



Reporte Técnico de los Experimentos y Actividades Realizadas por Estudiantes Involucrados en el Proyecto

Optimización y Desarrollo de Métodos de Inteligencia Computacional Aplicados a la Solución de Problemas en Biomedicina

CIIC 232/2019

Responsable: Dr. Luis Carlos Padierna García

Introducción

En este documento se proporciona la información de los estudiantes involucrados y las actividades y experimentos que cada uno de ellos realizó. En resumen, se involucraron 6 estudiantes de diversos programas de licenciatura ofrecidos en la División de Ciencias e Ingenierías (Ingeniería Biomédica, Licenciatura en Física y Licenciatura en Ingeniería Física).

Contenido

Reporte de Actividades de la Estudiante 1 (Karla Angélica Doctor Mauricio).....	2
Reporte de Actividades del Estudiante 2 (Edwin Armando Bedolla Montiel).....	4
Reporte de Actividades de la Estudiante 3 (Martha Susana García Delgado).....	6
Reporte de Actividades del Estudiante 4 (Walter Adame González)	10
Reporte de Actividades del Estudiante 5 (Abdiel Beltrán Aranda).....	13
Reporte de Actividades del Estudiante 6 (Joel Isai Sánchez Morán)	16



Reporte de Actividades de la Estudiante 1 (Karla Angélica Doctor Mauricio)



Nombre	Karla Angelica Doctor Mauricio
Programa Educativo	Licenciatura en Ingeniería Física
NUA	426883
Proyecto	Optimización y Desarrollo de Métodos de Inteligencia Computacional Aplicados a la Solución de Problemas en Biomedicina.
Programa	Convocatoria Institucional de Investigación Científica
Monto total recibido por concepto de beca	\$ 5500.00 (cinco mil quinientos pesos 00/100 M.N.)

A continuación se describe el listado de actividades que realicé durante el periodo Enero-Diciembre 2019. Cada actividad descrita se relaciona con la evidencia codificada por KADM-E#, donde # es un número consecutivo.

1. Creación de base de datos de Datasets de interés (1 - 14 ene 2019, evidencia KADM-E1)

Se revisaron en diversas bases de datos los datasets más representativos de diversas áreas. Esta revisión tuvo como propósito la familiarización con repositorios y entendimiento de la composición de datasets.

2.- Lectura de MSV (14 ene – 14 feb 2019, no genera evidencia)

Las MSV se encuentran entre los mejores (y muchos creen que son los mejores) algoritmos de aprendizaje supervisado "listos para usar", se basan en la teoría estadística de aprendizaje. En el aprendizaje supervisado, la máquina de aprendizaje recibe un conjunto de ejemplos de entrenamiento (o entradas) con etiquetas asociadas (o valores de salida). Por lo general, los ejemplos tienen la forma de vectores de atributos, de modo que el espacio de entrada es un subconjunto de R^n (**Murty & Raghava, 2016**). El modelo más simple de MSV, que también fue el primero en ser introducido, es el llamado clasificador de margen máximo. Solo funciona para datos que son linealmente separables en el espacio de las características, y por lo tanto no se pueden usar en muchas situaciones del mundo real. La introducción de kernels aumenta enormemente el poder expresivo de las máquinas de aprendizaje, al tiempo que conserva la linealidad subyacente que asegurara que el aprendizaje sea manejable. El aumento de la flexibilidad, sin embargo, aumenta el riesgo de sobreajuste a medida que la elección de la separación del hiperplano se vuelve cada vez más infructuosa debido a la cantidad



de grados de libertad. Para lo cual se utilizan herramientas como validación cruzada. (Cristianini & Shawe-Taylor, 2013)

2. Análisis de funcionamiento de LIBSVM en C++ (14 feb – 29 feb 2019, evidencia KADM-E2)

Después de descargar de <http://www.csie.ntu.edu.tw/~cjlin> (Lin, s.f.) la carpeta con todos los archivos un paso natural fue comprender como funcionaba el algoritmo de máquinas de soporte vectorial con el solver LIBSVM para trabajar con el lenguaje C++ y poder realizar futuras modificaciones, nos encontraremos estos archivos, svm.h (Contiene todas las clases y métodos implementados en svm.cpp), svm.cpp (archivo principal en donde están implementadas las máquinas de soporte y los kernels), svm.o (hace más eficaz la ejecución), svm_scale (para reescalar los datos), svm_train (crea los nodos), svm_predict (entrena al modelo), svm.def (únicamente se encarga del almacenamiento de atributos y funciones) y los archivos makefile.

3. Implementación de MSV en libsvm (29 feb – 25 may 2019, evidencia KADM-E3, KADM-E4)

Como consideraciones previas para aplicar las MSV se escalan los datos de los dataset utilizados en KADM-E3 se muestra uno de estos, lo cual nos da como principal ventaja evitar atributos dominando rangos muy pequeños, además de evitar dificultades numéricas durante el cálculo, esto debido a que los valores del kernel generalmente dependen de los productos internos de los vectores de características (Hsu, Chang, & Lin, s.f.). Seguido de esto se codificó para después de haber cargado los datos en svm_node “llamar a la función” svm_problem, svm_model y svm_predict KADM-E4. Después de que todo se probó que funcionaba se modificó la svm.h y svm.cpp para poder añadir el kernel s-Hermite y comprobar la supuesta ventaja reportada de este kernel en cuanto a Accuracy y número de vectores soporte.

4. Implementación de MSV en Python (8 ago -13 ago 2019, evidencia KADM-E5)

Después del entendimiento de MSV se implementó el algoritmo en este lenguaje para visualizar la frontera generada con el kernel de s-Hermite con el objetivo de entender cuál es la razón por la cual tiene mejor rendimiento que otros kernels KADM-E5.

5. Actividad 1 (15 ago 2019, evidencia KADM-E6)

Se presentó una conferencia en formato tipo póster en la tercera sema de la ingeniería biomédica organizado en la división de ciencias e ingenierías en León, Guanajuato. Se mostró un avance en la investigación relacionada con la diabetes en la cual se incluye un grid search para obtener los parámetros más adecuados para el kernel s-Hermite y una validación cruzada a 10 pliegues en KADM-E6 se puede ver el poster presentado.

6. Actividad 2 (5 - 7 sep 2019, evidencia KADM-E7, KADM-E8)

Se presentó una conferencia en formato tipo Póster en San José, Costa Rica, en el evento internacional LATInTY. En la conferencia se mostró una aplicación de máquinas de soporte vectorial con kernels ortogonales para clasificación de Diabetes Mellitus tipo 2 y la frontera de clasificación con el kernel s-Hermite. La evidencia KADM-E7 es el reconocimiento obtenido por la ponencia y reconocimiento KADM-E8.

7. Revisión de estado del arte (17 – 21 dic 2019, evidencia KADM-E9)

Se hizo una investigación acerca de los algoritmos implementados en la última década para diagnosticar y predecir diabetes, así como patologías asociadas como retinopatía diabética, daños en la macula, sensibilidad en pie diabético, entre otros. Esta investigación se puede leer en KADM-E8.

DEPARTAMENTO DE INGENIERÍAS QUÍMICA, ELECTRÓNICA Y BIOMÉDICA
DIVISIÓN DE CIENCIAS E INGENIERÍAS

Loma del Bosque #103, Col. Lomas del Campestre. León, Gto. C.P. 37150
Tel. +52 (477) 7 88 51 00 ext. 8522
lc.padierna@ugto.mx
www.dci.ugto.mx/depinguim



REFERENCIAS

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (s.f.). *A Practical Guide to Support Vector Classification*. Obtenido de <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

Cristianini, N., & Shawe-Taylor, J. (2013). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Lin, C.-J. (s.f.). *Support Vector Machine Solvers*. Obtenido de <https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html>

Murty, M., & Raghava, R. (2016). *Support Vector Machines and Perceptrons*. Springer International Publishing.

Reporte de Actividades del Estudiante 2 (Edwin Armando Bedolla Montiel)



Nombre	Edwin Armando Bedolla Montiel
Programa Educativo	Licenciatura en Física
NUA	426783
Proyecto	Repositorio de algoritmos de Machine Learning e Inteligencia Computacional para la división de Ciencias e Ingenierías de la Universidad de Guanajuato.
Programa	Convocatoria Institucional de Investigación Científica
Participación	Becario bajo Servicio Social Componente 2
Monto total recibido por concepto de beca	\$ 4000.00 (cuatro mil pesos 00/100 M.N.)

A continuación se describe el listado de actividades que realicé durante el periodo Enero-Junio 2019. Cada actividad descrita se relaciona con la evidencia codificada por EABM-E#, donde # es un número consecutivo.

1. Lectura sobre Reconocimiento de Patrones clásico (20 - 27 ene 2019, No genera evidencia)

Se revisaron los principales métodos de reconocimiento de patrones clásico [1] como lo son el clasificador de mínima distancia, el clasificador bayesiano y los fundamentos de la clasificación en el reconocimiento de patrones. Esta revisión tenía como propósito conocer los elementos estadísticos y rudimentos necesarios para ahondar en el análisis de datos, reconocimiento de patrones y posteriormente, Deep Learning.



2. Implementación de métodos de Inteligencia Computacional (1 feb - 30 abr 2019, evidencia EABM-E1)

Una vez se conocieron los elementos principales de la Inteligencia Computacional se procedió a realizar implementaciones de los siguientes métodos de **clasificación supervisada**: clasificador de mínima distancias, clasificador bayesiano [1], *k-Nearest Neighbors* [3], clasificación con Máquinas de Soporte Vectorial [4]. Se implementaron los siguientes métodos de **clasificación no supervisada**: *k-Means* y modelos de mezclas [3]. Sobre **regresión**, se emplearon los siguientes métodos: regresión simple, regresión de Ridge, regresión de tipo Lasso [6] y regresión mediante Máquinas de Soporte Vectorial [5]. El método de análisis de componentes principales [3] también se implementó como parte del **procesamiento de datos**.

Por último, para la parte de **inteligencia artificial** se generaron ejemplos y pruebas de concepto sobre redes neuronales y perceptrones multicapa [7]. En la evidencia EABM-E1, que consiste del siguiente repositorio en línea <https://github.com/edwinb-ai/intelicompu>, se pueden revisar todas estas implementaciones codificadas en el lenguaje de programación Python, disponibles para todo público y con licencia de software libre para toda persona interesada en utilizarlo pueda utilizarlo sin restricción de costo.

3. Paquete de software sobre segmentación semántica para imágenes médicas (18 ago - 27 oct 2019, evidencia EABM-E2)

La segmentación semántica consta de un tipo de clasificación en una imagen donde existen muchos objetos y se quieren separar unos de otros. De esta forma, la segmentación semántica resuelve un problema clave en el *Deep Learning*, y esto es que esta clasificación se pueda hacer de forma automática y precisa. En este trabajo se escribió un paquete de software en el lenguaje de Python, probado rigurosamente con datos reales y ayuda de expertos en el tema. Este paquete consta de dos implementaciones de modelos muy robustos en el área de segmentación semántica, la U-Net [2] y la MultiResUNet [8]. Estas implementaciones siguen muy de cerca las especificaciones de los artículos originales por lo que se considera que este paquete es de gran ayuda para la comunidad científica. Estas implementaciones constan de modelos de *Deep Learning*, técnicas de ingeniería de software, programación orientada a objetos y procesamiento de datos biomédicos. La evidencia EABM-E2 consta del siguiente repositorio <https://github.com/DCI-NET/segmed>, que también puede ser visitado con libertad y sin costo, donde se encuentra todas las implementaciones con su respectiva documentación.

REFERENCIAS

- [1] Sá, J. M. D. (2012). *Pattern recognition*. n.p.: Springer Science & Business Media.
- [2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention, pages 234{241, 2015. arXiv:1505.04597[cs.CV].
- [3] Hastie, T., Tibshirani, R., & Friedman, J. (2013). *The elements of statistical learning*. n.p.: Springer Science & Business Media.
- [4] Vladimir Vapnik. The nature of statistical learning theory. Springer science & business media, 2013.
- [5] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. Statistics and computing, 14(3):199–222, 2004.
- [6] John Fox et al. Robust regression. An R and S-Plus companion to applied regression, page 91, 2002.

DEPARTAMENTO DE INGENIERÍAS QUÍMICA, ELECTRÓNICA Y BIOMÉDICA
DIVISIÓN DE CIENCIAS E INGENIERÍAS

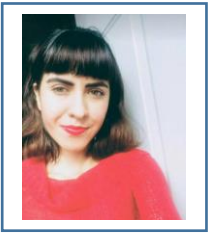
Loma del Bosque #103, Col. Lomas del Campestre. León, Gto. C.P. 37150
Tel. +52 (477) 7 88 51 00 ext. 8522
lc.padierna@ugto.mx
www.dci.ugto.mx/depinguim



[7] C.C. Aggarwal. Neural Networks and Deep Learning: A Textbook. Springer International Publishing, 2018.

[8] Ibtehaz, Nabil, and M. Sohel Rahman. "MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation." Neural Networks 121 (2020): 74-87

Reporte de Actividades de la Estudiante 3 (Martha Susana García Delgado)



Nombre	Martha Susana García Delgado
Programa Educativo	Licenciatura en Ingeniería Biomédica
NUA	758629
Proyecto	Optimización y Desarrollo de Métodos de Inteligencia Computacional Aplicados a la Solución de Problemas en Biomedicina
Programa	Convocatoria Institucional de Investigación Científica
Participación	Becario bajo Servicio Social componente 2
Monto total recibido por concepto de beca	\$3000.00 (tres mil pesos)

A continuación, se describe el listado de actividades que realicé durante el periodo Enero-junio 2019.

1. Lecturas acerca de Neurogénesis y procesamiento digital de imágenes (21 de ene- 25 enero del 2019).

Se revisaron varios artículos y libros para conocer el proceso de Neurogénesis y así como el procesamiento de imágenes microscópicas

2. Revisión del estado del arte (28 de enero-22 de febrero del 2019, evidencia MSGD-E1)

El análisis de las imágenes de microscopía puede realizarse tanto manualmente como automáticamente. Pero ya que las células son elementos vivos y los procesos celulares son normalmente estocásticos, los análisis deben realizarse masivamente en cientos o miles de células para un determinado experimento con el fin de que sean resultados estadísticamente significativos. El análisis manual de las imágenes de microscopía puede resultar tedioso y susceptible a errores. El error más común que se puede cometer en un análisis manual es el error de tendencia entre-ensayos o entre analistas. Por consiguiente, los métodos de diagnósticos ayudados por métodos computacionales mejoran la reproducibilidad y permiten una elevada velocidad de procesamiento de imágenes han atraído gran interés por parte de la comunidad científica y médica. Es por ello por lo que se mantiene la necesidad de crear y mejorar algoritmos automáticos para procesar y segmentar imágenes microscópicas de cortes de cerebro de ratón adulto que permitan obtener regiones de interés (o al menos un acercamiento inicial de la misma que luego pueda ser modificada por el experto) y características de morfología, alineación, etc., de células B que se encuentran en la zona

DEPARTAMENTO DE INGENIERÍAS QUÍMICA, ELECTRÓNICA Y BIOMÉDICA
DIVISIÓN DE CIENCIAS E INGENIERÍAS

Loma del Bosque #103, Col. Lomas del Campestre. León, Gto. C.P. 37150
Tel. +52 (477) 7 88 51 00 ext. 8522
lc.padierna@ugto.mx
www.dci.ugto.mx/depinguim



subventricular del proceso de Neurogénesis. El conocer bien las células que forman parte de este proceso se pueden hacer rastreos más precisos y realizar una mejor clasificación de cada una de las células que forman parte del proceso.

Todos los datos e imágenes con los cuáles se trabajaron fueron tomados del artículo “**Visualization and correction of automated segmentation, tracking and lineaging from 5-D stem cell image sequences**” Eric Wait, Mark Winter, Chris Bjornsson. En el artículo mencionan que se analizaron un total de 18 secuencias de imágenes 5-D que muestran células madre neurales de ratones adultos, células endoteliales y vasculatura. Las células madre fueron visualizadas dentro de un cultivo explante en 3D de la zona subventricular (SVZ) del cerebro. Cada ubicación de vóxel 5- D se especifica como (x, y, z, t, λ) donde λ se refiere a una señal de fluorescencia multicanal, NSCs de un canal de imágenes, el segundo canal que contiene células endoteliales y vasculatura. Las películas fueron capturadas con dos microscopios en dos laboratorios diferentes.

3. Métodos de segmentación (25 de febrero- 17 mayo del 2019. evidencia MSGD-E2)

Se usó el lenguaje de programación Python, gracias a la cantidad de librerías que existen para el procesamiento de imágenes y además que contiene cientos de algoritmos de visión artificial, además de ser multiplataforma tiene una gran eficiencia computacional para trabajar con aplicaciones en tiempo real. Una de las partes principales del proyecto es la segmentación inicial de la imagen mediante el método de Otsu

4. INTERPRETACION Y DISCUSION DE RESULTADOS (17 de mayo- 5 de julio del 2019, evidencia MSGD-E3)

A continuación, se muestra algunas imágenes del repositorio “Susan_overnight.zip” previamente segmentadas con su respectivo histograma de intensidad. Cabe mencionar que sólo se tomaron las imágenes del canal dos, ya que es la zona de interés (Células B)

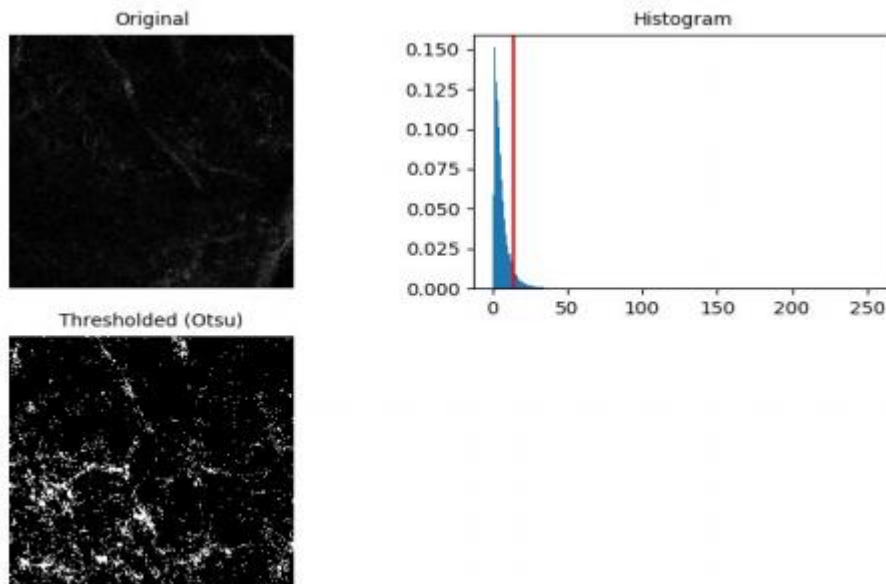




Figura 1. Imagen "Susan_overnight_c02_t0048_z0015.tif". Se muestra la imagen original, el umbral de Otsu y su histograma de intensidad.

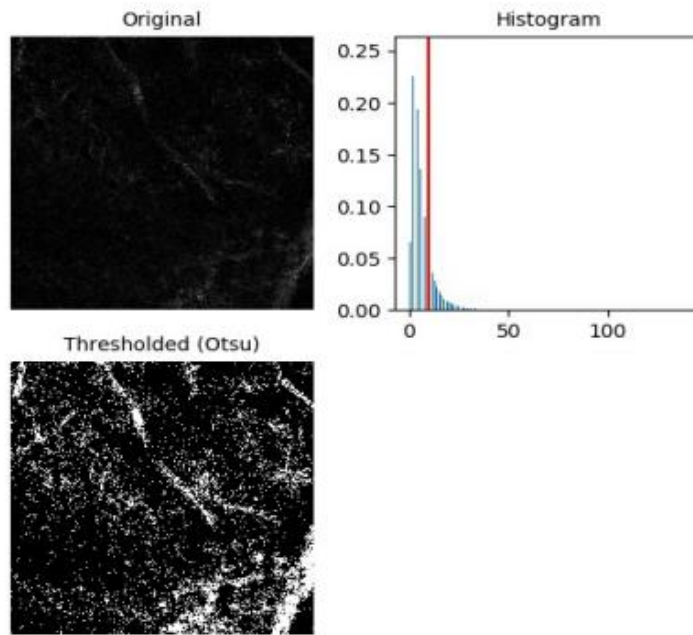


Figura 2. Imagen "Susan_overnight_c02_t0003_z0013.tif". Se muestra la imagen original, el umbral de Otsu y su histograma de intensidad.

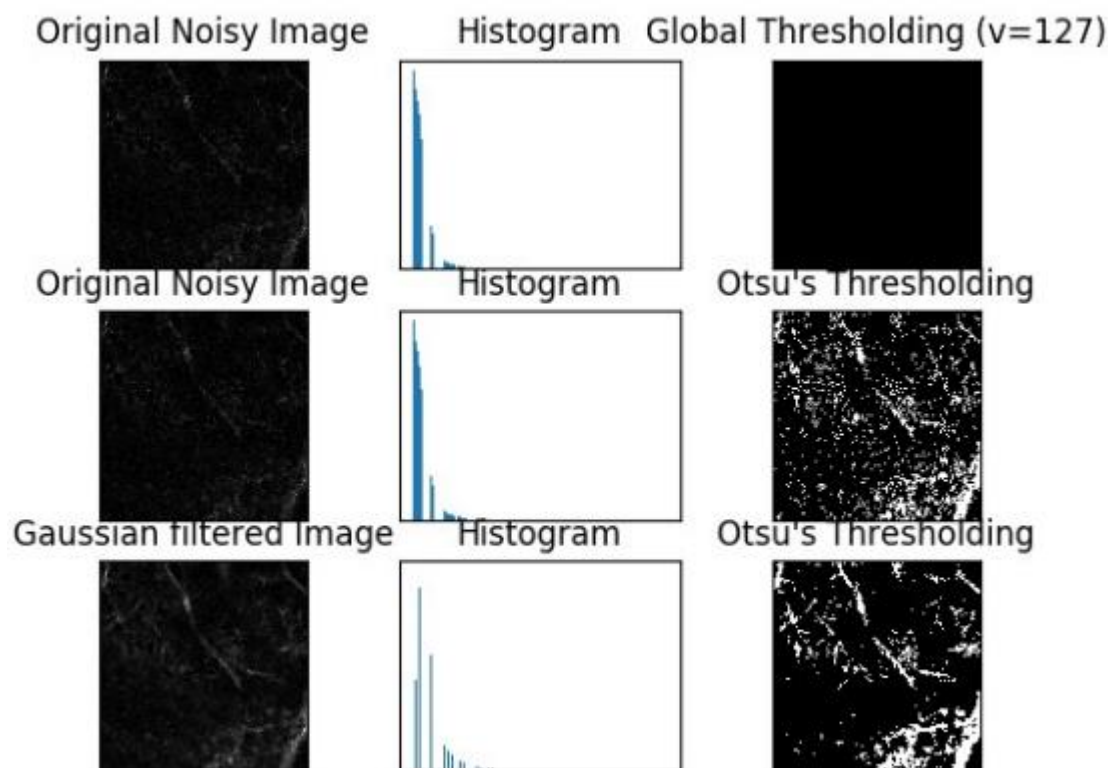


Figura 3. Imagen “Susan_overnight_c02_t0003_z0013.tif” En el primer caso se aplicó una umbralización global para un valor de 127, en el segundo caso, se aplicó el método de Otsu de forma directa. En el tercer caso se filtró la imagen con un kernel Gaussiano de 5x5 para remover ruido, luego se aplicó la umbralización de Otsu. Se observa como el filtro que remueve el ruido mejora los resultados.

REFERENCIAS

- [1] E. Wait, M. Winter and A. R. Cohen. “Hydra image processor: 5-D GPU image analysis library with MATLAB and python Wrappers” *Bioinformática*, volumen 35, número 24, 15 de diciembre de 2019, páginas 5393-5395, <https://doi.org/10.1093/bioinformatics/btz523>
- [2] Shen Q, Wang Y, Kokovay E, Lin G, Chuang SM, Goderie SK, Roysam B. “Temple S: Adult SVZ stem cells lie in a vascular niche: a quantitative analysis of niche cell-cell interactions”
- [3] Kirsten Obernier, Arantxa Cebrian-Silla, Matthew Thomson, ..., José Rodas Rodriguez, José - Manuel García-Verdugo, Arturo Alvarez -Buylla. “Adult Neurogenesis Is Sustained by Symmetric Self Renewal and Differentiation”
- [4] Edgar Cárdenas De La Hoz, Mark R. Winter, María Apostolopoulou, Sally Temple, and Andrew R. Cohen. “Measuring Process Dynamics and Nuclear Migration for Clones of Neural Progenitor Cells”
- [5] Arnold Kriegstein and Arturo Alvarez-Buylla.” *The Glial Nature of Embryonic and Adult Neural Stem Cells*”

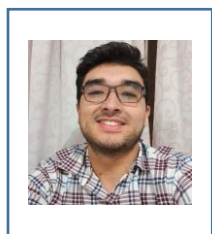
DEPARTAMENTO DE INGENIERÍAS QUÍMICA, ELECTRÓNICA Y BIOMÉDICA
DIVISIÓN DE CIENCIAS E INGENIERÍAS

Loma del Bosque #103, Col. Lomas del Campestre. León, Gto. C.P. 37150
Tel. +52 (477) 7 88 51 00 ext. 8522
lc.padierna@ugto.mx
www.dci.ugto.mx/depinguim



- [6] Kirsten Obernier and Arturo Alvarez-Buylla. *"Neural stem cells: origin, heterogeneity and regulation in the adult mammalian brain"*
- [7] Cohen AR, Bjornsson C, Temple S, Banker G, Roysam B. *Automatic summarization of changes in biological image sequences using algorithmic information theory*, Transactions on Pattern Analysis & Machine Intelligence, volumen 31, Agosto de 2009, páginas 1386-1403, doi: 10.1109/TPAMI.2008.162
- [8] Winter M, Wait E, Roysam B, Goderie S, Kokovay E, Temple S, Cohen AR. *Vertebrate neural stem cell segmentation, tracking and lineaging with validation and editing*. Revista, *Nature Protocols*, volumen 6, diciembre del 2011, páginas 1942–1952, doi:10.1038/nprot.2011.422
- [9] Mankowski WC, Winter M, Wait E, Lodder MJ, Schumacher TN, Naik SH, Cohen AR. *Segmentation of occluded hematopoietic stem cells from tracking*. Conference: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.
- [10] Walter T, Shattuck DW, Baldock R, Bastin ME, Carpenter AE, Duce S, Ellenberg J, Fraser A, Hamilton N, Pieper S, Ragan MA, Schneider JE, Tomancak P, Heriche JK. *Visualization of image data from cells to organisms*. Nat Methods 7, S26–S41 (2010) doi:10.1038/nmeth.1431
- [11] Ravishankar Chityala, Sridevi Pudipeddi (2004): *Image Processing and Acquisition using Python*. CRC press.

Reporte de Actividades del Estudiante 4 (Walter Adame González)



Nombre	Walter Adame González
Programa Educativo	Licenciatura en Ingeniería Biomédica
NUA	389007
Proyecto	Optimización y Desarrollo de Métodos de Inteligencia Computacional Aplicados a la Solución de Problemas en Biomedicina
Programa	Convocatoria Institucional de Investigación Científica
Participación	Becario bajo servicio social componente 2
Monto total recibido por concepto de beca	\$0.00 (cero pesos 00/00 M.N.)

A continuación, se describe el listado de actividades que realicé durante el periodo Enero-Junio 2019. Cada actividad descrita se relaciona con la evidencia codificada por WAG-E#, donde # es un número consecutivo.

DEPARTAMENTO DE INGENIERÍAS QUÍMICA, ELECTRÓNICA Y BIOMÉDICA
DIVISIÓN DE CIENCIAS E INGENIERÍAS

Loma del Bosque #103, Col. Lomas del Campestre. León, Gto. C.P. 37150
Tel. +52 (477) 7 88 51 00 ext. 8522
lc.padierna@ugto.mx
www.dci.ugto.mx/depinguim



1. Búsqueda de artículos científicos para el estudio (no genera evidencias).

Se hizo la búsqueda de artículos científicos del estado del arte y de otros artículos relacionados con el objetivo del proyecto:

1. Applications of Support Vector Machine (SVM) Learning in Cancer Genomics. Shujung Huang, et. al. Cancer Genomics & Proteomics. 2007.
2. Support vector machine applications in bioinformatics. Byvatov E, Schneider G. Applied Bioinformatics. 2003.
3. A Clustering Method Based on K-Means Algorithm. Youguo Li, Haiyan Wu. Physics Procedia Volume 25, 2012.
4. An effective and efficient hierarchical K-means clustering algorithm. Jianpeng Qi, et. al. International Journal of Distributed Sensor Network. 2017.
5. Mahalanobis Distance. G. J. McLachlan. Resonance. 1999.

2. Implementación del método K-Means en Python (evidencia WAG-E1)

Se realizó el análisis y la implementación del método de K-Means para la clasificación de muestras de una población partiendo del conocimiento de a cuántos grupos pertenecen. Se utiliza la información de los centros de cada grupo para calcular de mejor manera a qué clasificación pertenece la muestra. En la evidencia WAG-E1 se muestra una captura de pantalla de la implementación del código de clusterización mediante K-means, así como de un ejemplo trabajando.

3. Obtención de imágenes biológicas para su posterior conteo mediante métodos computacionales (evidencia WAG-E2)

Se realizó la obtención de algunas imágenes biológicas para su análisis y conteo por medio de métodos computacionales. Se implementarán técnicas de conteo en MatLab para su posterior comparación con los métodos de máquinas de soporte vectorial.

Se obtuvieron 50 imágenes en total de este tipo para su posterior análisis y comparación entre métodos computacionales propuestos.

4. Aplicación de métodos de conteo celular en imágenes biológicas (evidencia WAG-E3)

A las imágenes biológicas obtenidas se aplicaron técnicas para conteo celular. Se desarrollaron los códigos y se aplicaron los métodos obteniendo los resultados reportados en la evidencia WAG-E3.

Se lograron aplicar técnicas tradicionales de conteo celular de imágenes biológicas resolviendo también el problema de clusterización sin perder información de los píxeles en la región del cluster.



Se implementaron técnicas utilizando segmentación estadística usando distancias de Mahalanobis, además de operaciones morfológicas y de transformada Watershed para el conteo de regiones con problemas de clusterización.

5. Divulgación de resultados obtenidos (evidencia WAG-E4)

Se llevaron a cabo eventos para la divulgación de los resultados y la capacitación en las metodologías utilizadas para obtener dichos resultados con estudiantes de semestres iniciales cursando materias de programación orientada a objetos.

Se discutieron los procedimientos para la obtención de las imágenes biológicas de sangre, las características de una buena imagen, los procedimientos para la segmentación de las imágenes, ya que esto es un paso intrínseco del conteo mediante métodos computacionales tradicionales. Se resolvieron al final un conjunto de dudas e inquietudes que surgieron en los estudiantes participantes.

REFERENCIAS (No incluidas en las evidencias)

1. Dg, D. (2017, 07 26). *WHITE BLOOD CELLS MORPHOLOGY*. Retrieved from Bioscience.com: <https://www.bioscience.com.pk/topics/hematology/item/800-white-blood-cells-morphology>
2. Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2004). *Digital Image Processing Using MatLab*. New Jersey: Pearson Education, Inc.
3. Instituto Mexicano del Seguro Social. (2018). *Diagnóstico y Tratamiento de Leucemia Linfoblástica Aguda en el Adulto. Guía de Referencia Rápida Guía de Práctica Clínica*. Ciudad de México: Coordinación Técnica de Excelencia Clínica. Coordinación de Unidades Médicas de Alta Especialidad.
4. Jmarchn. (21 de 12 de 2017). *Hematopoyesis*. Recuperado el 04 de 01 de 2020, de Wikipedia: <https://es.wikipedia.org/wiki/Hematopoyesis>
5. McLachlan, G. J. (Junio de 1999). Mahalanobis Distance. *Resonance*, 20-26.
6. Meyer, F. (1994, July). Topographic distance and watershed lines. *Signal Processing*, 38, 113-125.
7. Piech, C. (2013). *K Means*. Recuperado el 07 de 01 de 2020, de Stanford.edu: <https://stanford.edu/~cpiech/cs221/handouts/kmeans.html>



Reporte de Actividades del Estudiante 5 (Abdiel Beltrán Aranda)



Nombre	Abdiel Ulises Beltrán Aranda
Programa Educativo	Licenciatura en Ingeniería Física
NUA	343474
Proyecto	Optimización y Desarrollo de Métodos de Inteligencia Computacional Aplicados a la Solución de Problemas en Biomedicina
Programa	Convocatoria Institucional de Investigación Científica
Participación	Servicio Social Segundo Componente
Monto total recibido por concepto de beca	\$ 0.00 (cero pesos 00/100 M.N.)

A continuación se describe el listado de actividades que realicé durante el periodo agosto-diciembre 2019. Cada actividad descrita se relaciona con la evidencia codificada por AUBA-E#, donde # es un número consecutivo.

0. Lectura sobre Deep Learning (20 ago - 23 ago 2019, No genera evidencia)

Se revisaron los primeros cuatro capítulos del libro *"Deep Learning"* [1], los cuales incluyen matemáticas básicas, así como el capítulo 9, el cual cubre el concepto de redes convolucionales.

1. Revisión del estado del arte (26 ago - 30 ago 2019, evidencia AUBA-E1)

Se buscaron artículos sobre segmentación semántica de imágenes biomédicas publicados a partir del año 2015, cuando publicó el artículo de la arquitectura U-Net [2]. Los artículos consultados comparan sus resultados con la U-Net al ser el modelo base de las arquitecturas. En la evidencia AUBA-E1 se dan más detalles acerca de la estructura de la U-Net, así como la idea principal de las arquitecturas subsecuentes.

2. Implementación U-Net en Python (02 sep - 06 sep 2019, evidencia AUBA-E2)

Se encontraron dos implementaciones en internet de la U-Net, en el lenguaje Python, ambas cuentan con un repositorio de GitHub. La primera tiene un conjunto de imágenes de prueba [8] y la segunda es una versión en Python de la U-Net en R que aparece en la página de Keras [9]. En la evidencia AUBA-E2 se muestra un cuaderno de Python en el cual se probó la primera implementación. Se decidió conservar solo el modelo de la U-Net sin añadir los módulos para hacer aumento de datos.



3. Reproducción de resultados: segmentación de células blancas en muestra de sangre (09 sep – 04 de oct, evidencia AUBA-E3)

Se leyó un artículo sobre segmentación semántica de células blancas en muestras de sangre [11] en el cual se propone un algoritmo no supervisado para realizar la segmentación y se comparan resultados contra la arquitectura U-Net. Se implementaron las métricas en Python, con las cuales se mide el rendimiento de los resultados, y se hicieron pruebas para comparar los resultados utilizando los parámetros referidos en el artículo. El código se corrió en Colaboratory de Google [12], en la evidencia AUBA-E3 se dan detalles acerca de las métricas así como el respectivo código de Python.

4. Optimización de métricas y primeras pruebas (07 oct – 01 nov, evidencia AUBA-E4)

Se modificó la implementación de las métricas para disminuir el tiempo de cómputo. Se realizaron varias pruebas, con el conjunto de 100 imágenes, utilizando validación cruzada (utilizando los parámetros reportados [11]), con distinto número de épocas. Posteriormente se realizó validación cruzada con el conjunto de 300 imágenes. En la evidencia AUBA-E4 se añade un reporte donde se muestran y discuten los resultados obtenidos, se añade una libreta de Python donde se pueden consultar los detalles de la implementación, las carpetas con los respectivos modelos entrenados y archivos con los valores de las métricas.

5. Corrección de métricas (04 – 15 nov, evidencia AUBA-E5)

Se detectó que las métricas no toman en cuenta el número de imágenes que se evalúan en el modelo en cada iteración del entrenamiento por lo que se modificaron y se realizó el entrenamiento con validación cruzada nuevamente en ambos conjuntos de imágenes. Se compararon los resultados con los obtenidos anteriormente. En la evidencia AUBA-E5 se presentan las correcciones hechas, así como los resultados obtenidos. Se añaden las carpetas con los modelos entrenados y los archivos con los valores de las métricas.

6. Instalación de software necesario para ejecutar los algoritmos en PC (18 – 29 nov, evidencia AUBA-E6)

Para medir los tiempos de ejecución y predicción adecuadamente se optó por ejecutar los algoritmos en una PC con hardware controlado, puesto que en Colaboratory la asignación de GPU no es controlada. Se instaló la versión 2.0 de TensorFlow y el software necesario para utilizar un GPU Nvidia GTX 1070 Ti; sin embargo, aunque la instalación se realizó correctamente, hay un problema con la alimentación del PC por lo que se reinicia al intentar correr las pruebas ya hechas en Colaboratory. En la evidencia AUBA-E6 se pueden consultar los detalles de la instalación.

7. Pruebas con imágenes del grupo (01 – 20 dic, evidencia AUBA-E7)

Se utilizaron imágenes de células blancas en muestras de sangre propias del departamento de Ingeniería Biomédica de la División de Ciencias e Ingenierías, Campus León, de la Universidad de Guanajuato para corroborar el funcionamiento de la U-Net. Posteriormente se entrenaron nuevos modelos de la arquitectura

DEPARTAMENTO DE INGENIERÍAS QUÍMICA, ELECTRÓNICA Y BIOMÉDICA
DIVISIÓN DE CIENCIAS E INGENIERÍAS

Loma del Bosque #103, Col. Lomas del Campestre. León, Gto. C.P. 37150
Tel. +52 (477) 7 88 51 00 ext. 8522
lc.padierna@ugto.mx
www.dci.ugto.mx/depinguim



utilizando el nuevo conjunto de datos; sin embargo, no se cambiaron los parámetros reportados en el artículo de Zheng y sus colaboradores [11]. En la evidencia AUBA-E7 se anexa un reporte con los resultados y detalles del entrenamiento, carpetas con las segmentaciones obtenidas con cada modelo, así como los modelos entrenados y una discusión de los resultados.

REFERENCIAS (No incluidas en las evidencias)

- [1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
<http://www.deeplearningbook.org>.
- [2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-Assisted Intervention, pages 234{241, 2015.
arXiv:1505.04597[cs.CV].
- [3] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. Fourth International Conference on 3D Vision (3DV), 2016.
arXiv:1606.04797[cs.CV].
- [4] X. Xia and B. Kulis. W-net: A deep model for fully unsupervised image segmentation.
- [5] W. Wang, K. Yu, J. Hugonot, and P. Fua. Recurrent u-net for resource-constrained segmentation. ArXiv, 2019. arXiv:1906.04913.
- [6] Z. Alom, C. Yakopcic, T. Taha, and V. Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation.
- [7] Z. Gu, J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao, T. Zhang, and S. Gao adn J. Liu.Ce-net: Context encoder network for 2d medical image segmentation. IEEE Transactions on Medical Imaging, page 10, 2019. <http://dx.doi.org/10.1109/TMI.2019.2903562>.
- [8] Unet using tensorow and keras. <https://github.com/zizhaozhang/unet-tensorflow-keras>. _Ultimo acceso: 2019-11-30.
- [9] Kaggle carvana image masking challenge solution with keras. <https://github.com/petrosgk/Kaggle-Carvana-Image-Masking-Challenge>. _Ultimo acceso: 2019-11-30.
- [10] A. Gulli and S. Pal. Deep Learning with Keras: Implement neural networks with Keras on Theano and TensorFlow. Packt Publishing, Birmingham, UK, 2017.
- [11] X. Zheng, Y. Wang, G. Wang, and J. Liu. Fast and robust segmentation of cell images by selfsupervised learning. Micron, pages 55,71, 2018. <https://doi.org/10.1016/j.micron.2018.01.010>.
- [12] Google colabory. <https://colab.research.google.com/>. _Ultimo acceso:2019-11-30.



[13] A. Fandango. Mastering TensorFlow 1.x: Advanced machine learning and deep learning concepts using TensorFlow 1.x and Keras. Packt Publishing, Birmingham, UK, 2018.

Reporte de Actividades del Estudiante 6 (Joel Isaí Sánchez Morán)



Nombre	Joel Isaí Sánchez Morán
Programa Educativo	Licenciatura en Física
NUA	427083
Proyecto	Optimización y Desarrollo de Métodos de Inteligencia Computacional Aplicados a la Solución de Problemas en Biomedicina
Programa	Convocatoria Institucional de Investigación Científica
Participación	Becario bajo servicio social universitario componente 1
Monto total recibido por concepto de beca	\$ 3000.00 (tres mil pesos 00/100 M.N.)

A continuación se describe el listado de actividades que realicé durante el periodo Enero-Diciembre 2019. Cada actividad descrita se relaciona con la evidencia codificada por JISM-E#, donde # es un número consecutivo.

1. Lectura sobre Deep Learning (20 ago - 23 ago 2019, No genera evidencia)

Se revisaron los primeros cuatro capítulos del libro Deep Learning [1], los cuales incluyen matemáticas básicas, así como el capítulo 9, el cual cubre el concepto de redes convolucionales.

2. Implementación de algoritmo PSO (17 mayo - 21 mayo 2019, evidencia JISM-E1)

Se implementó un algoritmo conocido como Particle Swarm Optimization (PSO) en C++. Este algoritmo es una metaheurística que minimiza o maximiza funciones objetivo. Esta implementación genera además una animación que muestra el funcionamiento del algoritmo. Aunque el código principal está escrito en C++, la animación se genera con Python. Para poder utilizar ambos lenguajes, se utilizó la librería "Pybind11" para envolver los códigos de C++. La evidencia JISM-E1 consiste en un directorio con los códigos de implementación. Se pueden encontrar más detalles del algoritmo en [2].

3. Creación de Interfaz gráfica (01 nov 2019- 23 ene 2020, evidencia JISM-E2)

Se creó una interfaz gráfica que muestra algunos proyectos hechos por integrantes del grupo DCI-Net. Esta interfaz permite generar de manera sencilla las animaciones del algoritmo PSO y también tiene algunas herramientas de preprocesamiento que generan archivos de tipo libsvm (a partir de archivos csv) y que validan

DEPARTAMENTO DE INGENIERÍAS QUÍMICA, ELECTRÓNICA Y BIOMÉDICA
DIVISIÓN DE CIENCIAS E INGENIERÍAS

Loma del Bosque #103, Col. Lomas del Campestre. León, Gto. C.P. 37150
Tel. +52 (477) 7 88 51 00 ext. 8522
lc.padierna@ugto.mx
www.dci.ugto.mx/depinguim



matrices gramianas. La evidencia JISM-E2 consiste en un directorio con los códigos de la implementación y con el manual de usuario.

4. Asesorías estudiantes de Programación Básica (02 sep - 22 nov 2019, evidencia JISM-E3)

Durante el semestre dí asesorías a estudiantes del curso de Programación Básica. Las asesorías eran de tres horas a la semana y en ellas se repasaban los conceptos vistos en clase que no habían sido comprendidos por completo. Además se realizaban ejercicios que ayudaban a reforzar los temas. En caso de que alguno de los asesorados tuviera problemas con alguna tarea o ejercicio, se le ayudaba a encontrar una manera en la que pudiera realizar el ejercicio y utilizar lo aprendido para problemas similares. La evidencia JISM-E3 consiste en un directorio con algunos de los códigos que se realizaron durante las asesorías.

5. Implementación de Red Neuronal para señales ABR (14 sep - 15 sep 2019, evidencia JISM-E4)

Se implementó una red neuronal en la plataforma Google Colaboratory, utilizando la librería Keras. Esta red neuronal recibe un arreglo de numpy con la señal a analizar. Este arreglo se consigue a través de la librería wfdb, que recomienda la base de datos de donde se obtuvo el dataset. Por el momento no se tienen las etiquetas con el valor correcto que debería devolver la red neuronal, por lo tanto, se utilizaron etiquetas generadas aleatoriamente para comprobar que la red neuronal está funcionando correctamente. Se pueden encontrar más detalles de la implementación con Keras en [3]. La evidencia JISM-E4 es un archivo de texto que contiene el link a la libreta de Google Colaboratory.

6. Envoltura de códigos (22 nov 2019 - 18 ene 2020, evidencia JISM-E5)

Durante el semestre se envolvieron códigos de C++ y de FORTRAN para poder ser utilizados desde Python. Esto se hizo con el objetivo de resolver algunos problemas de simulación que se encontraban escritos en FORTRAN con herramientas que ya estaban implementadas en Python. Además se creó un script general que muestra el proceso de envoltura para que alumnos de la materia de Programación Orientada a Objeto y Eventos pudieran envolver sus códigos escritos en C++ y crear interfaces gráficas desde Python. La evidencia JISM-E5 es un directorio que contiene un script que envuelve un código de C++ y uno que envuelve código de FORTRAN.

REFERENCIAS (No incluidas en las evidencias)

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, 2016.
<http://www.deeplearningbook.org>.

[2] Venter, G., & Sobieszczanski-Sobieski, J. (2003). Particle swarm optimization. *AIAA journal*, 41(8), 1583-1589.

[3] François, C. (2017). Deep learning with Python.