Jornadas de Automática

Preprocesado de imagen y OCR para mejorar detección de smishing

Blanco-Medina, P.[a,b,*], Carofilis, A.[a,b], Fidalgo, E.[a,b], Alegre, E.[a,b]

[a]*Departamento de Ingeniería Eléctrica y de Sistemas y Automática, Universidad de León, Campus de Vegazana, 24007, Leon, España*
[b]*Investigador Colaborador en INCIBE*

To cite this article: Blanco-Medina, P., Carofilis, A., Fidalgo, E., Alegre, E. 2024. Enhancing CERT Smishing Response through Automatic SMS Screenshot URL Extraction.
Jornadas de Automática, 45. https://doi.org/10.17979/ja-cea.2024.45.10955

Resumen

La globalización de las tecnologías de comunicación ha llevado a un aumento de las estafas mediante técnicas de phishing. Los Equipos de Respuesta ante Emergencias Informáticas (CERTs) reciben capturas de pantalla enviadas por usuarios cuyos smartphones reciben mensajes sospechosos. Estos SMS tratan de suplantar compañías conocidas para persuadir a sus usuarios de tomar acciones urgentes, robando sus datos o realizando acciones no autorizadas en sus cuentas bancarias. Estos mensajes se conocen como Smishing, y los CERTs están interesados en herramientas que permitan automatizar la extracción de URLs en capturas de pantalla para verificar si contienen phishing. En este trabajo, proponemos una estrategia de extracción de URLs de capturas de pantalla que combinan técnicas tradicionales de visión artificial, como preprocesado y operaciones morfológicas, con mecanismos de detección y reconocimiento de URL para recuperar las URLs sospechosas. Evaluando nuestra propuesta en 117 capturas de Smishing que contienen 121 URLs, logramos una precisión del 61,16 % en la recuperación de URLs en mensajes Smishing.

Palabras clave:
Seguridad, Aprendizaje Profundo, Apoyo a Operadores Humanos, Redes Sociales, Sistemas de Control y Automatización para la Ayuda Internacional

Enhancing CERT Smishing Response through Automatic SMS Screenshot URL Extraction.

Abstract

The globalization of communication technologies has led to an increase in the number of scams through phishing. Computer Emergency Response Teams receive screenshots of smartphones from citizens containing short messages with suspicious messages. These SMS try to impersonate well-known companies and persuade users to take urgent action through a URL to steal their data or make unauthorized charges to their bank account. These short messages are called Smishing, and CERTs could be interested in tools that can automatically extract the URLs from these screenshots to verify later if it is a phishing URL. In this work, we propose a pipeline for Smishing URL extraction from the screenshots that CERTs may receive. We have combined traditional computer vision techniques, such as preprocessing or morphological operations, with an OCR to recognize the suspicious URLs. We have used our pipeline to 117 screenshots of Smishing messages containing 121 URLs, achieving an accuracy of 61,16 % retrieving complete URLs from Smishing screenshots.

Keywords:
Security, Deep Learning, Human operator support, Social networking, Control and Automation Systems for International Aid

*Autor para correspondencia: pblanm@unileon.es

1. Introducción

Smishing (SMS + phishing) se refiere a la estafa realizada mediante mensajes SMS que contienen enlaces a sitios web falsos, pero que parecen reales, con el objetivo final de robar las credenciales del usuario o instalar software malicioso (Rahman et al., 2023). En los últimos años se han mejorado los mecanismos y estrategias que permiten que estos mensajes de Smishing puedan atravesar las distintas protecciones de las que los usuarios disponen (Jain et al., 2020).

La interacción con dispositivos electrónicos y servicios de mensajería instantánea es una parte esencial de nuestro día a día, por lo que es común recibir mensajes de remitentes desconocidos, sin poder garantizar la fiabilidad de la información o su procedencia (Jáñez-Martino et al., 2023b). Además, gracias al empleo de envíos masivos y técnicas psicológicas, tanto los usuarios expertos como los principiantes pueden ser objetivo de estafas dirigidas al ciudadano individual, obligando mediante estos mensaje a que se interactúe con software potencialmente malicioso (Timko and Rahman, 2023).

La Figura 1 ilustra varios ejemplos de campañas de Smishing con objetivos maliciosos comparados con mensajes normales, alentando al usuario a acceder a un enlace mediante el cual se suplanta a una entidad con el objetivo final de robar sus credenciales.

A pesar de los esfuerzos de Equipos de Respuesta ante Emergencias Informáticas (CERTs) en informar y alertar a los usuarios a través de diversas plataformas oficiales, los usuarios no expertos pueden confundir mensajes legítimos con maliciosos (Rahman et al., 2023). Estas campañas de Smishing pueden resultar muy eficaces en periodos cortos de tiempo, resultando difíciles de detectar e informar (Sánchez-Paniagua et al., 2022).

Para poder prevenir este tipo de estafas, es necesario desarrollar herramientas que permitan a los usuarios distinguir los mensajes maliciosos, así como permitir a los organismos oficiales realizar un cribado masivo de información de campañas previas. Para ello, también es necesario desarrollar y mantener conjuntos de datos actualizados, debido a la volatilidad de estas campañas (Jáñez-Martino et al., 2023a).

Los métodos estándar de análisis de Smishing se centran en el contenido textual del mensaje, utilizando técnicas de procesamiento de lenguaje natural (NLP, por sus siglas en inglés) para estudiar componentes como las palabras presentes en el mensaje, el remitente, o la URL final (Goel and Jain, 2018), permitiendo distinguir un mensaje de Smishing de uno real. Estos métodos no analizan aquellos mensajes que se comparten a través de caputras de pantalla, la forma más común de compartir y alertar sobre este tipo de campañas sin incrementar el riesgo de los usuarios (Timko and Rahman, 2023).

A través de la detección y reconocimiento de texto (Optical Character Recognition en inglés) en imágenes, pueden recuperarse las URLs presentes en una imagen para luego verificar si es una campaña de smishing activa (Wang et al., 2020). Sin embargo, debido a las múltiples configuraciones y disposiciones de texto en capturas de mensajes SMS, un único modelo genérico puede no recuperar toda la información encontrada en este tipo de imágenes.

Con el objetivo de recuperar la mayor cantidad de URLs en mensajes SMS, evaluamos el uso de técnicas de preprocesado, operaciones morfológicas y algoritmos OCR, en un conjunto de datos de 117 imágenes de Smishing con un total de 121 URLs documentadas. Tras recuperar las URLs completas, proponemos posibles mejoras para aumentar la precisión en esta tarea y localizamos las técnicas más útiles en el contexto de mensajes Smishing.

El resto del artículo se organiza de la siguiente forma. En la Sección 2 presentamos trabajos relacionados con las tareas de Smishing, y detección y reconocimiento de texto. En la Sección 3 presentamos la metodología seguida en nuestra experimentación. La Sección 4 detalla los resultados obtenidos con las técnicas de preprocesado de imágenes propuestas. Por último, la Sección 5 detalla nuestras conclusiones y líneas de trabajo futuras.

2. Trabajos Relacionados

Los métodos más destacados en las tareas de detección y prevención de campañas maliciosas en servicios de mensajería utilizan técnicas de Aprendizaje Profundo (Deep Learning) en combinación con NLP (Mishra and Soni, 2023). Una vez se dispone del texto base del mensaje, se analiza su contenido buscando patrones clave que permitan organizar los mensajes en distintas categorías, como pueden ser Smishing o Spam (Jain et al., 2020).

Goel and Jain (2018) propusieron un sistema de clasificación de SMS en "maliciosos" o "no maliciosos". Dividieron su método en tres partes, que permitían comprobar si una URL se encontraba en diversas listas negras o si instalaba software malicioso. Posteriormente, se analiza la procedencia del mensaje, para finalmente analizar todo el contexto textual antes de clasificarlo como malicioso.

Mishra and Soni (2023) desarrollaron un sistema que analiza mensajes en términos de smishing tanto si contiene o no una URL. Primero, se busca si existe la URL y se compara con los cinco mejores resultados para clasificarla como legítima. Posteriormente, se analizan cinco características del texto del mensaje, que incluyen faltas de ortografía y palabras clave de smishing, antes de clasificar el mensaje completo como Smishing.

Los métodos revisados no analizan los mensajes de smishing desde la perspectiva de Visión Artificial, sino que se centran en el procesado del contenido textual. Estos métodos ignoran que la manera más común de compartir un posible mensaje de Smishing es a través de capturas de pantalla, evitando poner en riesgo a otros usuarios, por lo que no analizan la información contenida en la imagen del Smishing. Para recuperar estos datos es necesario aplicar técnicas de detección y reconocimiento de texto (Choudhary and Jain, 2018) propias del campo de la visión artificial.

Dado el contexto de la tarea, para analizar estos textos consideramos más adecuado utilizar reconocedores ópticos de caracteres, comúnmente conocidos como OCRs, que normalmente se aplican al reconocimiento de caracteres en textos mecanografiados (Li et al., 2023).

3. Metodología

Para procesar las imágenes, utilizamos el detector y reconocedor de texto Pytesseract (Smith, 2007), una implementa-

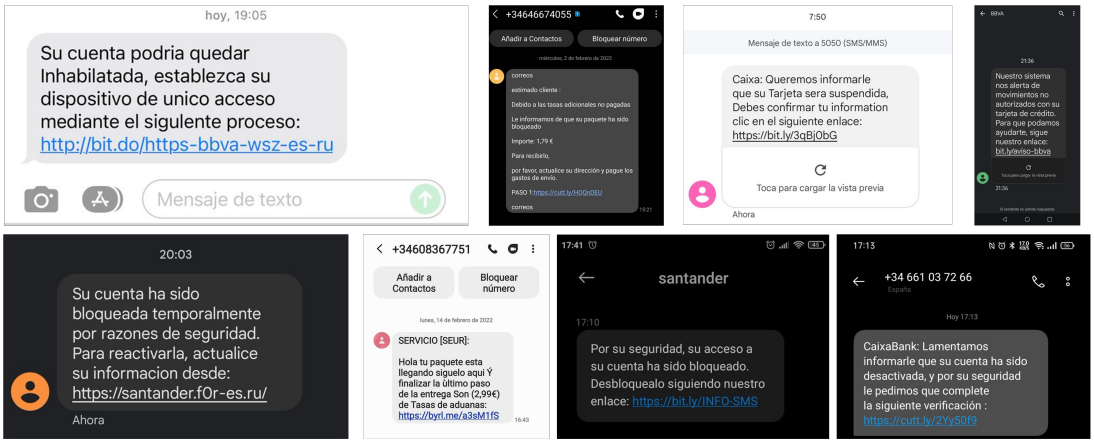

Figura 1: Ejemplos de mensajes con posibles contenidos de Smishing. Las capturas de pantalla pueden tener múltiples disposiciones de texto, colores y fuentes que dificultan la recuperación automatizada de las URLs.

ción del OCR Tesseract en Python3. Este OCR ofrece distintas opciones de lectura de la imagen, pudiendo procesarla como una única columna o bloque de texto, o realizando un escaneo completo de la imagen para recuperar la mayor cantidad de texto posible.

La selección de este OCR está motivada tanto por sus múltiples niveles de configuración como por su facilidad de implementación, al ser un método ligero que permite analizar una alta cantidad de imágenes lo más rápido posible, convirtiéndolo en una elección ideal en el contexto de los mensajes SMS.

Antes de que la imagen se procese por el extractor de texto se pueden aplicar diversas técnicas de preprocesado, como la conversión a escala de grises o el uso de la operación thresholding para resaltar ciertos tipos de texto. Proponemos el uso de las dos librerías más comunes de análisis de imágenes, CV2 y PIL, para observar cómo sus implementaciones difieren en la precisión final.

Seguidamente, aplicamos diversos preprocesados, como son la conversión de RGB a escala de grises y la posterior inversión de dicha escala. Esto se debe a las múltiples configuraciones gráficas existentes en los dispositivos móviles, lo que puede dificultar el rendimiento del OCR si se utiliza considerando una única configuración. Por último, proponemos el uso de la operación thresholding para destacar ciertos textos en mensajes cuyos caracteres no se diferencian del fondo presente, lo que permite recuperar las URLs con un mayor contraste, facilitando su reconocimiento.

Finalmente, añadimos el uso de operaciones morfológicas para resaltar más algunas regiones de texto que pueden no ser detectadas adecuadamente en el proceso de extracción, debido a su menor tamaño o distribución espacial (Uddin et al., 2012). Estas operaciones tienen la desventaja de que pueden modificar ligeramente los caracteres de la imagen, reduciendo la precisión final del método. Por ello, empleamos kernels convolucionales de tamaño pequeño (2x2, 1x2 y 2x1), para reducir su impacto en la fase de reconocimiento. La Figura 2 presenta los resultados visuales de la aplicación de operaciones morfológicas en el conjunto de datos de smishing.

Una vez hemos extraído todo el texto de una imagen, comprobamos si la URL documentada se encuentra en el texto extraído, obteniendo la precisión final del método y verificando si la URL encontrada está activa o no. La Figura 3 ilustra los pasos seguidos en nuestra experimentación.

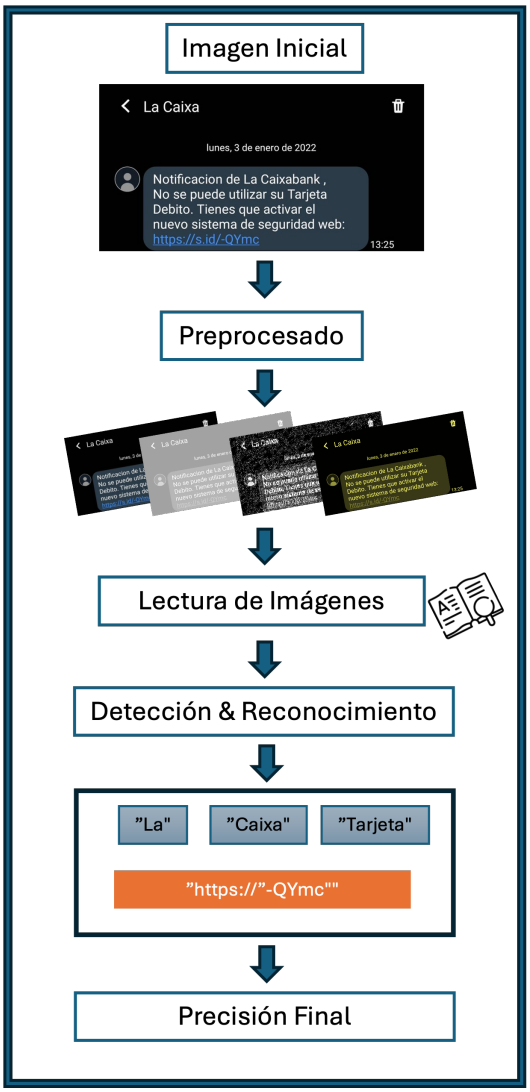

Figura 3: Propuesta para recuperar URLs en imágenes Smishing. Evaluamos los resultados en distintos niveles de preprocesado con múltiples configuraciones de lectura, evaluando la precisión en la tarea de reconocimiento de URLs.

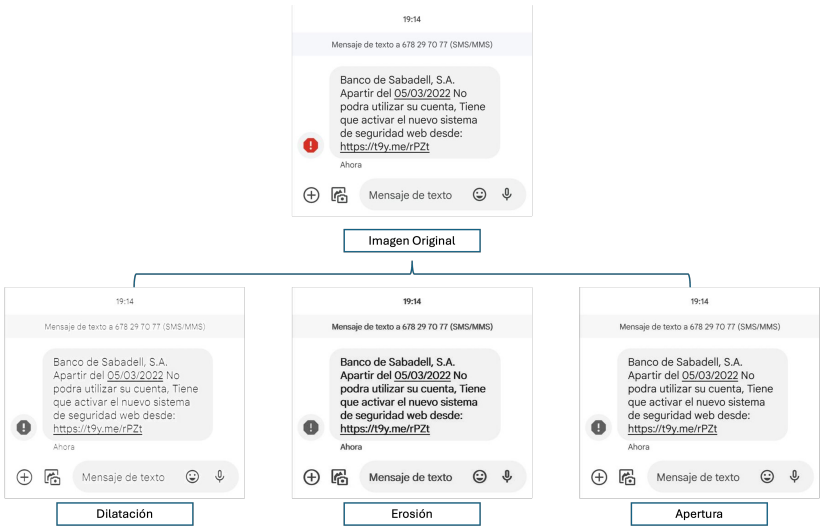

Figura 2: Aplicación de Kernels 2x1 a imágenes de Smishing de nuestro conjunto de datos. Los elementos de la imagen pueden resaltarse o reducirse, afectando a la detección y el reconocimiento final.

Evaluamos nuestra propuesta en un conjunto de datos de 117 imágenes, compuestas de mensajes que pueden contener enlaces que redirijan a un sitio web de phishing. Tras recibir estas imágenes, procedentes de un cribado recogido por INCIBE en plataformas como Facebook y Twitter, se realizó una extracción y documentación manual de las URLs contenidas. En total, se han recogido 121 URLs de sitios contiendo Smishing.

4. Experimentación

Toda nuestra experimentación ha sido realizada en un contenedor con 128 GB de RAM, dos procesadores Intel Xeon E5-2630v3 de 2,4GHz, y dos GPUs NVIDIA Titan X.

La Tabla 1 presenta los resultados obtenidos al ejecutar Pytesseract en distintos modos de lectura en seis configuraciones distintas de preprocesado. La configuración que mejores resultados obtuvo fue el preprocesado en escala de grises, con un 61,16 % de precisión en la configuración de Bloque Uniforme de Texto. Al mismo tiempo, la operación de thresholding obtuvo los resultados de precisión media más bajos entre todas las configuraciones elegidas.

Sorprendentemente, también observamos que la librería utilizada para abrir las imágenes (CV2 o PIL) influye en el reconocimiento de caracteres, con CV2 obteniendo mejores resultados que PIL en los diversos modos de lectura, pero obteniendo ambas la mejor precisión posible de 58,68 % en la configuración de bloque uniforme.

Tras obtener estos resultados, realizamos una revisión manual de las URLs no recuperadas, descubriendo que los dos errores más comunes que reducen la precisión del método son: (i) la omisión de zonas de texto detectadas debido a errores de preprocesado, y (ii) la confusión entre caracteres similares. En el contexto de las URLs, confundir caracteres como la i minúscula con la l es un error común que puede deberse a similitudes provocadas por la fuente elegida en el dispositivo móvil. Además, debido a los múltiples tamaños disponibles en los móviles, el incremento o reducción del volumen del texto puede conllevar una precisión menor en la extracción final del texto.

Analizamos también el resultado separado de los múltiples pre-procesados, revelando que algunas de las URLs son correctamente recuperadas por unas configuraciones específicas, pero erróneamente por otras. Esto se debe a los múltiples tipos de imágenes, disposiciones y distribuciones de texto de nuestro conjunto de datos, que podría beneficiarse de la combinación de múltiples preprocesados en una única ejecución de la recuperación de texto.

La Tabla 2 presenta los resultados de ejecutar operaciones morfológicas con distintos kernels sobre el preprocesado “Normal a Gris”. Apreciamos mejoras en varias de las configuraciones de lectura, aumentando de un 52,89 % a 53,72 %. Sin embargo, el mejor resultado obtenido inicialmente por esta configuración, 60,33 %, no fue superado por la aplicación de operaciones morfológicas.

Esto se debe a la modificación de los caracteres por los kernels morfológicos. A pesar de que el tamaño de las regiones detectadas se aumenta, la leve modificación de los caracteres afecta al reconocimiento final, disminuyendo la precisión total. Para solucionarlo, podrían implementarse múltiples niveles de morfología, y dividir su aplicación entre las etapas de detección y reconocimiento.

5. Conclusiones

En este trabajo hemos analizado el problema de recuperación de URLs en el contexto de campañas de Smishing, así como su importancia en el ámbito de la ciberseguridad y asistencia a equipos CERTs.

Con el fin de mejorar la recuperación de URLs en imágenes Smishing, combinamos la aplicación del OCR Pytesseract con distintos tipos de preprocesado y operaciones morfológicas. Evaluando las distintas propuestas en un conjunto de datos con 117 imágenes y 121 URLs, obtuvimos una puntuación de 61,16 % de precisión en la tarea de reconocimiento de URLs mediante el preprocesado en escala de grises, de-

Tabla 1: Precisión en nuestro conjunto de datos utilizando distintos métodos para abrir la imagen antes de procesar su contenido. Probamos distintos métodos de procesado, del color y de la escala de grises.

Modo Lectura	CV2	PIL	Gris	RGB a Gris	Thresholding	Gris Invertido
Segmentación de Página	55,37 %	51,24 %	51,24 %	52,89 %	37,19 %	52,07 %
Ejecución por Defecto	55,37 %	51,24 %	51,24 %	52,89 %	37,19 %	52,07 %
Columna Única de Texto	54,55 %	49,59 %	51,24 %	52,89 %	38,02 %	52,07 %
Bloque Uniforme de Texto	58,68 %	58,68 %	**61,16 %**	60,33 %	46,28 %	60,33 %
Localización Masiva de Texto	54,55 %	54,55 %	56,20 %	57,85 %	43,80 %	57,85 %
Localización sin Orientación	54,55 %	54,55 %	56,20 %	57,85 %	43,80 %	57,85 %

tallando como los distintos tipos de preprocesado afectan al reconocimiento de texto.

Analizamos los problemas encontrados en la detección y reconocimiento de texto, descubriendo los errores generados por la omisión de regiones detectadas de URLs, así como los fallos causados por el reconocimiento de caracteres similares. En el contexto de las URLs, al no seguir una lógica de generación en algunos de sus componentes, el reemplazo de estos caracteres puede no verse beneficiado por la ayuda de diccionarios o léxicos específicos. Además, analizamos el efecto de las operaciones morfológicas en el reconocimiento de URLs, destacando su mejora en algunas configuraciones de lectura, pero sin superar la precisión inicial de la escala de grises.

En trabajos futuros, analizaremos métodos que permitan comparar caracteres similares y reconstruir las URLs en sus distintos componentes, combinando múltiples posibilidades de palabras y caracteres. Otra línea de investigación interesante podría ser el múltiple preprocesado de las imágenes, con el objetivo de combinar múltiples técnicas en una única ejecución, uniendo la información de varias operaciones en una sola.

Agradecimientos

Este trabajo ha sido realizado gracias al Plan de Recuperación, Transformación y Resiliencia, financiado por la Unión Europea (Next Generation) gracias al Proyecto LUCIA (Lucha contra el Cibercrimen utilizando Inteligencia Artificial) concedido por INCIBE a la Universidad de León.

Referencias

Choudhary, N., Jain, A. K., 2018. Comparative analysis of mobile phishing detection and prevention approaches. In: Information and Communication Technology for Intelligent Systems (ICTIS 2017)-Volume 1 2. Springer, pp. 349–356.

Goel, D., Jain, A. K., 2018. Smishing-classifier: a novel framework for detection of smishing attack in mobile environment. In: Smart and Innovative Trends in Next Generation Computing Technologies: Third International Conference, NGCT 2017, Dehradun, India, October 30-31, 2017, Revised Selected Papers, Part II 3. Springer, pp. 502–512.

Jain, A. K., Yadav, S. K., Choudhary, N., 2020. A novel approach to detect spam and smishing sms using machine learning techniques. International Journal of E-Services and Mobile Applications (IJESMA) 12 (1), 21–38.

Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., Alegre, E., 2023a. Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach. Applied Soft Computing 139, 110226.

Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., Alegre, E., 2023b. A review of spam email detection: analysis of spammer strategies and the dataset shift problem. Artificial Intelligence Review 56 (2), 1145–1173.

Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F., 2023. Trocr: Transformer-based optical character recognition with pre-trained models. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 37. pp. 13094–13102.

Mishra, S., Soni, D., 2023. Dsmishsms-a system to detect smishing sms. Neural Computing and Applications 35 (7), 4975–4992.

Rahman, M. L., Timko, D., Wali, H., Neupane, A., 2023. Users really do respond to smishing. In: Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy. pp. 49–60.

Sánchez-Paniagua, M., Fernández, E. F., Alegre, E., Al-Nabki, W., Gonzalez-Castro, V., 2022. Phishing url detection: A real-case scenario through login urls. IEEE Access 10, 42949–42960.

Smith, R., 2007. An overview of the tesseract ocr engine. In: ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition. IEEE Computer Society, Washington, DC, USA, pp. 629–633.

Timko, D., Rahman, M. L., 2023. Commercial anti-smishing tools and their comparative effectiveness against modern threats. In: Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks. pp. 1–12.

Uddin, M. S., Sultana, M., Rahman, T., Busra, U. S., 2012. Extraction of texts from a scene image using morphology based approach. In: 2012 International Conference on Informatics, Electronics & Vision (ICIEV). IEEE, pp. 876–880.

Wang, Y., Liu, Y., Wu, T., Duncan, I., 2020. A cost-effective ocr implementation to prevent phishing on mobile platforms. In: 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security). IEEE, pp. 1–8.

Tabla 2: Precisión utilizando operaciones morfológicas en nuestro conjunto de datos. Los números indican el tamaño de Kernel en cada una de las operaciones, aplicada en los distintos modos de lectura.

Modo Lectura	Apertura			Dilatación			Cierre		
	1-2	2-2	2-1	1-2	2-2	2-1	1-2	2-2	2-1
Segmentación de Página	50,41 %	44,63 %	47,93 %	52,89 %	52,89 %	49,59 %	49,59 %	47,11 %	48,76 %
Ejecución por Defecto	50,41 %	43,80 %	47,93 %	53,72 %	52,89 %	49,59 %	49,59 %	47,11 %	48,76 %
Columna Única de Texto	50,41 %	42,98 %	47,11 %	53,72 %	52,07 %	49,59 %	47,11 %	46,28 %	49,59 %
Bloque Uniforme de Texto	58,68 %	57,02 %	57,02 %	**60,33 %**	59,50 %	57,02 %	54,55 %	51,24 %	58,68 %
Localización Masiva de Texto	48,76 %	45,45 %	54,55 %	53,72 %	49,59 %	55,37 %	49,59 %	42,98 %	57,85 %
Localización sin Orientación	48,76 %	45,45 %	54,55 %	52,89 %	49,59 %	55,37 %	49,59 %	42,98 %	57,85 %