

Actas Capitulares de la Habana

Edian Broche Castro
Roger Fuentes Rodríguez
Kevin Manzano Rodríguez
Massiel Paz Otaño
Jackson Claudio Vera Pineda

Índice

1. Introducción	1
1.1. Objetivo del Proyecto	1
1.2. Contexto	2
2. Metodología	2
2.1. Enfoque del Proyecto	2
2.2. Recursos Utilizados	3
3. Resultados	6
3.1. Logros Principales	6
3.2. Posprocesamiento y Combinación	7
3.3. Desafíos Superados	9
4. Conclusiones	9
4.1. Impacto del Proyecto	9
4.2. Recomendaciones para Futuras Implementaciones	9

1. Introducción

1.1. Objetivo del Proyecto

El objetivo de este proyecto es desarrollar una propuesta de transcripción y procesamiento de un conjunto de imágenes, para su posterior uso como dataset en futuros proyectos que identifiquen entidades nombradas y creen grafos de conocimiento.

1.2. Contexto

Se dispone de una serie documental perteneciente al fondo Ayuntamiento de La Habana del Archivo Histórico de la Oficina del Historiador de la Ciudad de la Habana.

Esta serie documental se divide en dos grupos o subseries: los libros originales (1550 - 1898) y los libros trasuntados (1550 - 1809). Los primeros destacan por su riqueza de contenido y forma; los segundos reflejan la labor del Ayuntamiento para garantizar la perdurabilidad de este tipo documental, al ser copias realizadas en la segunda mitad del siglo XIX.

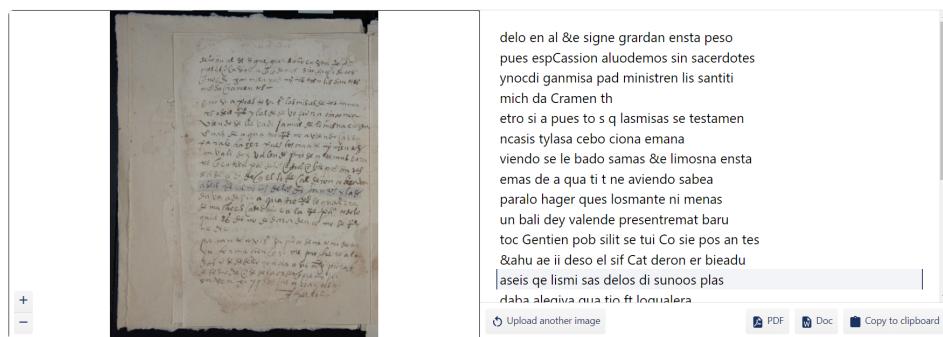
Las actas dejan la huella de una institución colonial y su devenir en el tiempo: el Ayuntamiento. En ellas se recogen los planteamientos y discusiones de aquellos problemas que interesaban a los pobladores del lugar, ya fueran de índole económica, política o social, reflejando los hechos más significativos de cada época.

Por el momento, se cuenta con el tomo 1 digitalizado, mientras que el resto se encuentra en proceso de digitalización.

2. Metodología

2.1. Enfoque del Proyecto

Existen múltiples herramientas para la transcripción de documentos históricos. Entre estas, destaca Transkribus, recomendada por el cliente por su uso en trabajos anteriores. No obstante, debido al deterioro de los documentos y al tipo específico de tipografía, los resultados no han sido satisfactorios. El bajo **accuracy** reportado por dicha herramienta, creada por especialistas en el tema, indica la complejidad del problema.



En este proyecto se afrontan dos grandes dificultades: la falta de datos para entrenar un modelo que maneje español con tipografía *procesal-cortesana*, y la complejidad propia de la tarea, que normalmente requiere la atención de un especialista (tipógrafo o paleógrafo).

Ante estas circunstancias, el proyecto se orienta hacia la búsqueda exhaustiva de datos clasificados, un procesamiento minucioso de las imágenes, el uso de varios OCR y un posprocesamiento apoyado en diccionarios del idioma y un LLM.

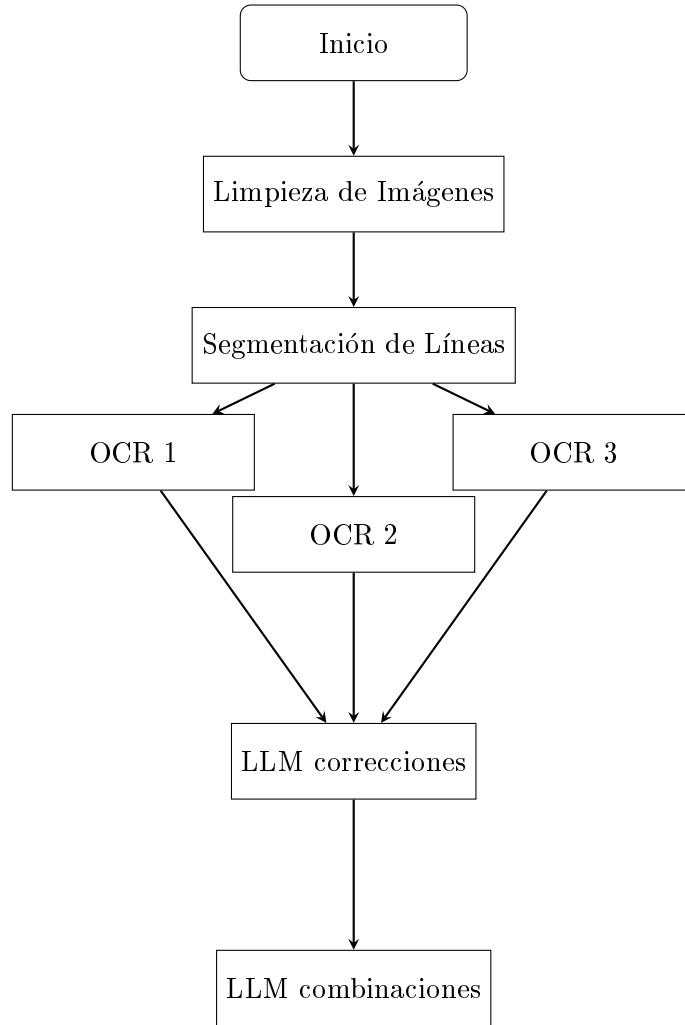


Figura 1: Diagrama de flujo con tres OCR que convergen en el LLM y un proceso de combinaciones.

2.2. Recursos Utilizados

Se ha elegido Python como lenguaje principal por su facilidad de acceso a bibliotecas de procesamiento de texto e imágenes. Entre las utilizadas destacan: **Kraken**, **PIL**, **TensorFlow**, **Numpy**, **matplotlib**, **scipy**, **spaCy**, **symspellpy**, **transformers**, **cv2** y **sam2**.

El espaciado entre palabras y líneas en la letra *procesal cortesana* no era tan generoso como lo es en la escritura moderna. La densidad del texto era mayor, lo que daba a los documentos un aspecto más compacto. Este espaciado pequeño reflejaba la naturaleza de la escritura manual en la época, donde los escribientes trataban de maximizar el uso del espacio en el pergamino o el papel.

Dadas estas características la segmentación del texto en las imágenes en líneas o palabras sería una tarea difícil. Para comenzar usamos técnicas clásicas para segmentar imágenes como proyección de histogramas, e implementamos un algoritmo sencillo que básicamente reconoce líneas de texto según la densidad de píxeles en una cierta vecindad. Posteriormente en busca de mejores resultados usamos un modelo entrenado específicamente para esta tarea, usando Kraken.

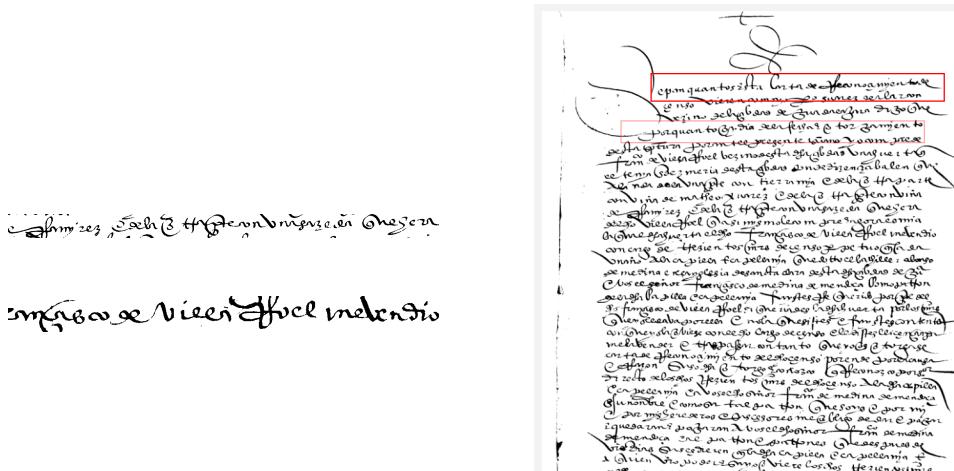


Figura 2: Segmentación de la imagen en líneas de texto usando Kraken.

Una vez segmentadas las imágenes en líneas pasamos al reconocimiento de caracteres, usando los siguientes OCR:

SimpleHTR (*Handwritten Text Recognition*) es un sistema de reconocimiento de texto escrito a mano (HTR) implementado con TensorFlow (TF) y entrenado en el conjunto de datos HTR de IAM. El modelo toma imágenes de palabras individuales o líneas de texto (varias palabras) como entrada y genera el texto reconocido. Tres cuartas partes de las palabras del conjunto de validación se reconocen correctamente y la tasa de error de caracteres ronda el 10 %. Dada la complejidad de la segmentación en las imágenes, ya que el texto está aglomerado y los trazos sobrepasan las líneas, no sería efectivo usar el modelo para reconocer palabras. Por lo tanto, usamos el modelo entrenado para reconocer líneas completas (<https://www.dropbox.com/s/7xwkcilho10rthn/line-model.zip?dl=1>). El modelo es una versión simplificada de un sistema HTR implementado en una tesis (<https://repositum.tuwien.ac.at/obvutwhs/download/pdf/2874742>). Consta de

5 capas CNN, 2 capas RNN (LSTM) y la capa de pérdida y decodificación CTC.

bdd-wormser-scriptorium-abbreviated-0.2 Se entrenó el modelo con datos de licencia abierta de HTR-United desde el siglo XVII hasta el XXI.

Hasta donde sabemos, todos los datos siguen las mismas pautas de transcripción.

El superíndice se presenta con un ^ antes de los caracteres. En algún momento, se usaría <> para marcar el tachado.

McCATMuS-nfd-nofix-V1 es un modelo de transcripción de documentos manuscritos, impresos y mecanografiados desde el siglo XVI hasta el siglo XXI. Basado en conjuntos de datos de instituciones y proyectos comprometidos con la ciencia abierta, McCATMuS proporciona un conjunto de datos interoperable que abarca más de 180 manuscritos en 7 idiomas diferentes (francés, latín, español, inglés, alemán, italiano y occitano). Incluye más de 118.000 líneas de texto y casi 4 millones de caracteres, abarcando un período desde principios del siglo XVI hasta la actualidad.

Todos los conjuntos de datos fueron corregidos automáticamente o, cuando fue preciso, manualmente, para corresponder a las directrices de transcripción CATMuS, disponibles aquí: <https://catmus-guidelines.github.io/>.

Las anotaciones en el conjunto de datos resultan de la extracción de diseño, extracción de líneas, escritura y transcripción de los creadores originales del conjunto de datos en la mayoría de los casos, o de correcciones automáticas o manuales realizadas por el curador del conjunto de datos moderno de CATMuS.

Este modelo fue entrenado en el conjunto de datos McCATMuS, con Kraken v.4.3.13, con normalización NFD Unicode y un tamaño de lote de 32 en 157 épocas.

Transcritas las líneas con diferentes OCR, usamos estas salidas para combinarlas intentando recuperar la mayor cantidad de texto posible, corregir errores ortográficos y dar coherencia usando un LLM.

El LLM utilizado es **Gemini**, principalmente por motivos económicos. Como diccionario de frecuencias se empleó spanish frequency dictionary, pues la creación de un diccionario a partir de obras antiguas resultaba muy limitada. El modelo de **spaCy** usado fue **es_core_news_sm**.

Gemini adopta una arquitectura *transformer* basada en autoatención, de forma similar a los LLM modernos. Además, se ha entrenado en un corpus mixto multilingüe que incluye español. Aunque no se especializa en español antiguo, su sólido manejo del idioma y su facilidad de uso resultaron decisivos.

Por contraste, GPT-2 se centra mayoritariamente en inglés y requiere optimizaciones adicionales; T5 (incluso en sus variantes **base** o **large**) demanda más recursos computacionales, encareciendo su uso continuo.

Por ello, se optó por **Gemini** como un equilibrio entre rendimiento y compatibilidad con el español, un factor esencial al manejar gran volumen de documentos y requerir un ciclo iterativo de corrección.

Para la recopilación de datos se comenzó con el dataset de Rodrigo, que abarca español antiguo pero tipografía gótica (no procesal-cortesana). Luego se recurrió al corpus CODEA, con documentos anteriores a 1900 que incluyen imágenes, transcripciones paleográficas y versiones críticas. Tras filtrar por el tipo de letra requerida, el total se redujo a 546 documentos, lo cual demuestra la dificultad de obtener datos adecuados.

3. Resultados

3.1. Logros Principales

En la primera etapa se logra una mejora considerable de las imágenes, aplicando un pipeline de limpieza que las convierte a escala de grises e incluye varias técnicas de filtrado (filtro gaussiano para reducir ruido, ecualización de histograma para mejorar contraste, y operaciones morfológicas de erosión y dilatación para resaltar los trazos). Finalmente, se emplea la binarización de **Kraken**, con redes neuronales profundas y binarización adaptativa entrenada en documentos históricos. En la figura 3 se observa la evolución desde la imagen original hasta la binarizada.

Para tratar las manchas producidas por la antigüedad (tonos amarillos), se implementa un conversor personalizado a escala de grises que pondera r y g de forma distinta a la habitual, atenuando así el efecto amarillo.

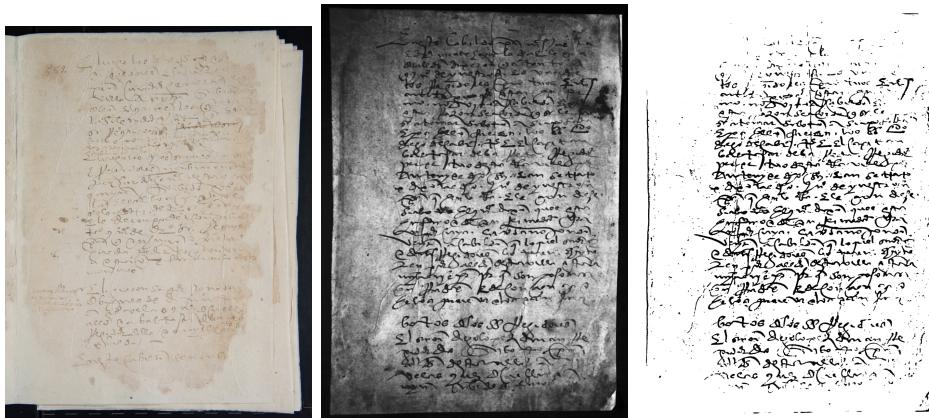


Figura 3: Original, en escala de grises con letra reforzada, e imagen binarizada.

También se evalúan algoritmos de detección de bordes como *sobel*, *Canny* y *laplaciano*. El último no funciona bien por su sensibilidad a cambios drásti-

cos pixel a pixel (papel vs. tinta). Sobel y Canny muestran algunos buenos resultados en ciertos documentos, como se aprecia en la figura 4, pero no de modo uniforme.

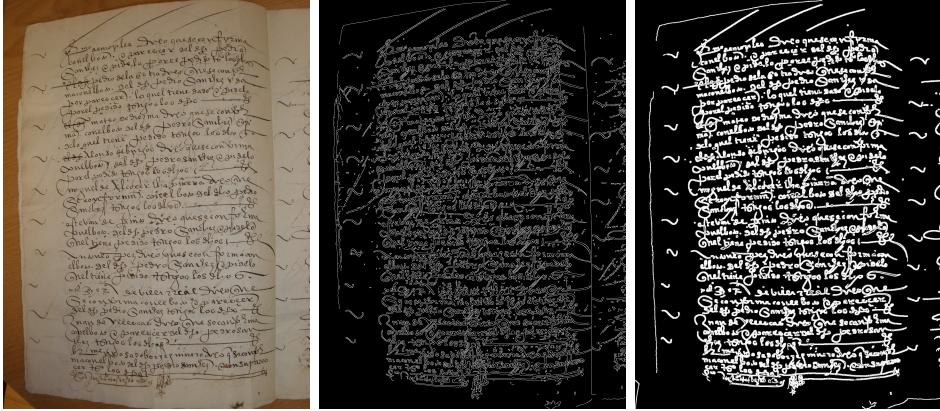


Figura 4: Original, detección de bordes y operaciones morfológicas (de izquierda a derecha).

También se prueba el filtrado mediano para reducir el ruido “sal y pimienta” y el filtrado bilateral, sin mejoras significativas. La binarización mediante un umbral fijo no detecta bien la letra en zonas de sombra, y la binarización adaptativa convencional recupera demasiado ruido.

Tras la binarización, las imágenes pasan por dos procesos de segmentación: uno interno de **Kraken** y otro basado en proyección de histogramas. El primero ofrece resultados superiores. Para segmentaciones manuales se investiga **SAM2**, sin lograr automatización completa. Dicha herramienta es efectiva con líneas rectas, algo que no se cumple en manuscritos antiguos. Resultado obtenido hasta el momento por **SAM2se** muestra en las figuras 5 y 6.

Finalmente, la imagen segmentada se procesa con tres OCR distintos: **SimpleHTR**, **sinai-sam-rec-v4-best** y **McCATMuS-nfd-nofix-V1**. Los dos últimos aprovechan **Kraken**, mientras que **SimpleHTR** requiere una segmentación previa en líneas.

3.2. Posprocesamiento y Combinación

Después de extraer el texto con los distintos OCR, se procede a un posprocesamiento para mejorar la fidelidad de las transcripciones. Inspirándose en el dataset de CODEA, se adopta una transcripción crítica de los documentos. Dada la complejidad de los manuscritos y los errores de OCR, se implementa un pipeline que corrige errores ortográficos y refina la coherencia.

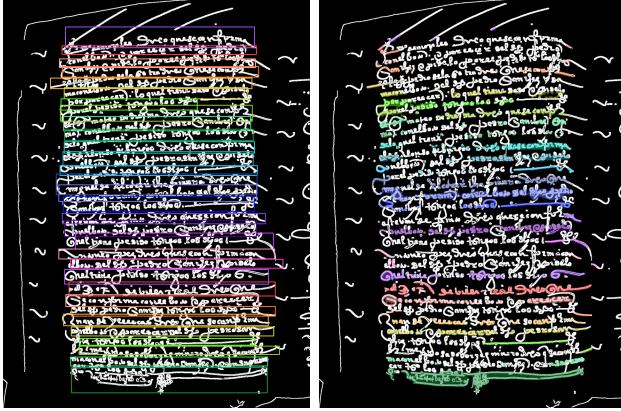


Figura 5: Imagen original con boxes aplicados manualmente, y la imagen segmentada a partir de estos

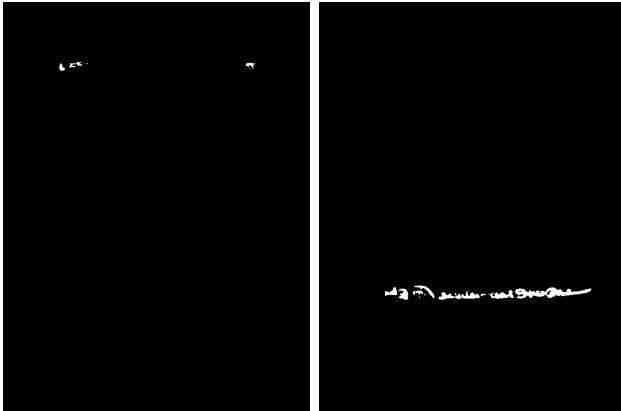


Figura 6: Imágenes de varias segmentaciones

En primer lugar, se usa spaCy para segmentar el texto en *tokens*, aislando palabras y símbolos no alfabéticos. Cada palabra se corrige con SymSpell, que se apoya en un diccionario de frecuencias para separar palabras unidas y reponer grafías correctas. Luego, un modelo generativo (**Gemini**) refina la semántica y el estilo, ayudando a mantener un tono cercano al español antiguo.

Combinación de OCR con el LLM. Para unificar las salidas de los tres OCR (SimpleHTR, sinai-sam-rec-v4-best y McCATMuS-nfd-nofix-V1), se emplea Gemini con un enfoque de “prompt colaborativo”. Este recibe:

- Texto A: salida del OCR 1
- Texto B: salida del OCR 2
- Texto C: salida del OCR 3

Se le indica que compare las tres versiones, identifique coincidencias y divergencias, corrija inconsistencias ortográficas o gramaticales, y produzca una transcripción unificada fiel al original histórico. Gracias a la base en español y la capacidad generativa, **Gemini** pondera cada fragmento, compensando errores de un OCR con aciertos de otro.

Por ejemplo:

1. Essta es una prueba de ectraccion de teexto. Connoscida cosa sea a todos los queesta carta uieren como yo don Fferrando por la gracia de dios hey de Castiella
2. Esta es una prueba de extracción de texto. Conocida cosa sea a todos los que esta carta vieren como yo don Fferrando por la gracia de Dios hay de Castilla
3. Esta es una prueba de extracción de texto. Conocida cosa sea a todos aquellos que esta carta vieran, como yo, don Fernando, por la gracia de Dios, rey de Castilla.

3.3. Desafíos Superados

Uno de los principales retos se observa en la segmentación, que precisa ajustar parámetros para optimizar los siguientes pasos.

Además, la conservación del español antiguo complica el posprocesamiento. Al no existir un diccionario de frecuencias específico, se recurre a un LLM que, si bien no es perfecto, ofrece una aproximación plausible. Se evalúan GPT-2, google/mt5-small y flax-community/spanish-t5-small, pero fracasan por limitaciones de recursos o mala adaptación al dominio. Finalmente, **Gemini** cumple con la exigencia, equilibrando precisión y costo de uso.

4. Conclusiones

4.1. Impacto del Proyecto

Este proyecto se concibe para la transcripción de documentos históricos resguardados en la Oficina del Historiador. Inicialmente se cuenta con un tomo digitalizado, y a medida que se digitalicen los siguientes, se podrán aplicar directamente estas técnicas de transcripción y análisis. Asimismo, los resultados pueden servir de base para productos reutilizables, como herramientas que busquen entidades o información valiosa en estos manuscritos.

4.2. Recomendaciones para Futuras Implementaciones

Se sugiere explorar el uso de SAM2 en la segmentación de imágenes. También se recomienda entrenar o afinar (*fine-tuning*) SimpleHTR y el mo-

delo generativo para mejorar la precisión en español antiguo. Por último, GPT-2 podría reutilizarse entrenándose específicamente en este dominio, lo que potencialmente derivaría en buenos resultados.

Referencias

Referencias

- [1] Vicente Marcet Rodriguez, *El copus de documentos de Ávila del Hispanic Museum and Library (siglos XV y XVI). Descripción y análisis paleográfico y gráfico-fonológico*.
- [2] A. Granet, E. Morin, H. Mouchère, S. Quiniou, y C. Viard-Gaudin, *Transfer Learning for Handwriting Recognition on Historical Documents*, ICPRAM, 2018.
- [3] S. Torres Aguilar, *Handwritten Text Recognition for Historical Documents using Visual Language Models and GANs*, ArXiv, 2025.
- [4] E. Granell, E. Chammas, L. Likforman-Sulem, C. D. Martínez-Hinarejos, C. Mokbel, y B. I. Cîrstea, *Transcription of Spanish Historical Handwritten Documents with Deep Neural Networks*, Journal of Imaging, vol. 4, no. 15, 2018.
- [5] THI TUYET HAI NGUYEN, ADAM JATOWT, MICKAEL COUSTATY and ANTOINE DOUCET, *Survey of Post-OCR Processing Approaches*, 2025.
- [6] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, Haoshuang Wang, *PP-OCR: A Practical Ultra Lightweight OCR System*
- [7] Prof. Anuradha Thorat, Mayur Zagade, Shivani More, Manish Pasalkar, Anand Narute, *Research Paper on Text Extraction using OCR*, IJARSCT, Volume 3, Issue 14, May 2023
- [8] Ojas Kumar Barawal, and Dr Yojna Arora, *Text Extraction from Image*, IJIREM, Volume 9, Issue 3, May 2022