2b muestra un ejemplo de parte de adelante donde ya se realizaron las tareas 1 a 5. La tarea 6 tendrá como entrada cada una de las porciones de imagen que corresponden a campos de interés (rectángulos grandes).

Además de las imágenes de fichas personales, existe un índice en formato tabla, que contiene información personal sobre los individuos. Un segundo objetivo es recortar las



(a) Ejemplo de una parte de adelante de ficha recortada y enderezada.

(b) Detección de las etiquetas (rectángulos blancos pequeños) y de sus correspondientes campos para extraer información relevante (rectángulos blancos grandes)

Fig. 2: Ejemplo de parte de ficha

líneas y reconocer el texto de dicha tabla, puesto que contiene información complementaria a las fichas. Todas las partes de ficha cuentan con un número, generalmente localizado en el tercio superior, que permite vincular las partes de ficha de una misma persona. Este índice también sufre los problemas de imágenes rotadas y calidades de imagen heterogéneas a lo largo de todos los ejemplos.

Los documentos pertenecientes al *Archivo Berrutti* son información reservada al momento de realizar este trabajo. Esto supone una dificultad adicional ya que no es posible utilizar herramientas o servicios externos que impliquen subir las imágenes a servidores de terceros, por lo que el procesamiento debe llevarse a cabo localmente o en servidores de confianza como Cluster.uy [6], una iniciativa nacional uruguaya para proveer infraestructura con alto poder de cómputo para fomentar los proyectos de investigación e innovación.

III. Antecedentes

En el caso que nos ocupa no se cuenta con datos anotados, lo que descarta el uso de métodos supervisados. El problema planteado se puede resolver con técnicas conocidas, pero ello implica el estudio de las particularidades de cada caso para realizar los ajustes y adaptaciones necesarios.

Por ejemplo en [7], se propone un sistema para reconocer montos de cheques en inglés o francés. Desarrollan un sistema para documentos manuscritos y otro para documentos impresos. Ambos son capaces de detectar los campos en los cheques y aplican técnicas de segmentación para obtener los números o caracteres. El resultado es transcrito utilizando un OCR. El ratio de reconocimiento en producción varía de un 65% a un 85%. Se menciona que el sistema es muy sensible a pequeños cambios en la tipografía de diferentes idiomas. Incluso cuando los caracteres son los mismos, en dos países pueden escribirse de manera diferente causando errores. Un ejemplo mencionado en el artículo es que el número 4 escrito en Francia es parecido al número 6 en Estados Unidos.

En [8] se desarrolla un sistema para aplicar OCR a documentos históricos impresos en alemán, generalmente diarios. Primero segmenta para detectar bloques de texto, luego extrae los renglones y finalmente aplica OCR a cada línea detectada. Para la segmentación de bloques y de líneas utilizan redes neuronales convolucionales. Para el OCR utilizan un extractor de características con redes convolucionales y luego redes LSTM para el reconocimiento. Utilizando pocos datos etiquetados los autores informan que alcanzan resultados al mismo nivel que el estado del arte, logrando desarrollar un OCR muy eficiente pero específico para textos históricos en alemán.

En [9] se utiliza una combinación de *template matching* y similitud de texto para clasificar documentos (varios tipos de facturas) y extraer campos relevantes de ellos. Se genera una base de datos de plantillas de imágenes y texto (el texto es común a todos los documentos de la misma plantilla). Al momento de clasificar una imagen candidata, se calcula la similitud visual y textual para todas las plantillas. La plantilla que obtenga la similitud visual y textual sumada más alta y que además supere cierto umbral es seleccionada como la plantilla del documento nuevo. En caso de no haber coincidencia, se envía la imagen para un etiquetado manual. Los autores utilizaron el servicio Textract de Amazon Web Services como OCR [10]. Para la similitud visual, los autores calculan la distancia coseno entre las matrices σ de la descomposición SVD de la imagen candidata y las imágenes de la plantilla. Para la similitud de texto, se aplica un OCR a la imagen candidata y luego se calcula el promedio de la distancia de edición entre las primeras y últimas n líneas del texto obtenido de la imagen con el texto de la plantilla. Una vez se determina la plantilla correcta para el nuevo documento, se seleccionan las regiones de interés donde están los campos a extraer. Este proceso se realiza por separado para cada campo. Primero, se recorta el campo en la imagen plantilla (requiere como paso previo anotar manualmente en las plantillas la posición de cada campo). Luego, se busca la zona de la imagen candidata que tenga la mayor similitud al campo recortado de la plantilla utilizando correlación. En este punto, se realiza una estimación de la ubicación del campo a extraer utilizando el texto en común en la plantilla y la imagen candidata. Finalmente, se toma el texto del OCR cuya posición se encuentre dentro de la región en que se estimó que se encontraba el campo. En promedio, los autores obtienen un 86% de accuracy.

En los ejemplos mencionados aparecen algunas técnicas

que utilizaremos en la aproximación desarrollada en este trabajo: la conveniencia de separar el problema en etapas, incluyendo por ejemplo la segmentación de bloques y líneas [11], el agrupamiento en clases [12] y el reconocimiento de ciertos patrones. El uso del *template matching* [13], [11], para determinar la existencia y posición de ciertos patrones (por ejemplo el texto de los campos a identificar). La conveniencia del uso de OCR basado en aprendizaje para la transcripción [14], [15].

Un procedimiento utilizado frecuentemente en la literatura [16], [17] que empleamos de manera intensiva en este trabajo, y que exponemos brevemente a título de antecedente general, es el cálculo de la variación total del vector suma según líneas o columnas. Supongamos una región rectangular R de la imagen (esta región puede ser la imagen entera), definida por las coordenadas de su esquina superior izquierda (f_i, c_i) y las de su esquina inferior derecha (f_f, c_f). Llamemos vector suma horizontal S_h al vector formado por la suma de los niveles de gris a lo largo de cada fila de la región R:

$$S_h(j) = \sum_{i=c_i}^{i=c_f} R(i,j)$$

Dónde j representa el índice de las filas en R.

Del mismo modo podemos definir el vector suma vertical S_v al vector formado por la suma de los niveles de gris a lo largo de cada columna de la región R, en ese caso las sumas se realizan según el índice j que recorre las filas.

La variación total del vector S_h es la suma del valor absoluto de sus diferencias sucesivas:

$$VT(S_h) = \sum_{j=f_i+1}^{j=f_f} |S_h(j) - S_h(j-1)|$$

$VT(S_v)$ se calcula de manera análoga.

IV. Metodología

La solución desarrollada se compone de varias etapas según se observa en la Figura 3. Recordemos que nos referimos como *hoja* a una imagen individual de un rollo de microfilm. Usaremos indistintamente hoja e imagen como sinónimos de acá en más. Hay dos entradas: hojas que contienen fichas y hojas del índice. En el primer caso, es necesario determinar cuántas partes de ficha hay en la imagen, separarlas, estimar su inclinación respecto a la horizontal, rotarlas de manera tal que queden horizontales, y guardar cada parte de ficha en una imagen independiente. A continuación, se clasifican las partes de ficha según la clase, se extraen los campos con información, se extrae el número de ficha y finalmente se transcribe el texto que aparece en los campos identificados. En el caso de las hojas con índices, se procesan las imágenes para detectar y extraer cada celda de la tabla a las que luego se le aplica OCR. Finalmente, los resultados de ambos procesos pueden visualizarse en una interfaz gráfica. A continuación detallamos cada paso.

A. Extracción de partes de fichas de las hojas

Dada una imagen, el objetivo es separar cada una de las partes de ficha en una imagen individual. Las imágenes son binarias con valores cero y uno. En la primer etapa, se invierten el valor de gris de forma que el blanco tenga valor cero. Se calcula el vector S_h para toda la imagen y se utiliza para determinar cuántas partes de ficha hay en la misma. Las regiones con fichas producirán regiones con altos valores en S_h separados por regiones con valores cercanos a cero. Esto se puede ver en la figura 4, donde se muestra una hoja con tres partes de ficha junto con el vector S_h.

Se aplica una umbralización de S_h y una dilatación morfológica del resultado a fin de eliminar pequeños mínimos formados por zonas débiles dentro de una ficha. El parámetro de la dilatación está determinado por el alto de una ficha A_f, que es conocido. Se obtiene un vector con valores iguales a 1 donde hay ficha y valores iguales a 0 donde no hay. El número de fichas existentes en la imagen es

$$N_{pf} = (\sum S_h)/A_f$$

La posición de las fichas se obtiene maximizando la suma de los valores de S_h en N_{pf} ventanas de tamaño A_f, bajo la restricción de que no puede haber intersecciones entre las ventanas (es decir entre fichas) y que tampoco puede quedar una parte de ficha fuera de la imagen. Es decir, las ventanas deben estar totalmente contenidas en el vector S_h. Al maximizar la suma, se busca dejar dentro de las ventanas la mayor cantidad de texto y plantilla posibles. Para buscar el óptimo se realiza una búsqueda exhaustiva. Dado que el índice del vector corresponde al índice de las filas de la imagen, el resultado permite definir franjas de la misma donde está contenida una ficha.

Cada rectángulo R_i contiene una ficha que puede estar rotada hasta 15 grados respecto a la horizontal. Se procede a calcular el vector suma horizontal $S_h^{\theta_k}(R_i)$ para la región R_i en un rango de ángulos $\theta_k = -15, 15$. El ángulo θ_k que maximiza $VT(S_h^{\theta_k}(R_i))$ es la orientación más horizontal de la ficha. Esto es bastante intuitivo, pues cuando la ficha está más horizontal en el vector suma de intensidades aparecerán más nítidamente las zonas altas correspondientes a texto separadas por zonas cercanas a cero correspondientes a los espacios entre las líneas de texto. Esto puede observarse en la figura 4. Es notorio que la primera ficha (que tiene una inclinación mayor que las otras dos), produce un vector suma horizontal S_h con variaciones menos bruscas de los picos correspondientes a las líneas de texto de la ficha.

El vector suma vertical $VT(S_v(R_i))$ permite identificar las columnas inicial y final que contienen a la ficha. Ello permite recortar la ficha a lo largo.

Con esto finaliza esta etapa y se puede generar una imagen conteniendo solo la ficha enderezada y recortada.

B. Clasificación de partes de ficha

No se cuenta con imágenes etiquetadas por lo que se selecciona un algoritmo de *template matching* para clasificar

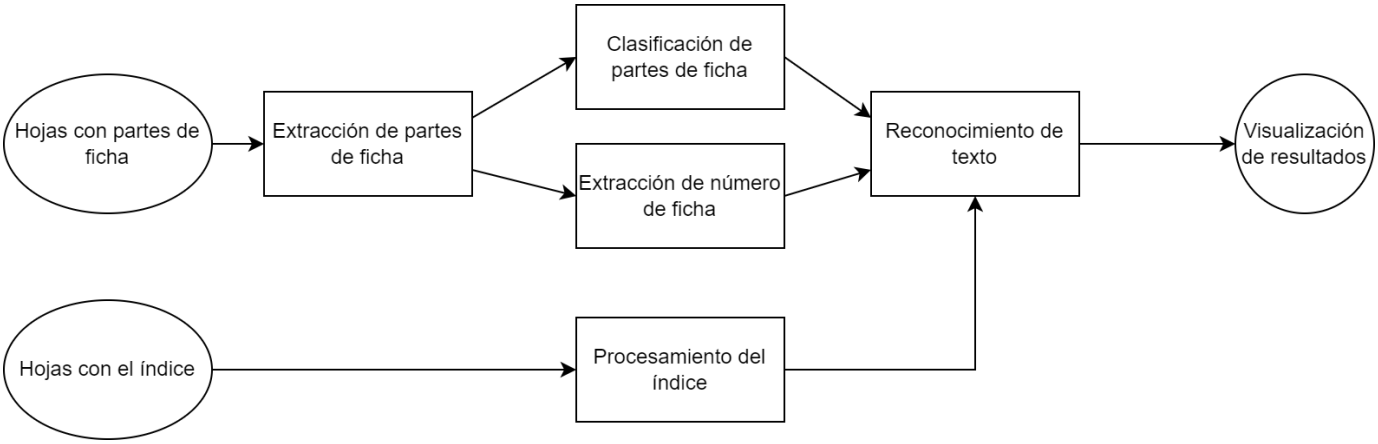


Fig. 3: Diagrama con las principales etapas de la metodología definida.

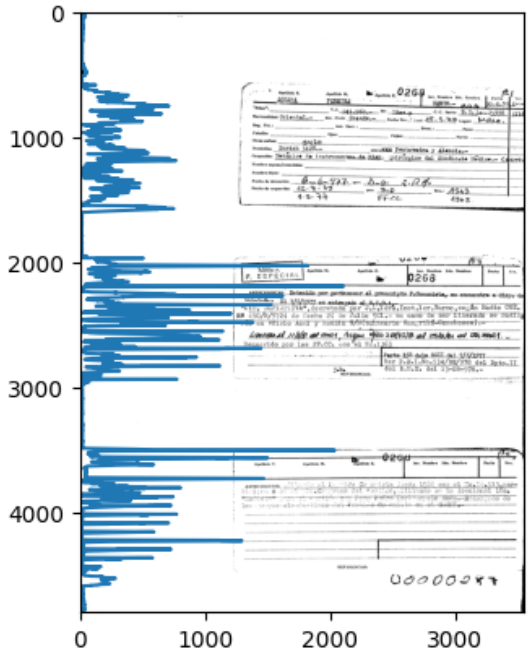


Fig. 4: Ejemplo de una hoja junto con el vector S_h, en azul.

las partes de ficha detectadas. Todas las partes de ficha de una misma clase tienen en común las etiquetas de los campos y lo único que cambia es el contenido de estos. Por ejemplo todos tendrán la misma etiqueta **nombre** (con igual tipografía) pero el contenido del campo será diferente. Con el objetivo de utilizar estos campos para la clasificación, se busca una imagen de cada clase con poco ruido y se extraen las etiquetas que los caracterizan.

Por ejemplo, en la figura 2b están marcadas las etiquetas de los campos que se utilizaron para realizar la clasificación. Las etiquetas que aparecen en esa ficha son: Apellido P, Apellido M, Apellido E, 1er Nombre, 2do Nombre, Fecha, ALIAS, C.I, de, C.C. Serie, No., Nacionalidad, Est. Civil, Edad, Lugar, Reg. Fot, Estat, Peso, Cabello, Ojos, Cejas, Nariz, Otras señas, Domicilio, Ocupación, Dirección trabajo, Nombre esposa/concubina, Nombre hijos. Algunas son abreviaciones. Muchas de ellas aparecen en todas la partes frontales de fichas pero algunas solo en algunas o se diferencian en la tipografía o por la posición dentro de la estructura general de la ficha. Ello da lugar a la definición de 10 clases de partes frontales.

La imagen de cada etiqueta es recortada y constituye un *template* (que llamaremos patrón) que puede ser buscado en las imágenes mediante técnicas de *template matching* [13]. Cada clase está definida por un conjunto de patrones formado por recortes de etiquetas de la ficha representativa de dicha clase. Por ejemplo, en la Figura 5 se pueden observar todos los patrones asociados a la clase 1.

La clasificación de las partes de ficha puede dividirse en dos problemas: el primero, determinar qué patrones se encuentran en la ficha y en qué posición. El segundo, determinar la clase a la que pertenece la parte de ficha analizada en función del conjunto de patrones que se lograron detectar en la misma.

Para conocer si un patrón se encuentra en una imagen se utiliza la correlación utilizando la distancia coseno [11]. El máximo de la correlación entre el patrón y la imagen indica la posición en que hay mayor similitud entre el patrón y la zona correspondiente de la imagen. Si el coeficiente de correlación en esa posición supera un umbral (0,93 en nuestro caso), se determina que dicho patrón está presente en esa posición y se considera que hubo un *match*.

Este proceso se itera para todos los patrones de todas las clases. En caso de detectar 13 o más patrones de una clase en una parte de ficha, se clasifica esa parte de ficha como de esa clase. Se comienza probando con los patrones de las clases más frecuentes a fin de acelerar el proceso. El ruido puede provocar que algunas imágenes se clasifiquen incorrectamente o no se clasifiquen en ninguna clase.

C. Extracción de campos

Luego de clasificada una parte de ficha, como subproducto se obtiene la ubicación de las etiquetas. La posición donde se encuentra la información de los campos asociados a cada etiqueta es relativa a la posición de la etiqueta. De este modo,

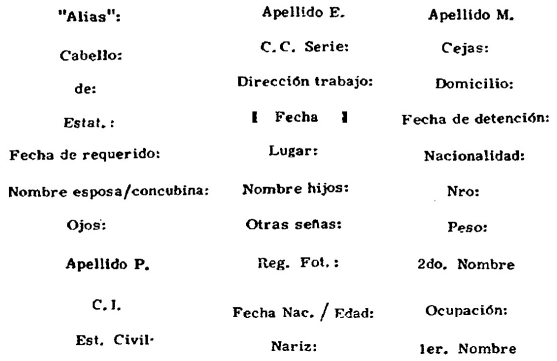


Fig. 5: Ejemplo de todos los patrones extraídos para una de las clases de parte frontal de ficha.

para cada patrón se define la ubicación del campo y su tamaño. Se logran extraer así los campos de cada una de las fichas clasificadas. La Figura 2b muestra un ejemplo de una parte frontal donde se resalta la ubicación de las etiquetas detectadas y los campos asociados. El texto que aparece en cada campo es transcrito utilizando un OCR.

D. Extracción de número de ficha

Extraer el número de cuatro o cinco dígitos que se encuentra en la parte superior de cada parte de ficha es importante ya que permite vincular todas las partes de ficha asociadas a una misma persona y a su línea en el índice.

A diferencia del caso anterior, no existe una etiqueta que se pueda utilizar como patrón. Este número se puede encontrar en cualquier posición del tercio superior de la imagen. Además, puede aparecer ligeramente rotado con respecto al resto de la ficha, dado que no forma parte del texto impreso sino que es producto de un estampado independiente.

Estos números tienen la misma tipografía y tamaño en todas las partes de ficha, con independencia de su clase. Esto permite utilizar nuevamente *template matching*. Se genera un patrón para cada dígito. En general el patrón con la correlación más alta en el tercio superior de la ficha se posiciona correctamente sobre uno de los dígitos del número. Dado que el número tiene como máximo cinco dígitos, esta detección primaria permite acotar la zona de búsqueda a un rectángulo formado por el dígito detectado y una región equivalente a cuatro dígitos a ambos lados. De esta forma se obtiene una zona de búsqueda compuesta por nueve rectángulos de los cuales sabemos que solo cuatro o cinco tienen un dígito. A partir de este punto, se intentaron dos técnicas para encontrar los restantes dígitos.

La primera técnica consiste en aplicar *template matching* hacia la izquierda y derecha del primer dígito detectado iterativamente. Primero, se busca hacia la derecha. En caso de que haya un patrón de los diez que supere un umbral en una posición candidata, se continúa con el siguiente. Cuando se encuentra una posición en que ningún patrón supera el umbral o se probaron todas las posiciones, se repite el mismo proceso hacia la izquierda. Una desventaja de esta técnica es que se pueden detectar más dígitos de los que realmente hay. Además, es posible que haya un dígito con mucho ruido en el que ningún patrón supere el umbral haciendo que ese dígito y todos los que se encuentran a continuación no sean detectados.

La segunda técnica utilizada se idea con el objetivo de mitigar los problemas mencionados con la técnica anterior. Con los patrones correspondientes a los 10 dígitos, se aplica *template matching* en todas las posibles posiciones dentro de la región de búsqueda y se genera un vector con los coeficientes de similitud máximos en cada caso. Luego, se busca la combinación que maximiza la suma de los coeficientes de correlación máximos dentro de una ventana de tamaño igual a la cantidad de dígitos del número buscado (5 en este caso). Esta técnica evita el uso de un umbral, ya que se considera siempre el patrón del dígito con máximo coeficiente de similitud. Además, es robusto a la presencia de algún dígito con ruido, pues se valora la suma de la correlación de los patrones de todas las cifras y las cifras vecinas pueden compensar la baja performance de un dígito individual.

E. Procesamiento del índice

Los índices de los rollos contienen datos en formato tabla, con tipografía uniforme en todos los rollos y celdas del mismo tamaño. El problema principal para extraer los datos fue detectar donde comenzaba la columna de más a la izquierda y la separación entre las filas. Además, en algunos casos la tabla se encontraba rotada.

El primer paso al momento de procesar el índice consiste en aplicar el algoritmo de alineación por proyección de perfil [18] para alinear la tabla. Luego, con el objetivo de detectar el comienzo de la tabla se calcula la suma de intensidades por columna S_v y se busca maximizar la posición de una ventana de tamaño igual al largo de la tabla. Se recorta la imagen en los índices donde se da el máximo.

Para detectar las filas se aplica un algoritmo se segmentación de líneas inspirado por el utilizado en [19]. Se realiza el cálculo de la suma de intensidades por filas y se buscan máximos en el vector resultante (el valor blanco, que significa separación se representa como 1). Además se agrega la restricción de que debe haber por lo menos 35 píxeles de distancia entre cada máximo, aproximadamente la altura de una línea de texto. En caso de haber dos máximos a menos de esa distancia se considera el más alto.

Finalmente, una vez detectadas las líneas y conociendo el largo de cada columna, se logra extraer cada una de las celdas de la tabla. A las imágenes resultantes se les aplica OCR para reconocer el texto.

F. Reconocimiento de texto

Para el reconocimiento de texto de todas las imágenes generadas (celdas del índice y campos extraídos de las fichas), se utilizan dos algoritmos de OCR.

El primer OCR utilizado es Calamari [15], una herramienta abierta que toma como entrada líneas de texto. Utiliza redes neuronales convolucionales como extractor de características y redes LSTM como método de reconocimiento. La biblioteca permite cambiar fácilmente la arquitectura al entrenar un modelo nuevo. En este trabajo se utiliza la arquitectura por defecto.

El segundo OCR utilizado es Tesseract [5]. Creado en 1984 por HP se convirtió en SW abierto en 2005. Entre 2006 y 2018 fue desarrollado por Google. En su última versión, utiliza redes neuronales de tipo LSTM, similar a Calamari, pero no soporta entrenamiento o inferencia utilizando GPU, lo que lo hace más lento.

Ya que no se cuenta con datos etiquetados de las fichas, se decide utilizar *fine-tuning* para mejorar los resultados. Como parte del proyecto Cruzar, se ha desarrollado el sistema LUISA, que ha permitido transcribir de manera colaborativa miles de páginas del *Archivo Berrutti* [20]. Como resultado de ese proyecto, se cuenta con un subconjunto de los datos del *Archivo Berrutti* que están etiquetados a nivel de líneas de texto. Ello permitió conformar un conjunto de entrenamiento de 21.056 líneas, uno de validación con 6.016 líneas y uno de evaluación con 3.008 líneas. Se entrenaron ambos modelos (Calamari y Tesseract) utilizando estos datos.

Se transcribieron algunos campos considerados relevantes para la búsqueda de información en una Base de Datos. Los campos o columnas utilizadas fueron: nombres del índice, número de ficha del índice, nombre y apellidos de las fichas, número de documento de identidad, número de ficha extraído de las fichas. Para cada uno de esos campos o columnas, se generó un conjunto de datos de entrenamiento, uno de validación y uno de evaluación etiquetando imágenes y luego para cada uno de ellos se entrenó un modelo de Tesseract y uno de Calamari.

Estos modelos fueron utilizados para transcribir los otros campos o columnas en las que no se etiquetaron datos. Los modelos para nombres del índice se utilizaron para el resto de las columnas con datos de texto del índice. Los modelos para número de ficha del índice se utilizaron para las restantes columnas del índice que contenían números. Los modelos de documento de identidad de las fichas fueron utilizados para los campos numéricos de las fichas. Finalmente, los modelos entrenados con nombres y apellidos de las fichas se utilizaron para los restantes campos de las fichas.

Se intentó además corregir algunos de los resultados del reconocimiento utilizando diccionarios. Se generó un diccionario de organizaciones políticas activas en el periodo y presentes en estos textos (se detectaron nueve siglas en total). Cuando se detecta una palabra similar, se cambia el texto detectado por la palabra en el diccionario que se encuentra a menor distancia de edición. En caso de hallar una palabra con distancia menor o igual a 1 entonces se corrige el resultado. El uso de una distancia de edición tan pequeña solo permite corregir errores menores. No es posible utilizar una distancia de edición mayor ya que podría haber conflictos entre las palabras del diccionario al ser siglas de pocos caracteres. Por ejemplo, si se obtiene como resultado para una organización ABR, se encuentra a distancia 2 tanto de OPR[1] como de PCR[2] y no es posible determinar a cuál corresponde.

También se utiliza un diccionario de apellidos con 25.057 entradas para corregir nombres. En caso de detectar un apellido que pertenece al diccionario, no se realiza ningún cambio. Si el apellido no pertenece al diccionario, entonces la palabra detectada se sustituye por el apellido del diccionario que se encuentre a menor distancia de edición.

V. Resultados

La evaluación de resultados se realiza para cada una de las etapas del proceso.

A. Extracción de partes de fichas de las hojas

Se considera que una parte de ficha fue extraída correctamente si la misma se encuentra contenida totalmente en la imagen resultante, no hay fichas vecinas dentro de la imagen y la orientación es correcta.

Para la evaluación se considera un conjunto aleatorio de 50 hojas de un rollo. En total, las hojas seleccionadas contienen 150 partes de ficha. Se evalúa tanto la detección de la cantidad de fichas como el resultado de extraerlas.

Se detectaron únicamente dos instancias en las que se encontró una cantidad incorrecta de fichas. En una de ellas, se esperaban cinco partes de ficha, pero solo se detectaron tres. En la segunda instancia, la hoja en cuestión debía contener tres partes de ficha, pero solo se detectaron dos. En ambos casos, las partes de ficha no extraídas corresponden a partes de atrás, que no cuentan con una plantilla definida y además tienen pocas líneas de texto. La falta de información (fichas vacías) explica que el algoritmo no las haya detectado. En total, se detectaron 147 partes de ficha sobre un total de 150 disponibles, un 98% de partes de ficha correctamente detectadas. De las fichas detectadas, siete no se extrajeron correctamente. Estas excepciones corresponden exclusivamente a hojas que contienen cinco partes de ficha muy apretadas y parcialmente sobrepuestas. Todas estas extracciones incorrectas pertenecen solo a cuatro hojas. En suma, de las 147 fichas detectadas se extrajeron 140 correctamente, representando un 95% de extracciones exitosas.

Considerando que había 150 partes de ficha y se extrajeron correctamente 140 entonces se tiene que un 93% del total de partes de ficha que se extrajeron correctamente. Una vez

[1]Organización Popular Revolucionaria 33 Orientales

[2]Partido Comunista Revolucionario

ejecutado el algoritmo en todos los rollos, se obtuvieron 61.476 partes de ficha.

B. Clasificación de partes de ficha

Para evaluar los resultados de la clasificación, se seleccionaron aleatoriamente 10 ejemplos de cada una de las clases de parte frontal, 20 ejemplos que son partes traseras y 10 partes de ficha que son fotografías o huellas dactilares (correspondientes a una categoría *otros*). Esto da un total de 130 partes de ficha. Se ejecutó el algoritmo de clasificación sobre estas imágenes y se generó la matriz de confusión de la Figura 6.

Se observan muy pocos errores en la matriz de confusión. Ocho clases no tienen errores. Para las clases restantes, hay una parte trasera que se clasificó incorrectamente en la categoría *otros*. Tres partes de ficha de la clase 5 fueron incorrectamente clasificadas dentro de la categoría *otros*. Una parte de ficha de la clase 7 y una de la clase 10 fueron clasificadas incorrectamente dentro de otra clase.

Analizando las imágenes clasificadas incorrectamente para intentar comprender qué sucedió, en el ejemplo de la parte trasera clasificada incorrectamente, se observa que la etiqueta de la plantilla, que es el patrón utilizado para clasificar, quedó cortado al recortar la parte de ficha y por lo tanto no fue detectado. Señalemos que la parte trasera tiene pocas etiquetas pues está compuesta fundamentalmente de espacio para escribir. El ejemplo mal clasificado de la clase 7, tiene una calidad muy mala y posiblemente por eso no haya sido clasificada correctamente. En el caso de la ficha de la categoría 10 y las tres fichas de la clase 5 clasificadas incorrectamente, no se observa ningún problema en las imágenes a simple vista. El accuracy global obtenido del experimento es de 95%.

C. Extracción de número de ficha

Para la evaluación de esta etapa, se seleccionan aleatoriamente 100 imágenes. Se considera que la extracción fue correcta si en la imagen resultante se encuentra todo el número. Para el primer método utilizado, se extrajeron correctamente 97 números mientras que para el segundo este número desciende a 89 casos. Ningún caso se extrajo incorrectamente con ambos métodos. Por lo tanto, utilizar ambos métodos en simultáneo fue una decisión acertada.

D. Procesamiento del índice

Se considera que una imagen del índice se encuentra adecuadamente cortada si cada celda contiene la información correcta. Esto es, la información de la celda debe coincidir con la columna a la que pertenece y además, el texto no debe estar cortado horizontalmente.

Se seleccionaron aleatoriamente 5 imágenes por rollo y se analizó si la extracción de las celdas fue satisfactoria. En total, son 115 imágenes para analizar, representando un poco más del 10% del total de imágenes.

De las 115 imágenes, 98 fueron procesadas correctamente y 17 incorrectamente. Esto es un 85% del total.

E. Reconocimiento de texto

Para el reconocimiento de texto, se utiliza CER (Character Error Rate) como medida. Cuanto más cercano a cero sea el valor, mejor es el resultado. Los conjuntos de datos generados se separaron en un 90% como conjunto de entrenamiento y validación y el 10% restante fue utilizado para evaluación.

En la tabla I se pueden observar los resultados de ambos OCR para cada uno de los conjuntos de datos utilizados, para los conjuntos de entrenamiento y evaluación. Los modelos *baseline* son los que se obtuvieron al entrenar con los datos etiquetados de LUISA y los ajustados son los que se obtuvieron una vez que se entrenó con los datos etiquetados de las fichas y el índice.

Los resultados muestran variaciones significativas entre los diferentes conjuntos de datos y los modelos utilizados. En general, se observa que los modelos ajustados tienen un desempeño notablemente mejor en comparación con los baseline, tanto en entrenamiento como en evaluación. Por ejemplo, en el conjunto formado por nombres y apellidos de las fichas, se aprecia que el modelo de Calamari ajustado presenta un CER de 0,0154 en el conjunto de entrenamiento y 0,0794 en el conjunto de evaluación, mientras que el modelo Tesseract ajustado alcanza un CER de 0,0689 en entrenamiento y 0,1145 en evaluación. Estas cifras muestran una mejora considerable del reconocimiento en comparación con los modelos baseline. Los modelos Tesseract muestran una tendencia a tener un desempeño ligeramente inferior en comparación con los modelos Calamari en la mayoría de los conjuntos de datos. A excepción del conjunto de datos formado por números de ficha extraídos de las fichas.

Estos resultados demuestran la utilidad de ajustar los modelos para adaptarse específicamente a los conjuntos de datos. Es importante resaltar que el entrenamiento requiere contar con datos debidamente etiquetados o tiempo disponible para etiquetarlos. El etiquetado es un proceso muy costoso en tiempo. No obstante, a la luz de los resultados obtenidos, en el presente caso se puede concluir que fue una decisión acertada.

En cuanto a la corrección de resultados, en el caso de los apellidos el diccionario se encontraba bastante incompleto y muchos apellidos que fueron detectados correctamente quedaron mal al ser corregidos. En el caso de las organizaciones, en que se contaba con un diccionario completo, mejoraron los resultados luego de realizada la corrección. Para evaluar se seleccionaron 99 ejemplos manualmente. En el caso de Calamari, antes de realizar la corrección, en 89 de ellos el texto había sido detectado correctamente y este número aumenta a 95 luego de realizada la corrección. Con Tesseract, solamente 54 ejemplos habían sido reconocidos correctamente, número que aumenta hasta 84 luego de corregir. Esto muestra una mejora muy sustancial en el caso de Tesseract, explicable en parte por su peor performance respecto a Calamari.

VI. Conclusiones

Se trabajó con fichas de la OCOA perteneciente al *Archivo Berrutti*. El objetivo principal consistió en extraer la mayor cantidad de información posible de manera de poder realizar

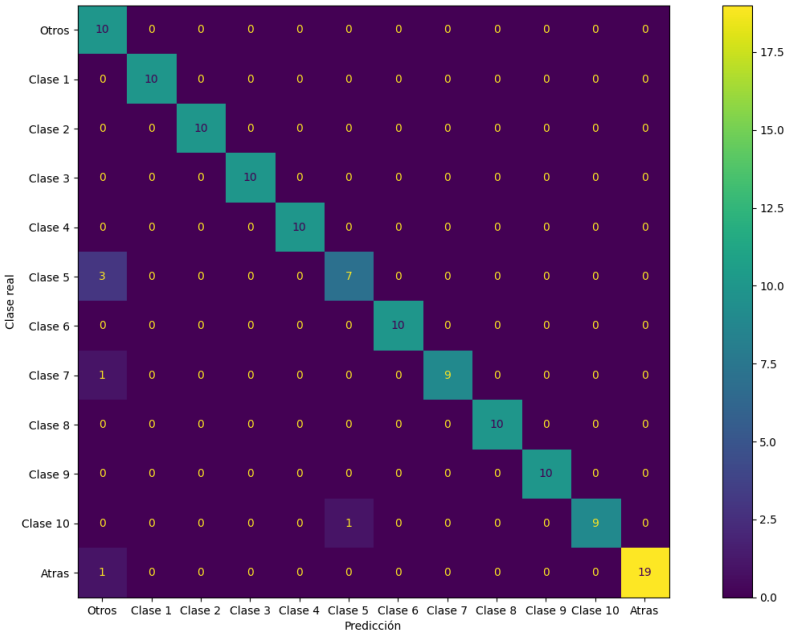


Fig. 6: Matriz de confusión de los resultados del experimento de clasificación.

búsquedas dentro del fichero fácilmente y poder conocer qué personas están en el fichero.

Se comenzó por desarrollar una técnica para separar todas las fichas pertenecientes a una hoja. Luego, se generó una técnica para clasificar cada una de las partes de ficha y extraer los campos con información relevante utilizando *template matching*.

Asimismo, se ha trabajado en el procesamiento del índice, el cual se presenta en formato de tabla junto con las fichas. Se ha diseñado una técnica para separar las filas y columnas de la tabla, y se han guardado cada una de las celdas en imágenes individuales.

Para reconocer el texto de las imágenes extraídas se trabajó con dos OCR, Tesseract y Calamari. Ambos han demostrado rendimiento muy bueno en otras investigaciones. Fue necesario dedicar tiempo a etiquetar datos ya que los modelos con los que se contaba no dieron buenos resultados. Luego del entrenamiento de los OCR, se obtuvieron buenos resultados de detección de texto en los campos que se utilizaron.

En suma, se desarrolló un sistema, basado tanto en técnicas clásicas como más modernas, que permite el tratamiento automatizado del fichero de la OCOA del *Archivo Berrutti*.

VII. Trabajo futuro

El fichero general de la OCOA con el que se trabajó durante esta investigación es uno de los ficheros pertenecientes al *Archivo Berrutti*. Existen otros ficheros, pertenecientes a otros organismos represivos, con muchas más fichas en otros rollos.

Una primer aproximación a extraer información de esos rollos puede consistir en aplicar las mismas técnicas que se utilizaron en este trabajo a esos ficheros. En particular, se puede probar utilizar *template matching* como método de clasificación y extracción de campos para luego intentar entrenar un modelo de OCR para aplicar a los campos extraídos.

La técnica utilizada para separar las fichas pertenecientes a una misma hoja es posiblemente la que más tenga que modificarse ya que es muy específica a los datos del fichero de la OCOA. No obstante, el método de detección de texto rotado y alineación así como el método utilizado para separar las líneas del índice son generales y por lo tanto pueden aplicarse en cualquier otra imagen del *Archivo Berrutti*, no solo en fichas.

En cuanto a las fichas de la OCOA, se debe analizar y profundizar en los resultados obtenidos por los OCR para los campos en los que no se entrenó un modelo. Sobre la parte de atrás de las fichas con los antecedentes de los individuos no se realizó ningún procesamiento. En el futuro, podría realizarse la correspondiente segmentación de líneas de las mismas y buscar entrenar un modelo de OCR. En este caso, se agrega la dificultad de que hay texto escrito a mano.

Entendemos que este trabajo puede dar insumos para procesar de manera similar otros archivos documentales con datos similares. Ello es importante pues son numerosos los archivos del periodo dictatorial en nuestro continente que deben ser procesados para avanzar en la lucha por verdad, justicia y nunca más terrorismo de estado.

Conjunto de datos	Modelo	CER Entrenamiento	CER Evaluación
Número de ficha en el índice	Calamari baseline	0,0392	0,0385
	Calamari ajustado	0,0011	0,0033
	Tesseract baseline	0,0656	0,0549
	Tesseract ajustado	0,00048	0,0033
Columna de nombre en el índice	Calamari baseline	0,0766	0,0917
	Calamari ajustado	0,0025	0,0142
	Tesseract baseline	0,1138	0,1268
	Tesseract ajustado	0,0471	0,0942
Campo nombres y apellidos de las fichas	Calamari baseline	0,2913	0,2924
	Calamari ajustado	0,0154	0,0794
	Tesseract baseline	0,3365	0,3354
	Tesseract ajustado	0,0689	0,1145
Campo documento de identidad de las fichas	Calamari baseline	0,2051	0,1848
	Calamari ajustado	0,0044	0,0157
	Tesseract baseline	0,1691	0,1563
	Tesseract ajustado	0,0035	0,0187
Número de ficha extraído de las fichas	Calamari baseline	0,4869	0,4570
	Calamari ajustado	0,0293	0,0333
	Tesseract baseline	0,5012	0,4913
	Tesseract ajustado	0,0139	0,0301

TABLE I: Resultados de aplicar OCR a todos los conjuntos de datos

AGRADECIMIENTOS

Los experimentos computacionales fueron desarrollados en Cluster.uy [6], la plataforma de computación científica de alto desempeño del Centro Nacional de Supercomputación, Uruguay.

REFERENCES

[1] L. Etcheverry, L. Agorio, V. Bacigalupe, S. Barreiro, E. Bing, S. Blixen, D. Calegari, L. Cardozo, F. Carpani, F. Chavat, D. Garat, A. Gómez, F. Hernández, V. Marabotto, G. Moncecchi, I. Ramírez, A. Rosá, J. Tiscornia, D. Wonsever, G. Randall, G. Zorron, L. Rivero, N. Patiño, J. Stabile, E. Fernandez, F. Fioritto, and R. Laguna, "A computational framework for the analysis of the uruguayan dictatorship archives," in *Proceedings of the Conference on Digital Curation Technologies (Qurator 2021), Berlin, Germany, February 8th - to - 12th, 2021*, ser. CEUR Workshop Proceedings, A. Paschke, G. Rehm, J. A. Qundus, C. Neudecker, and L. Pintscher, Eds., vol. 2836. CEUR-WS.org, 2021.

[2] Cruzar – archivos del pasado reciente. [Online]. Available: https://cruzar.edu.uy/

[3] Organismo coordinador de operaciones antisubversivas (OCOA) | sitios de memoria uruguay. [Online]. Available: https://sitiosdememoria.uy/node/929

[4] S. Blixen and N. Patiño. Un modelo de guerra sucia: El rol operativo del ocoa en la represión. [Online]. Available: https://www.cruzar.edu.uy/wp-content/uploads/2019/02/Unmodelodeguerrasucia.pdf

[5] R. Smith, "An overview of the tesseract OCR engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, 2007, pp. 629–633, ISSN: 2379-2140.

[6] S. Nesmachnow and S. Iturriaga, "Cluster-UY: Collaborative scientific high performance computing in uruguay," in *Supercomputing*, ser. Communications in Computer and Information Science, M. Torres and J. Klapp, Eds. Springer International Publishing, 2019, pp. 188–202.

[7] N. Gorski, V. Anisimov, E. Augustin, O. Baret, and S. Maximov, "Industrial bank check processing: the a2ia CheckReaderTM," vol. 3, no. 4, pp. 196–206, 2001. [Online]. Available: https://doi.org/10.1007/PL00013561

[8] J. Martínek, L. Lenc, and P. Král, "Building an efficient OCR system for historical documents with little training data," vol. 32, no. 23, pp. 17 209–17 227, 2020. [Online]. Available: https://doi.org/10.1007/s00521-020-04910-x

[9] P. Dhakal, M. Munikar, and B. Dahal, "One-shot template matching for automatic document data capture," in *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, vol. 1, 2019, pp. 1–6.

[10] (2023) Extracción inteligente de texto y datos con OCR - amazon textract - amazon web services. [Online]. Available: https://aws.amazon.com/es/textract/

[11] W. Burger and M. J. Burge, *Digital Image Processing: An Algorithmic Introduction Using Java*, ser. Texts in Computer Science. Springer, 2016. [Online]. Available: http://link.springer.com/10.1007/978-1-4471-6684-9

[12] B. S. Everitt, S. Landau, M. Leese, and D. Stahl, *Cluster Analysis*. John Wiley & Sons, Ltd, 2011.

[13] R. C. González and R. E. Woods, *Digital image processing, 3rd Edition*. Pearson Education, 2008. [Online]. Available: https://www.worldcat.org/oclc/241057034

[14] Tesseract user manual. [Online]. Available: https://tesseract-ocr.github.io/tessdoc/

[15] C. Wick, C. Reul, and F. Puppe, "Calamari - a high-performance tensorflow-based deep learning package for optical character recognition," vol. 014, no. 2, 2020.

[16] H. H. Sheng, *Digital document processing*. John Wiley & Sons, Ltd, 1983.

[17] E. Bidyut B. Chaudhuri, *Digital Document Processing*. Springer London, 2010.

[18] W. Postl, "Detection of linear oblique structures and skew scan in digitized documents," in *In Proceedings of the 8th International Conference on Pattern Recognition*, 1986, pp. 687–689.

[19] T. Pavlidis and J. Zhou, "Page segmentation and classification," vol. 54, no. 6, pp. 484–496, 1992.

[20] LUISA. [Online]. Available: https://mh.udelar.edu.uy/luisa/