



**Universidad Nacional de Misiones. Facultad de Ciencias Exactas,  
Químicas y Naturales. Secretaría de Investigación y Postgrado. Maestría  
en Tecnología de la Información.**

Maestranda  
***Yanina Andrea Caffetti***

## **Reconocimiento y clasificación de patrones en imágenes de microscopia, parametrizadas según la Técnica de Micronúcleos**

**Tesis de Maestría presentada para obtener el título de “Magíster  
en Tecnología de la Información”**

“Este documento es resultado del financiamiento otorgado por el Estado Nacional, por lo tanto  
queda sujeto al cumplimiento de la Ley N° 26.899”.

Director  
***Dr. Nelson Acosta***  
Co-Directora  
***Dra. Jacqueline Diana Caffetti***

**Posadas, Misiones 2020**



Esta obra está licenciado bajo Licencia Creative Commons (CC) Atribución-NoComercial-  
CompartirIgual 4.0 Internacional. <https://creativecommons.org/licenses/by-nc-sa/4.0/>

**Universidad Nacional de Misiones**  
**Facultad de Ciencias Exactas, Químicas y Naturales**

**“Tesis de Maestría en Tecnologías de la Información”**

**“Reconocimiento y clasificación de patrones en imágenes  
de microscopia, parametrizadas según la Técnica de  
Micronúcleos”**

**Autor: Yanina Andrea Caffetti**

**Director: Dr. Nelson Acosta**

**Co-Director: Dra. Jacqueline Diana Caffetti**

**Año 2020**

*Dedicatoria:*

*A ustedes, que hacen mi vida bella. Gracias, gracias, gracias: familia, amigos, conocidos,  
desconocidos...*

*Ricardo, Mamá y Papá gracias por todo, gracias por tanto...*

*A mis grandes maestros: Augusto José y Lucio Emanuel Canteros Leyes...*

## Resumen

La presente investigación, analiza diferentes técnicas de clasificación, segmentación y extracción de contornos de imágenes digitales. En particular centra la atención en los algoritmos de Inteligencia Artificial de Facebook, algoritmos evolutivos, algoritmos genéticos, y redes neuronales convolucionales. Se analizará su aplicación a imágenes de microscopía, específicamente aquellas parametrizadas según la técnica de Micronúcleos. La prueba de Micronúcleos es una de las técnicas citogenéticas más empleadas para la evaluación de genotoxicidad y mutagenicidad. Los Micronúcleos surgen a partir de cromosomas completos o fragmentos acéntricos que se retrasan durante la anafase, ya sea por la falta de centrómeros o por daño al huso mitótico. Además, los cambios en la morfología normal de los núcleos, también se consideran indicadores de daño genotóxico y son descritos como anomalías nucleares que constituyen un análisis complementario al recuento de Micronúcleos. El objetivo de este estudio es el diseño de una herramienta bioinformática capaz de colaborar con los profesionales del Laboratorio de Citogenética General y Monitoreo Ambiental de la UNaM-IBS-CONICET para la detección de daños celulares.

*Palabras claves: tecnologías de la información, biotecnología, algoritmo, genética.*

## Abstract

The present investigation analyzes different techniques of classification, segmentation and extraction of contours of digital images. In particular, it focuses on Facebook's Artificial Intelligence algorithms, evolutionary algorithms, genetic algorithms and convolutional neural networks. Its application to microscopy images will be analyzed, specifically those parameterized according to the Micronucleus technique. The Micronucleus test is one of the most commonly used cytogenetic techniques for the evaluation of genotoxicity and mutagenicity. Micronuclei arise from complete chromosomes or acentric fragments that are delayed during anaphase, either due to lack of centromeres or damage to the mitotic spindle. In addition, changes in the normal morphology of the nuclei are also considered indicators of genotoxic damage and are described as nuclear abnormalities that constitute a complementary analysis to the Micronucleus count. The objective of this study is to design a bioinformatic tool capable of collaborating with the professionals of the General Cytogenetics and Environmental Monitoring Laboratory of the UNaM-IBS-CONICET for the detection of cellular damage.

*Keywords: information technologies, biotechnology, algorithm, genetics.*

*Reconocimientos:*

*Al Dr. Nelson Acosta por su predisposición, guía y ayuda en todo momento.*

*A la Dra. Jacqueline D. Caffetti por cada intervención que derivó en un nuevo aprendizaje.*

*A todo el personal que desarrolla sus funciones en el Laboratorio de Citogenética General y Monitoreo Ambiental del Instituto de Biología Subtropical (UNaM-CONICET).*

*Al Dr. Eduardo Zamudio por las sugerencias brindadas.*

*A vos, Ricardo J. Canteros Leyes por cada corrección, por cada detalle, gracias por tanto.*

## Índice

Dedicatoria: .....	II
Resumen .....	III
Índice .....	V
Lista de figuras .....	VII
Lista de Abreviaciones .....	VIII
Capítulo I .....	9
Introducción.....	9
Contexto y justificación.....	12
Objetivos.....	13
Objetivo General: .....	13
Objetivos Específicos: .....	13
Organización de la tesis .....	14
Capítulo II.....	15
Estado del Arte .....	15
Redes Neuronales Convolucionales y métodos basados en delimitación y segmentación de imágenes. ....	16
Arquitecturas CNN y DCNN.....	18
Convolutional Layers .....	18
Pooling Layers .....	20
Fully Connected Layers.....	21
Support Vector Machine (SVM) .....	22
Máscara de Segmentación .....	23
DeepMask.....	24
SharpMask .....	27
FastMask.....	29
MultiPath .....	30

Estructura Foveal.....	31
Omisión de Conexiones.....	31
Función Integral de Pérdida.....	32
Estrategias evolutivas para la detección de contornos y operador gradiente .....	32
Algoritmos Genéticos .....	33
Operador de cruzamiento.....	35
Operador de mutación .....	36
Operador Gradiente .....	37
Combinación de metodologías para la extracción de contornos .....	38
Análisis de imágenes biomédicas .....	39
Capítulo III .....	43
Presentación del Estudio de Caso.....	43
Procesamiento de Imágenes en el estudio de caso abordado.....	44
Procesamiento de Imágenes Digitales .....	44
Especificaciones del Estudio de Caso .....	46
Materiales y Métodos .....	47
Muestreos .....	47
Técnica de Micronúcleos.....	49
Clasificación de Imágenes .....	51
Hardware para digitalización de imágenes.....	54
Tareas desarrolladas .....	54
Capítulo IV .....	57
Herramienta Bioinformática.....	57
Ciclo de Vida del Software en base a Metodologías Agiles.....	58
Especificaciones de Funcionalidad.....	59
Requerimientos Computacionales .....	65
Arquitectura de Software.....	65
Entrenamiento de la DCNN.....	66
Resultados obtenidos .....	66
Capítulo V .....	69
Conclusiones.....	69

Resultados.....	70
Futuras líneas de investigación.....	71
Bibliografía.....	73

## Tabla de Contenidos

### Lista de figuras

<i>Ilustración 1. Descripción general de la arquitectura CNN. Fuente:[35].</i> .....	18
<i>Ilustración 2. Arquitectura AlexNet. Fuente: [24].</i> .....	19
<i>Ilustración 3. Descripción gráfica de una pooling layer. Fuente:[35].</i> .....	21
<i>Ilustración 4. Arquitectura del modelo DeepMask. Fuente: [49].</i> .....	26
<i>Ilustración 5. Arquitectura modelo SharpMask. Fuente: [50].</i> .....	28
<i>Ilustración 6. Una descripción general de la arquitectura FastMask propuesta. Fuente: [57].</i> .....	29
<i>Ilustración 7. Arquitectura MultiPath. Fuente: [58].</i> .....	30
<i>Ilustración 8. Diagrama de Flujo de un Algoritmo Genético. Fuente: [65].</i> .....	35
<i>Ilustración 9. Representación de la aplicación del operador gradiente.</i> .....	38
<i>Ilustración 10. Resumen de publicaciones de revistas basadas en los algoritmos subyacentes de los métodos de detección y segmentación. Fuente: [68].</i> .....	41
<i>Ilustración 11. Espacio de color RGB. Fuente:[74].</i> .....	45
<i>Ilustración 12. Espacio HSV. Fuente:[75].</i> .....	46
<i>Ilustración 13. Fotomicrografías de MN y NA en eritrocitos de las especies de peces analizadas. Fuente: [7].</i> .....	50
<i>Ilustración 14. Fotomicrografías de MN y NA en hemocitos de las especies de moluscos bivalvos analizadas. Fuente:[7].</i> .....	51
<i>Ilustración 15. Limnoperna fortunei, especie de molusco bivalvo .Fuente [78].</i> .....	52
<i>Ilustración 16. S. brevipinna, especie de pez .Fuente: [78].</i> .....	53
<i>Ilustración 17. Esquema Scrum Fuente:[83].</i> .....	58
<i>.Ilustración 18. Tablero Jira</i> .....	59
<i>Ilustración 19. Librerías de inicio</i> .....	60
<i>Ilustración 20. Activación Softmax.</i> .....	60
<i>Ilustración 21. Eritroblastos y eritrocitos de peces para entrenamiento.</i> .....	61
<i>Ilustración 22. Pre-entrenamiento datos “Peces”.</i> .....	62
<i>Ilustración 23. Pre-entrenamiento datos “Moluscos”.</i> .....	63
<i>Ilustración 24. Instancia de segmentación, eritrocitos de peces.</i> .....	64
<i>Ilustración 25. Arquitectura de Software</i> .....	65



## **Lista de Abreviaciones**

CNN = Red neuronal convolucional. (Convolutional Neural Network)

DCNN = Red neuronal convolucional profunda. (Deep Convolutional Neural Network)

MN = Técnica de Micronúcleos.

AN = Anormalidades Nucleares.

# Capítulo I

## *Introducción*

La esencia de este proyecto, tiende a determinar la posibilidad de brindar una solución bioinformática auxiliar al profesional que desempeña sus labores en el Laboratorio de Citogenética General y Monitoreo Ambiental, y de manera específica, al área de Monitoreo Ambiental; a través de la manipulación de imágenes digitales con el fin de caracterizarlas y clasificarlas según la Técnica de Micronúcleos [1].

Una vez que se ha obtenido una imagen digital, mediante la utilización de una cámara digital y microscopio, la primera tarea que se realiza sobre la misma consiste en un pre procesamiento, orientado a mejorar la imagen obtenida, intentando con ello alcanzar resultados óptimos en los procesos subsiguientes.

La segmentación, tarea posterior, consiste en redefinir una imagen de entrada en sus partes constituyentes y objetos. Al terminar la segmentación se obtiene un conjunto de datos en bruto, constituido por el contorno de una región o bien todos sus píxeles significativos.

En cualquier caso se requiere transformar esta información a un formato adecuado para su posterior procesamiento. Particularmente, en ésta investigación, se busca caracterizar la imagen según la Técnica de Micronúcleos.

La última etapa es el reconocimiento de cada una de las regiones obtenidas por segmentación, basándose en la información que proporcionan sus contornos y comparándola a la brindada por la técnica genética aplicada.

Actualmente los algoritmos de reconocimiento de patrones en imágenes digitales, resultan aceptables, en orden a su rendimiento, dependiendo de la evaluación de dos componentes: extracción de características y método de detección. Es decir, la posibilidad de detectar algún patrón específico, sea aislando y segmentando por colores o intensidad de la muestra, realizando una separación de objetos mediante una definición de límites o contornos, o creando máscaras de segmentación de instancias dentro de un objeto; entre otras.

Los trabajos que abordan la segmentación de imágenes por características de color emplean diferentes técnicas [2] pero las de mayor aceptación según sus resultados, son las redes neuronales (RN) [3] y métodos basados en agrupamiento, específicamente, fuzzy c-means (FCM) [4]. Las RN son preparadas para reconocer colores específicos, es decir, que son entrenadas con los colores de la imagen a ser segmentada. Si se da una nueva imagen la RN debe ser entrenada nuevamente.

Otra técnica empleada en la manipulación de imágenes digitales es la extracción de contornos, puesto que colaboran efectivamente en los procesos de segmentación e

identificación de patrones, tanto para tareas de reconocimiento e interpretación, como también de clasificación del objeto.

El operador gradiente es una herramienta ampliamente utilizada para la extracción de contornos. Este permite detectar las variaciones en niveles de intensidad que pudieran corresponder a los contornos de interés. En general, las técnicas utilizadas caracterizan a los contornos de un objeto mediante la detección de sus bordes, en función de discontinuidades importantes entre los niveles de intensidad de pixeles vecinos [5].

El siguiente paso introductorio, es el encuadre de la propuesta en un estudio de caso. El caso como tal refleja una situación, problema o decisión que representa la realidad, eso no significa que sea una narración de algo que ha ocurrido, pero si de algo que podría ocurrir o podría haber ocurrido[6]. El estudio de caso, permite que se desarrollen actividades de análisis, comparación, evaluación y síntesis, pudiendo exponer una solución tecnológica o auxiliar para la resolución del problema que se presente.

El Laboratorio de Citogenética General y Monitoreo Ambiental, analiza el daño genético producidos en organismos acuáticos como consecuencia de la exposición a contaminantes urbanos e industriales, presentes en los ríos y arroyos de la provincia de Misiones y su impacto potencial sobre la salud humana. Actualmente se aplican dos técnicas específicas: el test de Micronúcleos y el Ensayo Cometa, a muestras tomadas de los modelos biológicos en estudio [7] [8] [9]. Los organismos acuáticos son denominados “bioindicadores” y el daño en el ADN de los mismos se detecta mediante las técnicas antes mencionadas, que constituyen “biomarcadores genéticos”. En consecuencia, resultan útiles como señales de alerta temprana en cursos de agua contaminados, siendo su análisis de interés predictivo en evaluaciones del estado de calidad de cursos hídricos y manejo de cuencas [10] [11].

En el test de Micronúcleos, los protocolos aceptados internacionalmente exigen el recuento de 1000 a 2000 células por individuo (considerándose al menos el análisis de 10-15 individuos por cada sitio de muestreo o tratamiento estudiado) [12] [13]. Estas células son clasificadas en distintos subtipos, de acuerdo al daño que presenten, por ejemplo: células normales, células con Micronúcleos, y otro grupo denominado células con alteraciones de la morfología nuclear, que a su vez incluyen: núcleos con lobulaciones, con muescas, con hendiduras, en forma de ocho, con gemaciones, con puentes, binucleadas; entre otras [7]. La posibilidad de plantear una solución tecnológica que realice un recuento de células para analizarlas y luego clasificarlas según su morfología facilitaría el trabajo del profesional en el Laboratorio.

## **Contexto y justificación.**

El Instituto de Biología Subtropical (IBS) es una unidad ejecutora de investigaciones científicas y tecnológicas de doble dependencia. Depende de la Universidad Nacional de Misiones (UNaM) y del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

El IBS tiene dos sedes, una en la ciudad de Posadas, que funciona en dependencias de la Facultad de Ciencias Exactas, Químicas y Naturales, y la otra en la ciudad de Puerto Iguazú, que está relacionada académicamente a la Facultad de Ciencias Forestales, ambas de la UNaM.

Los temas abordados en las investigaciones que se llevan adelante en este instituto son muy diversos, abarcando desde estudios de biología molecular hasta investigaciones sobre comportamiento y ecología, incluyendo como objetos de estudio a las plantas, los animales, los microorganismos y la salud de los seres humanos. Uno de los objetivos del IBS es difundir el conocimiento científico que se genera en el Instituto a la comunidad [14].

El Grupo de Investigación en Citogenética General y Monitoreo Ambiental, ubicado en el nodo de la ciudad de Posadas del IBS, desarrolla sus tareas en Citogenética de Vertebrados desde el año 1989, habiendo incentivado, a través de la formación de recursos humanos, el desarrollo de tesinas de graduación y posgraduación, ofreciendo pasantías en estudios de citogenética en vertebrados: peces, anfibios, reptiles y aves.

Actualmente, el grupo se encuentra abocado a la investigación científica en Citogenética descriptiva y evolutiva de peces y moluscos de agua dulce, y en Mutagénesis y monitoreo ambiental, incluyendo en este campo experimentos con diversos modelos biológicos (peces, moluscos, *Allium cepa* y linfocitos humanos).

El cuerpo de integrantes del grupo de investigación en Citogenética Animal y Monitoreo Ambiental del IBS, cuenta con docentes investigadores, becarios doctorales y postdoctorales, graduados y estudiantes de grado, que realizan estudios enmarcados en diversos campos, proyectos incentivados (SPU), proyectos colaborativos con asociaciones civiles (Centro de Investigación Antonia Ramos, CIAR) y un proyecto de investigación PICTO “Evaluación Citogenético Molecular de Stocks de Especies empleadas en Piscicultura en la Provincia de Misiones”.

El Director del Laboratorio es el Dr. Alberto Sergio Fenocchio, y la presente investigación se lleva a cabo con la Co-Dirección de la Dra. Jacqueline Diana Caffetti, investigadora perteneciente al Grupo de Citogenética Animal y Monitoreo Ambiental, experta en: Proyectos de Investigación relacionados al monitoreo e impacto ambiental de

sistemas acuáticos naturales y ensayos toxicológicos mediante la evaluación de distintos tipos de biomarcadores genéticos y bioquímicos, aplicación de técnicas de genética toxicológica con principal énfasis en el test de Micronúcleos (MN) y ensayo cometa (EC), en diferentes modelos biológicos tanto *in vivo* como *in vitro*; manejo de técnicas de citogenética convencional y bandeos cromosómicos aplicados principalmente en modelos de peces, moluscos y citogenética humana orientada al diagnóstico.

En el área de investigación en Mutagénesis y Monitoreo Ambiental de aguas de ríos y arroyos de la provincia de Misiones, se aplica a las muestras, entre otras técnicas, la de Micronúcleos. Ésta fue desarrollada por W. Schmid en 1975 [1], quien originalmente la propuso para practicarse en médula ósea de ratón. Posteriormente, la misma se ha instrumentado en una gran variedad de tejidos y especies, aplicándose en la actualidad a numerosos modelos biológicos que abarcan desde plantas y animales, hasta seres humanos, tanto *in vivo* como *in vitro* [15].

Los Micronúcleos son fragmentos de cromosomas o cromosomas completos, que espontáneamente o por causa de agentes genotóxicos, quedan fuera del núcleo durante la división celular. Como consecuencia de ello, pueden generarse genotipos potencialmente desbalanceados que se asocian a la transformación celular y al riesgo de desarrollo de cáncer [16] [17].

El análisis de las imágenes producidas por el test de Micronúcleos, puede ser realizado tanto de manera manual como automática, mediante una solución informática. Esta última opción permitirá fundamentalmente imprimir velocidad en el procesamiento y análisis de tales imágenes, siendo la principal justificación del presente trabajo.

## **Objetivos**

### **Objetivo General:**

- Generar un prototipo óptimo para el reconocimiento de patrones de imágenes microscópicas parametrizadas según la técnica de Micronúcleos empleada por el Laboratorio de Citogenética General y Monitoreo Ambiental para la detección de daños celulares.

### **Objetivos Específicos:**

- Evaluar el potencial de las redes neuronales convolucionales (CNN) para generar cuadros de delimitación, máscaras de segmentación y clasificación de objetos en tiempo real.
- Valorar la conveniencia de utilizar las imágenes biológicas microscópicas del estudio de caso abordado, como conjunto de datos de entrada para el

aprendizaje de una CNN, juntamente con la metodología de un algoritmo de segmentación a través de máscaras.

- Evaluar el potencial de los algoritmos evolutivos y la posibilidad de extracciones de contornos mediante grados de semejanza o similitud en el conjunto de imágenes generadas de las muestras microscópicas del caso en estudio, codificando un patrón de interés y clasificándolas en tiempo real.

### **Organización de la tesis**

- Capítulo I, *Introducción*: presentación del tema de tesis. Objetivos y justificación.
- Capítulo II, *Estado del Arte*: documentación perteneciente a investigaciones referidas al procesamiento de imágenes. Se expone qué se ha hecho recientemente sobre dos líneas concretas, la primera acerca de redes convolucionales, métodos basados en delimitación y segmentación de imágenes. La segunda, sobre algoritmos evolutivos, operador gradiente y la utilización de ambas metodologías en la optimización de extracción de contornos en imágenes digitales.
- Capítulo III, *Presentación del Estudio de Caso*: especificaciones de la problemática planteada en el tema de tesis. Test de Micronúcleos y su implementación en el Laboratorio de Citogenética General y Monitoreo Ambiental. Tratamiento de imágenes digitales, descripción de los métodos, técnicas e instrumentos de recolección de datos. Especificaciones de los recursos utilizados.
- Capítulo IV, *Herramienta Bioinformática*: propuesta de solución al estudio de caso proyectado. Presentación de la herramienta diseñada en Lenguaje Python 3.8. Especificaciones del tratamiento de las imágenes. Resultados obtenidos.
- Capítulo V, *Conclusiones*: resultados esperados y futuras líneas de investigación.

# Capítulo II

## *Estado del Arte*



## **Redes Neuronales Convolucionales y métodos basados en delimitación y segmentación de imágenes.**

Generalmente, las redes neuronales artificiales son definidas en base a analogías propias del cerebro humano. De hecho, una red neuronal artificial, presenta similitudes con el funcionamiento y jerarquización de las redes neuronales biológicas, en donde la comunicación de la información se presenta a través de la interconexión de nodos.

En la comunicación de las redes neuronales artificiales, la comunicación es unidireccional y el procesamiento se realiza en la neurona receptora de la información, siendo enviado a la capa de neurona superior de la red. En definitiva, las redes neuronales artificiales, están inspiradas en el comportamiento biológico de las neuronas y en la estructura del cerebro.

La clasificación de imágenes, se puede definir como la tarea de catalogar las imágenes en una de varias clases predefinidas, siendo un problema fundamental que se presenta en la visión de los desarrollos actuales por computadora; sirviendo de base para otras tareas en la manipulación de imágenes, tales como la localización, la detección y la segmentación [18].

Aunque la tarea puede considerarse un proceso naturalizado para los humanos, en un sistema automatizado ésta resulta ser mucho más difícil.

Algunas de las complicaciones encontradas incluyen la variabilidad del objeto dependiente del punto de vista y la alta variabilidad en su clase, al presentar muchos tipos de objetos [19].

Originalmente, se trabajaba con las imágenes de manera artesanal mediante descriptores de características que sirvieron como entrada para un clasificador entrenable. El principal obstáculo de este enfoque se debió a que la precisión de la tarea de clasificación dependía profundamente del diseño de la característica en la etapa de extracción [20].

En los últimos años, se ha demostrado que los modelos de aprendizaje profundo que explotan múltiples capas de procesamiento de información no lineal, para extracción y transformación de características, así como para análisis y clasificación de patrones, superan estos desafíos referidos a la precisión en la tarea de clasificación. Entre ellos, las redes neuronales convolucionales se han convertido en la arquitectura líder para la mayoría de las tareas de reconocimiento, clasificación y detección de imágenes.

El aprendizaje profundo permite que los modelos computacionales que se componen de múltiples capas de procesamiento aprendan representaciones de datos con múltiples niveles de abstracción.

Estos métodos han mejorado dramáticamente el estado del arte en reconocimiento de voz, reconocimiento de objetos visuales, detección de objetos y muchos otros dominios como el descubrimiento de fármacos y la genómica.

El aprendizaje profundo descubre una estructura compleja en grandes conjuntos de datos mediante el uso del algoritmo de retropropagación, para indicar cómo una máquina debe cambiar sus parámetros internos, aquellos que se utilizan para calcular la representación en cada capa a partir de la representación en la capa anterior. Las redes convolucionales profundas han logrado avances en el procesamiento de imágenes, video, voz y audio, mientras que las redes recurrentes han iluminado datos secuenciales como el texto y la voz [21].

El aprendizaje profundo generalmente se refiere a métodos que mapean datos a través de múltiples niveles de abstracción, donde los niveles más altos representan entidades más abstractas. El objetivo es que un algoritmo aprenda automáticamente funciones complejas que mapean entradas a salidas, sin usar características o reglas hechas a mano [22].

El avance más significativo que ha capturado un intenso interés en las redes neuronales convolucionales de aprendizaje profundo (DCNN), especialmente para las tareas de clasificación de imágenes, se logró en el ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 [23], al utilizarse una DCNN para clasificar aproximadamente 1.2 millones de imágenes en 1000 clases, con resultados récord [24].

Desde entonces, las DCNN han dominado las versiones posteriores de la ILSVRC y, más específicamente, su componente de clasificación de imágenes. Además, se lograron ejemplos representativos de otros intentos de mejora relacionados con los siguientes aspectos diferentes de DCNN:

- Arquitectura de red [25] [26] [27].
- Funciones de Actividad No Lineal [28] [29].
- Componentes de Supervisión [30] [31].
- Mecanismos de Regulación [32] [25].
- Técnicas de Optimización [33].

Las redes basadas en modelos de aprendizaje profundo, han sido ampliamente adoptadas para la resolución de problemas referidos a la manipulación de datos en formato imagen.

En particular, la red entrenada por Alex Krizhevsky, popularmente llamada "AlexNet" se ha utilizado y modificado para varios problemas de clasificación [34].

### Arquitecturas CNN y DCNN

En la descripción general de una arquitectura como se especifica en la Ilustración 1, la imagen ingresa directamente a la red, y esto es seguido por varias etapas de convolución y agrupación. Posteriormente, las representaciones de estas operaciones alimentan una o más capas completamente conectadas. Finalmente, la última capa completamente conectada genera la etiqueta de clase [35].

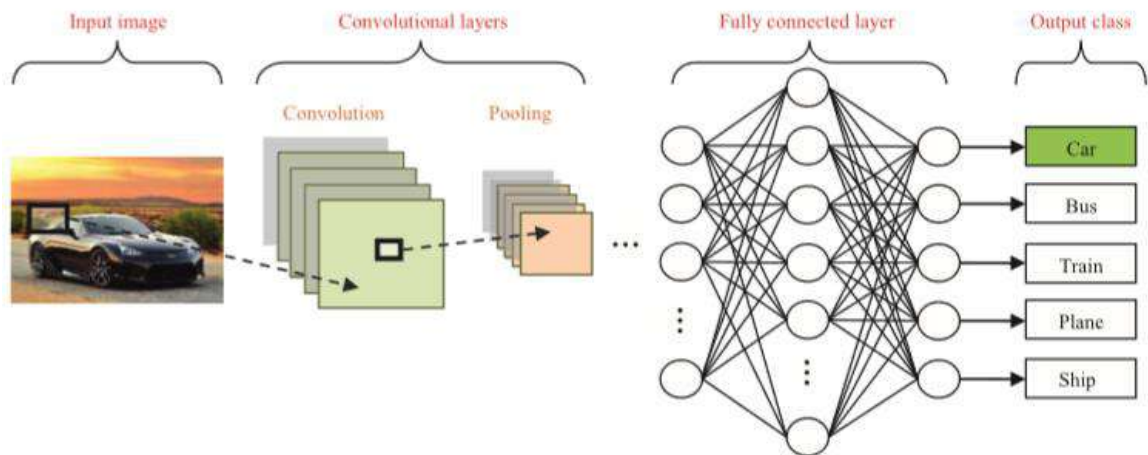


Ilustración 1. Descripción general de la arquitectura CNN. Fuente:[35].

### Convolutional Layers

Las capas convolucionales sirven como extractores de características, y así aprenden las representaciones características de las imágenes de entrada. Las neuronas en las capas convolucionales están organizadas en mapas de características. Cada neurona en un mapa de características tiene un campo receptivo, que está conectado a un vecindario de neuronas en la capa anterior mediante un conjunto de pesos entrenables, a veces denominado banco de filtros [21].

Las entradas están argumentadas con los pesos aprendidos para calcular un nuevo mapa de características y los resultados se envían a través de una función de activación no lineal. Todas las neuronas dentro de un mapa de características tienen pesos que están

obligados a ser iguales; sin embargo, los diferentes mapas de características dentro de la misma capa convolucional tienen pesos diferentes para que varias características se puedan extraer de cada ubicación [20] [21].

Podría calcularse el  $k$ -ésimo mapa de características de salida  $y_k$ , mediante la siguiente expresión matemática:

$$y_k = f(w_k * x)$$

La imagen de entrada es denotada por  $x$ ;  $w_k$  es el filtro convolucional relacionado con el mapa de características iniciales de la imagen; el signo de multiplicación se refiere al operador convolucional 2D que se usa para calcular el producto interno del modelo de filtro en cada ubicación de la imagen de entrada  $y$ ;  $f(.)$  representa la función de activación no lineal [36].

Los métodos de aprendizaje profundo apuntan a aprender jerarquías de características con particularidades de niveles superiores de la jerarquía formados por la composición de características de nivel inferior. Las funciones de aprendizaje automático en múltiples niveles de abstracción permiten que un sistema aprenda funciones complejas que mapean la entrada a la salida directamente de los datos, sin depender completamente de las características creadas por el hombre. Esto es especialmente importante para las abstracciones de nivel superior, dado que los humanos a menudo no saben cómo especificar explícitamente en términos de información sensorial en bruto. La capacidad de aprender automáticamente características potentes será cada vez más importante a medida que la cantidad de datos y el rango de aplicaciones para los métodos de aprendizaje automático continúen creciendo [22].

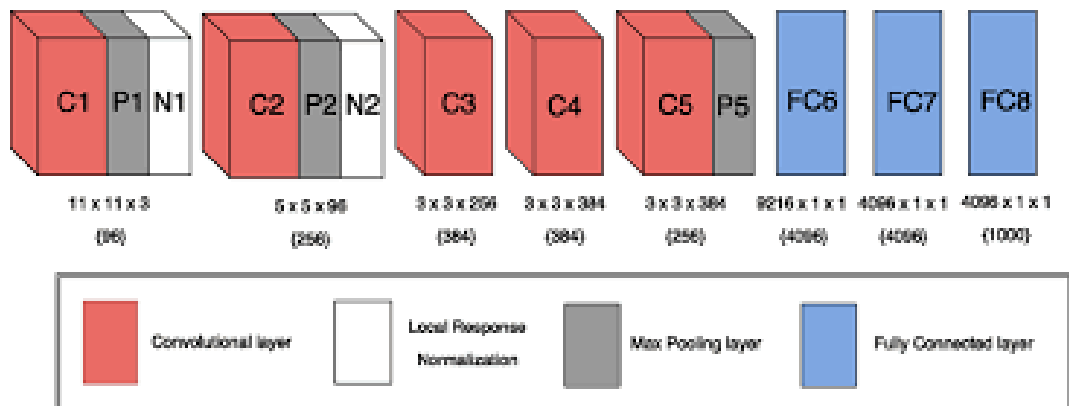


Ilustración 2. Arquitectura AlexNet. Fuente: [24].

En la Ilustración 2, se diagrama la arquitectura de la denominada “AlexNet” [24], es una red convolucional entrenada a partir de los 1.2 millones de imágenes de alta resolución del concurso ImageNet LSRVRC-2010. Con estos datos de prueba, logra tasas de error de top 1 y top 5 de 37,5% y 17,0%, que es considerablemente mejor que el estado del arte anterior[24]. Se pueden visualizar los pesos en la Ilustración 2 y, después de cada capa, hay una no linealidad ReLU implícita. El número dentro de las llaves representa el número de filtros con las dimensiones mencionadas en la parte superior, debajo del diseño de las capas [34]. La red neuronal tiene 60 millones de parámetros y 650.000 neuronas, consta de cinco capas convolucionales, algunas de las cuales son seguidas por capas de agrupación máxima (Max Pooling layer) y tres capas completamente conectadas con un softmax final de 1000 vías. Para acelerar el entrenamiento, “AlexNet” utiliza neuronas no saturantes y la implementación por medio de GPU.

Para la competencia ILSVRC-2012 se presentó una variante del modelo original, logrando una tasa de error de prueba entre los 5 mejores del 15,3%, en comparación con el 26,2% alcanzado por la segunda mejor entrada [24].

### **Pooling Layers**

El propósito de las capas de agrupación es reducir la resolución espacial de los mapas de características y así lograr invariancia espacial para introducir distorsiones y traducciones [20] [21]. Inicialmente, era una práctica común usar capas de agregación de agrupación promedio para propagar el promedio de todos los valores de entrada, de un pequeño vecindario de una imagen a la siguiente capa [20]. Sin embargo, en modelos más recientes [19] [24] [37] las capas de agregación de agrupación máxima propagan el valor máximo dentro de un campo receptivo a la siguiente capa [38]. Formalmente, la agrupación máxima selecciona el elemento más grande dentro de cada campo receptivo de manera que:

$$Y_{kij} = \max_{(p,q) \in R_{ij}} x_{kpq}$$

De tal manera, la salida de la operación de agrupación asociada con el mapa de características -ésimo, se denota por  $Y_{kij}$ ;  $X_{kpq}$  denota la ubicación del elemento  $(p, q)$  contenida en la región de agrupación  $ij$ , que incorpora un campo receptivo alrededor de la posición  $(i, j)$  [36].

Para ser más precisos, se puede pensar que una capa de agrupación consiste en una cuadrícula de unidades de agrupación separadas por  $s$  píxeles, cada una de las cuales resume un vecindario de tamaño  $z \times z$  centrado en la ubicación de la unidad de agrupación.

Dada una imagen de entrada de tamaño  $4 \times 4$ , si se aplica un filtro  $2 \times 2$ , la agrupación máxima coloca el valor máximo de cada región [35], como se ejemplifica en la Ilustración 3.

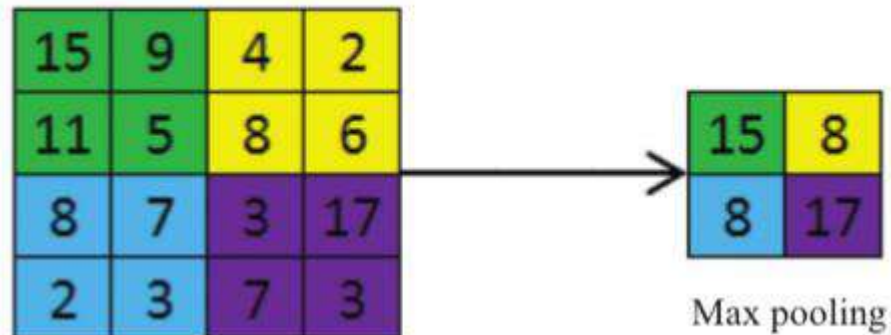


Ilustración 3. Descripción gráfica de una pooling layer. Fuente:[35].

### Fully Connected Layers

Varias capas convolucionales y de agrupación generalmente se apilan unas encima de otras para extraer representaciones de características más abstractas al moverse por la red. Las capas completamente conectadas que siguen a estas capas, interpretan estas representaciones de características y realizan la función de razonamiento de alto nivel [24] [37] [25]. Para problemas de clasificación generalmente se utiliza el operador Softmax en la cima de la DCNN [24] [29]. Investigaciones recientes demuestran que reemplazar el operador softmax con una máquina de vectores de soporte (SVM) conduce a una precisión de clasificación mejorada [30].

### Softmax

Para problemas de clasificación que usan técnicas de aprendizaje profundo, se utiliza básicamente la codificación softmax o 1-of-K, en la parte superior de la red convolucional. Por ejemplo, dadas 10 clases posibles, la capa softmax tiene 10 nodos denotados por  $P_i$ , donde  $i = 1, \dots, 10$ .  $P_i$  especifica una distribución de probabilidad discreta, por lo tanto,  $\sum_i^{10} P_i = 1$ . Sea  $h$  la activación de los penúltimos nodos de capa,  $W$  es el peso que conecta la penúltima capa con la capa softmax, la entrada total en una capa softmax, dada por  $a$ , es [39]:

$$a_i = \sum_k h_k w_{ki}$$

Entonces tenemos [40]:

$$p_i = \frac{\exp(a_i)}{\sum_j \exp(a_j)}$$

La clase predelictiva  $\hat{t}$  sería [41]:

$$\begin{aligned} \hat{t} &= \underset{i}{\operatorname{argmax}} p_i \\ &= \underset{i}{\operatorname{argmax}} a_i \end{aligned}$$

### **Support Vector Machine (SVM)**

Si bien las máquinas de vectores de soporte lineal (SVM) se han utilizado previamente en combinación con las CNN para mejorar el rendimiento de clasificación, la desventaja que presentan es que las características de nivel inferior de la CNN no se aprenden con respecto al objetivo de la SVM. Para abordar el problema se propuso un entrenamiento paralelo en niveles más bajos respectivamente, introduciendo así nuevas funciones de costos. Las SVM se formularon originalmente para la clasificación binaria. Datos de entrenamiento dados y sus etiquetas correspondientes  $(x_n, y_n)$ ,  $n = 1, \dots, N$ ,  $x_n \in R^D$ ,  $t^n \in \{-1, +1\}$ , el aprendizaje de un SVM, consiste en la siguiente optimización[42]:

$$\min_{w, \varepsilon_n} \frac{1}{2} w^T w + C \sum_{n=1}^N \varepsilon_n$$

$$s.t \ w^T x_n t_n \geq 1 - \varepsilon_n \forall_n$$

$$\varepsilon_n \geq 0 \forall_n$$

$\varepsilon_n$  son variables que penalizan los puntos de datos que violan los requisitos de margen. Se puede incluir el sesgo aumentando todos los vectores de datos  $x_n$  con un valor escalar de 1. El problema de optimización correspondiente sin restricciones es el siguiente [43]:

$$\min_w \frac{1}{2} w^t w + C \sum_{n=1}^N \max(1 - W^t x_n t_n, 0)$$

La ecuación anterior se conoce como el problema de la forma primaria de L1-SVM, con la pérdida de bisagra estándar. Dado que L1-SVM no es diferenciable, una variación popular se conoce como L2-SVM que minimiza la pérdida de bisagra al cuadrado [44]:

$$\min_w \frac{1}{2} w^t w + C \sum_{n=1}^N \max(1 - W^t x_n t_n, 0)^2$$

L2-SVM es diferenciable e impone una pérdida mayor (cuadrática versus lineal) para los puntos que violan el margen. Para predecir la etiqueta de clase de un dato de prueba  $x$  [45]:

$$\arg \max_t (w^t x) t$$

### **Máscara de Segmentación**

La segmentación semántica a nivel de instancia o la segmentación de instancias es la tarea de detectar y fraccionar conjuntamente instancias individuales de objetos en una imagen. A diferencia de la detección de objetos, sus salidas son máscaras de segmentación de objetos que representan ubicaciones de objetos y límites, en lugar de solo cuadros delimitadores. A diferencia de la segmentación semántica, su objetivo es asignar no solo la etiqueta de categoría sino también la etiqueta de instancia a cada píxel en la imagen de entrada. En otras palabras, requiere distinguir instancias individuales de la misma categoría además de distinguir diferentes categorías de objetos [46].

El estudio del arte en la actualidad nos determina varios enfoques referidos a la segmentación, en el primero sus métodos generan un conjunto de candidatos a objetos en forma de cuadros delimitadores independientes de la clase, luego predicen una máscara de segmentación de objetos para cada cuadro delimitador y finalmente clasifican los objetos en categorías [47]. Se puede también generar directamente propuestas de objetos en las formas de máscaras de segmentación agnósticas de clase, que luego realice la clasificación de objetos [48].



## **DeepMask**

Es un método innovador para generar propuestas de objetos de segmentación directamente desde los píxeles de la imagen [49]. DeepMask es uno de los métodos representativos de propuesta de objetos basados en CNN, en el que un DCNN aprende a generar máscaras de segmentación de objetos a partir de una imagen de entrada. La parte inferior (capas cercanas a la capa de entrada) de la red se inicializa con una red existente que fue entrenada para realizar la clasificación de imágenes. Esta red pre-entrenada, conocida como body net, es una CNN con todas las capas completamente conectadas eliminadas. La parte superior de DeepMask se divide en dos ramas: una rama genera una máscara de segmentación con una porción de imagen a escala fija, y la otra genera un puntaje de confianza que mide la probabilidad de que un objeto completo esté centrado en dicha porción o parche. Entonces DeepMask sigue el enfoque independiente de la clase, no la categoría de objeto. Después de ser entrenada, la red se puede aplicar densamente en múltiples escalas y ubicaciones en la imagen de entrada para generar todas las máscaras de segmentación de objetos posibles [46].

Si bien DeepMask tiene éxito en la generación de máscaras que representan aproximadamente la ubicación y los límites de los objetos, no es capaz de generar máscaras con precisión de píxeles. Por lo tanto, DeepMask utiliza solo un paso de alimentación a través de la red y genera máscaras de segmentación desde sus capas superiores. Se ha revelado que las primeras capas en una red profunda tienden a capturar características de bajo nivel, como colores y bordes, mientras que las capas superiores tienden a capturar información más abstracta y semánticamente significativa, como la presencia de un objeto. Para resolver este problema, SharpMask [50] aumenta la arquitectura de alimentación hacia adelante (de abajo hacia arriba) de DeepMask, con un proceso de refinamiento de arriba hacia abajo, que combina información de bajo nivel de las primeras capas con información de alto nivel de las capas superiores. Dada una porción de imagen de entrada, el pase de abajo hacia arriba genera una máscara gruesa con baja resolución espacial, que es la entrada del pase de arriba hacia abajo. Múltiples módulos de refinamiento en el paso de arriba hacia abajo depuran iterativamente esa máscara combinándola con características en capas sucesivamente más bajas en la red. Este proceso determina máscaras de segmentación que se adhieren mejor a los límites de los objetos [46].

## **Arquitectura de Red: DeepMask**

Las predicciones de máscara y puntaje se logran con una sola red convolucional. Durante el entrenamiento, las dos tareas se aprenden conjuntamente. En comparación con un modelo que tendría dos redes distintas para las dos tareas, esta opción de arquitectura reduce la capacidad del modelo y aumenta la velocidad de la inferencia de escena completa en el momento de la prueba.

Los parámetros para las capas compartidas entre la predicción de máscara y la predicción de puntuación de objeto son inicializados con una red que fue entrenada previamente para realizar la clasificación en el conjunto de datos [49]. Y el modelo se ajusta para generar propuestas de objetos durante el entrenamiento. La arquitectura de DeepMask se basa en VGG-A [37], que consta de ocho capas convolucionales 3x3 (seguidas de no linealidades ReLU o rectificadoras) y cinco capas de agrupación máxima de 2x2 y ha demostrado un excelente rendimiento. Para inferir máscaras de segmentación, la información espacial proporcionada en los mapas de características convolucionales es importante. Por lo tanto, se eliminan todas las capas finales completamente conectadas del Modelo VGG-A y se descarta la última capa de agrupación máxima [49].

La salida de las capas compartidas tiene un factor de disminución de resolución de 16 debido a las cuatro capas de agrupación máxima 2x2 restantes; dada una imagen de entrada de dimensión  $3 * h * w$ , la salida es un mapa de características de dimensiones:

$$\frac{\frac{512 * h}{16} * w}{16}$$

### **Segmentación DeepMask**

La rama de la red dedicada a la segmentación se compone de una sola capa de convolución 1x1 (y no linealidad ReLU) seguida de una capa de clasificación. Esta capa consta de clasificadores de píxeles  $h \times w$ , cada uno responsable de indicar si un determinado píxel pertenece al objeto en el centro del parche o porción de la imagen. Se destaca que cada clasificador de píxeles en el plano de salida debe poder utilizar la información contenida en todo el mapa de características y, por lo tanto, tener una vista completa del objeto. A diferencia de la segmentación semántica, DeepMask debe generar una máscara para un solo objeto, incluso cuando hay varios objetos presentes.

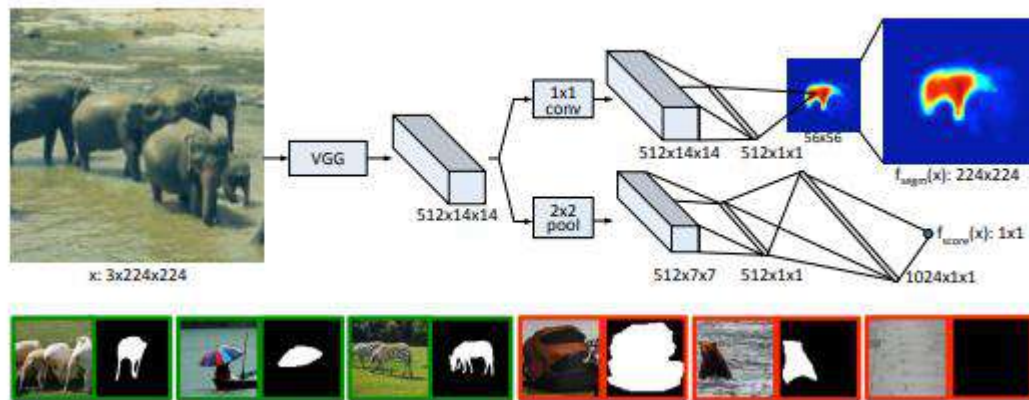


Ilustración 4. Arquitectura del modelo DeepMask. Fuente: [49].

En la Ilustración 4, se puede observar que la red se divide en dos ramas después de la función compartida de capas de extracción. La rama superior predice una máscara de segmentación para el objeto ubicado en el centro, mientras que la rama inferior predice una puntuación de objeto para el parche de entrada. En el sector inferior se presentan ejemplos de tripletas de entrenamiento: entrada parche  $x$ , máscara  $m$  y etiqueta  $y$ . Los parches verdes contienen objetos que satisfacen las restricciones especificadas, y por lo tanto se les asigna la etiqueta  $y = 1$ . Las máscaras para negativo, correspondientes a los ejemplos mostrados en rojo, no se utilizan [49].

DeepMask presenta la capa de clasificación como dos capas lineales sin ninguna iteración en el medio. Las predicciones de máscara y puntaje se logran con una sola red convolucional. Su naturaleza flexible permite diseñar un modelo en el que las dos tareas (predicciones de máscara y puntaje) pueden compartir la mayoría de las capas de la red. Tal enfoque reduce el número de parámetros de red al tiempo que permite que cada clasificador de píxeles aproveche la información del mapa de características completo. Finalmente, se configura la capa de salida como  $h^o \times w^o$  donde  $h^o < h$  y  $w^o < w$ , muestreando la salida a  $h \times w$  para que coincida con las dimensiones de entrada [49].

## **Puntuación DeepMask**

La segunda rama de la red se dedica a predecir si un parche de imagen cumple las restricciones (i) y (ii), es decir, si un objeto está centrado en el parche y en la escala adecuada. Se compone de una capa de agrupación máxima de 2x2, seguida de dos capas completamente conectadas (más la no linealidad ReLU). El resultado final es una

puntuación única de "objetividad" que indica la presencia de un objeto en el centro del parche de entrada (y en la escala adecuada) [49].

## **Aprendizaje**

Dado un parche de entrada  $x_k \in I$  el modelo está entrenado para inferir conjuntamente una máscara de segmentación en píxeles y una puntuación de objeto. La función de pérdida está representada por la suma de pérdidas de regresión logística binaria, una para cada ubicación de la red de segmentación, y otra para la puntuación del objeto, sobre todas las tripletas de entrenamiento (entrada parche  $x$ , máscara  $m$  y etiqueta  $y$ )  $x_k, m_k, y_k$  [49].

$$L(\theta) = \sum_k \left( \frac{1 + y_k}{2w^o h^o} \sum_{ij} \log \left( 1 + e^{-m_k^{ij} f_{segm}^{ij}(x_k)} \right) + \lambda \log \left( 1 + e^{-y_k f_{score}(x_k)} \right) \right)$$

Dónde  $\theta$  es un conjunto de parámetros;  $f_{segm}^{ij}$  es la predicción de la red de segmentación en el lugar  $(i,j)$ ; y  $f_{score}(x_k)$  es la puntuación prevista del objeto. Se alterna entre propagación hacia atrás a través de la rama de segmentación y la rama de puntuación configurando  $\lambda = \frac{1}{32}$ . Para la rama de puntuación, los datos se muestrean de manera que el modelo se entrene con un número igual de muestras positivas y negativas. Sólo se propaga el error si  $y_k = 1$  [49].

## **SharpMask**

La arquitectura DeepMask es la base para la segmentación de instancias de objetos debido a su simplicidad y efectividad, y se le agrega un módulo de refinamiento con el cuál se enfatiza la capacidad de producir máscaras de segmentación de objetos más nítidas y de mayor fidelidad. La arquitectura final se denomina SharpMask [50].

SharpMask mejora la calidad de la máscara de segmentación en relación con DeepMask. Para la generación de propuestas de objetos, la recuperación promedio en el conjunto de datos COCO [51] mejora 1020% y establece el nuevo estado del arte en esta

tarea. Las CNN se han adoptado con gran efecto en las tareas de etiquetado de píxeles, como la estimación de profundidad [52], el flujo óptico [53] y la segmentación semántica [54]. SharpMask es una arquitectura de refinamiento ascendente / descendente a la segmentación de instancia de objeto. Específicamente, genera propuestas de objetos [55] basada en la arquitectura de DeepMask, lo que constituye la piedra angular de la detección moderna de objetos [56]. Los algoritmos de propuesta de objeto tienen como objetivo encontrar diversas regiones en una imagen que probablemente contengan objetos [55]. En el módulo de refinamiento de SharpMask,  $R$  es responsable de invertir el efecto de la agrupación y duplicar la resolución de la codificación de la máscara de entrada. Es decir que:  $R^i$  toma como entrada una máscara que codifica  $M^i$  generado en el pase de arriba hacia abajo, junto con las funciones de coincidencia  $F^i$  generadas en el pase de abajo hacia arriba, y aprende a fusionar la información para generar un nuevo objeto de muestreo superior que codifica  $M^{i+1}$  [50].

$$M^{i+1} = R^i(M^i, F^i)$$

Varios de estos módulos están apilados, presentando un módulo por capa de agrupación. La salida final de SharpMask es un etiquetado de píxeles de la misma resolución que la imagen de entrada [50].

La Ilustración 5 representa la arquitectura SharpMask y especificaciones del módulo de refinamiento, donde se puede apreciar la segmentación de instancias de objetos (b) [50]. Además de las redes de avance DeepMask [50], que predicen las máscaras, utilizando solo características CNN de capa superior, dando como resultado máscaras de píxeles gruesas (c,d).

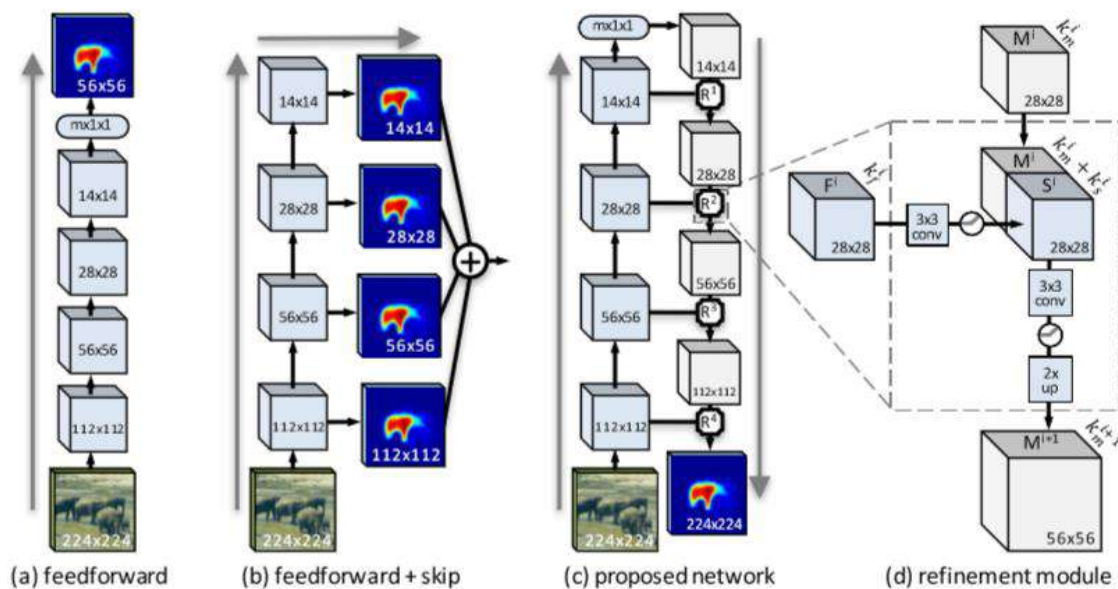


Ilustración 5. Arquitectura modelo SharpMask. Fuente: [50].

El módulo de refinamiento tiene como objetivo mejorar la codificación de la máscara  $M^i$  utilizando características de  $F^i$ .  $M^{i+1} = R^i(M^i, F^i)$ . Este proceso se aplica iterativamente  $n$  veces (donde  $n$  es el número de etapas de agrupación) hasta que el mapa de características tenga las mismas dimensiones que el parche de imagen de entrada. Cada módulo  $R^i$  tiene parámetros separados, lo que permite a la red aprender refinamientos específicos del escenario. El enfoque de refinamiento presentado en SharpMask podría aplicarse a otras tareas de etiquetado de píxeles [50].

### FastMask

La llamada FastMask, es una arquitectura de red para la generación de propuestas de objetos, que puede segmentar objetos de múltiples escalas de una sola vez. La red consta de tres módulos funcionales: cuerpo, cuello y cabeza. Al igual que DeepMask y SharpMask, el módulo del cuerpo se inicializa con una red previamente capacitada y es responsable de extraer mapas de características de la imagen de entrada. El módulo de cuello es la diferencia clave; aleja repetidamente los mapas de entidades extraídos para crear una pirámide de entidades que contiene mapas de entidades a múltiples escalas. La pirámide característica se pasa a través de una capa convolucional  $1 \times 1$  para reducir las dimensiones. A continuación, las ventanas deslizantes densas se extraen de los mapas de características de salida y se introducen en el módulo principal para la inferencia a múltiples escalas. El módulo principal es responsable de generar máscaras de segmentación a partir de esas ventanas deslizantes [46].

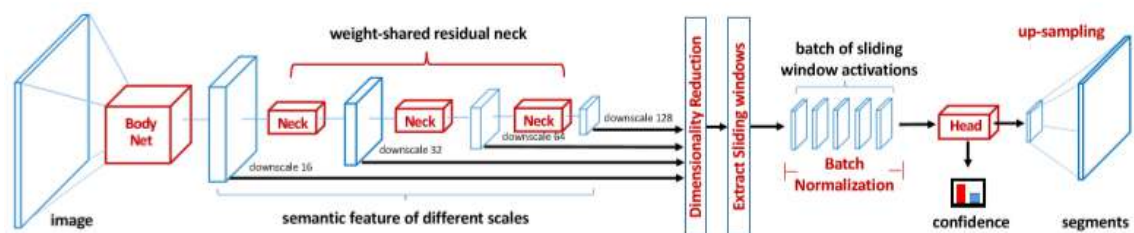


Ilustración 6. Una descripción general de la arquitectura FastMask propuesta. Fuente: [57].

En lugar de construir una pirámide de imagen de entrada, FastMask aprende a codificar la pirámide de características por un módulo de cuello y realiza una formación e inferencia de una sola vez, tal como se puede observar en la Ilustración 6. Junto con el proceso, el módulo de cabeza trabaja a escala para resaltar el objeto en primer plano

diferenciándolo del ruido de fondo, con una precisión de segmentación significativamente mejor. FastMask puede segmentar objetos en tiempo casi real ( 13 fotogramas por segundo) con imágenes a una resolución de  $800 \times 600$  [57].

## MultiPath

Para la manipulación de objetos dentro de una amplia gama de escalas, imágenes menos prototípicas y su localización más precisa, surge una arquitectura denominada MultiPath [58]. La misma propone tres modificaciones al detector estándar de objetos Fast R-CNN [59]:

- Omitir las conexiones que le dan acceso al detector a las funciones en múltiples capas de red.
- Una estructura foveal para explotar el contexto del objeto en múltiples resoluciones de objeto, similar a las características receptivas de la retina humana.
- Una función de pérdida integral y el correspondiente ajuste de red que mejora la localización de un objeto.

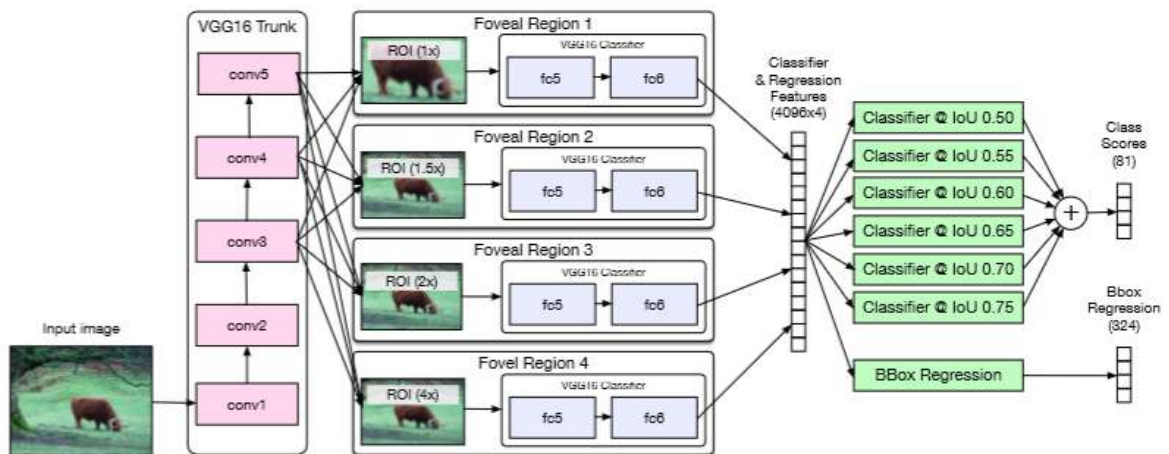


Ilustración 7. Arquitectura MultiPath. Fuente: [58].

MultiPath se basa en el marco Fast R-CNN [59]. Al igual que en Fast R-CNN, la red VGG-D [37] (pre-entrenada en ImageNet [60]) se aplica a cada imagen de entrada y la agrupación RoI se usa para extraer características para cada propuesta de objeto. Usando estas características, el clasificador final genera una puntuación para cada clase (más el fondo) y predice una localización de objetos más exacta a través de la regresión del cuadro

delimitador. En lugar de una sola cabeza de clasificador, el modelo tiene cuatro cabezas que observan regiones de contexto de diferentes tamaños alrededor del cuadro delimitador en una estructura "foveal". En segundo lugar, cada uno de estos cabezales combina características de las capas conv3, conv4 y conv5. Finalmente, los resultados de los cuatro clasificadores se concatenan y se utilizan para calcular una puntuación basada en la pérdida integral propuesta. Similar a Fast R-CNN, la red también realiza una regresión de cuadro delimitador utilizando estas mismas características. La información puede fluir a través de varias rutas paralelas [58] como se representa en la Ilustración 7.

### **Estructura Foveal**

Fast R-CNN realiza la agrupación de RoI en el cuadro delimitador de propuesta de objeto sin utilizar explícitamente la información circundante. Sin embargo, el contexto juega un papel importante en el reconocimiento de objetos [61] y para perfeccionarlo, MultiPath basa su estructura foveal en el modelo Multiregión [62]. El modelo Multiregión logra mejores resultados de localización al enfocarse en 10 cultivos separados de un objeto con un contexto variable [62]. La estructura foveal del Modelo MultiPath es en definitiva una mejora en la localización, al observar el objeto a múltiples escalas con un contexto creciente, en lugar de enfocarse en diferentes partes del objeto [58]. Para incorporar contexto, se agregan cuatro cultivos de región con campos de visión "foveal" de  $1 \times$ ,  $1.5 \times$ ,  $2 \times$  y  $4 \times$  del cuadro original, todos centrados en la propuesta de objeto. En cada caso, se utiliza la agrupación RoI para generar mapas de características de las mismas dimensiones espaciales dada cada región foveal de diferente tamaño. El procesamiento posterior comparte una estructura idéntica para cada región (pero con parámetros separados), y las características de salida de las cinco clases de clasificadores se concatenan en un solo vector. Este vector de función se utiliza tanto para la regresión de clasificación como para el cuadro delimitador [58]. El modelo foveal puede interpretarse como una versión simplificada del modelo Multiregión, dado que solamente se usan cuatro regiones en lugar de las diez que se proponen en el modelo Multiregión [62]. Con el número reducido de cabezas, se entrena la red de extremo a extremo en lugar de cada cabeza por separado [58].

### **Omisión de Conexiones**

La localización efectiva de objetos pequeños requiere características de mayor resolución de capas anteriores [63]. Por lo tanto, el modelo MultiPath presenta concatenaciones en las características normalizadas agrupadas con RoI de las capas conv3, conv4 y conv5 similares al Modelo Multiregión [62] y proporciona esto como entrada para



cada clasificador foveal. Una convolución  $1 \times 1$  se usa para reducir la dimensión de las características concatenadas a la dimensión de entrada del clasificador. Las características foveales más grandes no necesitarán características de grano fino, por lo que como optimización, se espacian ligeramente estas conexiones. Específicamente, sólo se conecta la conv3 a la cabeza del clasificador  $1 \times 1$  y conv4 a las cabezas  $1 \times 1$ ,  $1.5 \times 1.5$  y  $2 \times 2$ . En general, estas conexiones de omisión le dan al clasificador acceso a la información de las funciones en múltiples resoluciones [58].

### **Función Integral de Pérdida**

La modificación implementada por el Modelo MultiPath es la siguiente [58]:

$$L(p, k^*, t, t^*) = \frac{1}{n} \sum_u [L_{cls}(p_u, k_u^*) + \lambda [k_u^* \geq 1] L_{loc}(t, t^*)]$$

En definitiva, el Modelo MultiPath permite que la información fluya a lo largo de múltiples rutas a través de la red, lo que le permite explotar información en múltiples escalas de imagen y en el contexto de la imagen circundante [58].

### **Estrategias evolutivas para la detección de contornos y operador gradiente**

Una red neuronal artificial es un procesador distribuido en paralelo de forma masiva que tiene una tendencia natural para almacenar conocimiento de forma experimental y lo hace disponible para su uso. Se parece al cerebro humano en dos aspectos:

- El conocimiento es adquirido por la red a través de un proceso de aprendizaje. Dicho proceso se lleva a cabo a partir de los datos disponibles.
- Los pesos sinápticos o fuerza con que están interconectadas las neuronas se utilizan para almacenar la información [64].

Los algoritmos evolutivos son algoritmos de optimización meta heurísticos basados en una población de posibles soluciones a un problema determinado. Utilizan mecanismos inspirados en la biología y la teoría de supervivencia del más apto para refinar un conjunto de soluciones de forma iterativa [65]. Básicamente, establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural, generando nuevas soluciones mediante cruzamientos y mutaciones. Para desarrollar

buenas soluciones e implementar la selección natural, los algoritmos evolutivos necesitan una función de evaluación o una medida para distinguir las buenas soluciones de las malas.

Podría ser una función objetiva que sea un modelo matemático o una simulación por computadora, o puede ser una función subjetiva donde los humanos eligen mejores soluciones sobre las peores. En esencia, la medida de aptitud debe determinar un candidato de aptitud relativa a la solución, que posteriormente será utilizada por el algoritmo para guiar la evolución de las buenas soluciones [66].

Los algoritmos evolutivos representan métodos robustos de búsqueda, que permiten tratar problemas de optimización donde el objetivo es encontrar un conjunto de parámetros que minimizan o maximizan una función de adaptación (fitness). Estos algoritmos operan con una población de individuos  $P_{(t)} = \{x_1^t, \dots, x_n^t\}$ , para la iteración  $t$ , donde cada individuo  $x_i$  representa un punto de búsqueda en el espacio de las soluciones potenciales a un problema dado. El desempeño de un individuo  $x_i$  se evalúa según la función de adaptación  $f(x_i)$ . Esta función permite ordenar del mejor al peor de los individuos de la población en un continuo de grados de adaptación [67].

La población inicial evoluciona sucesivamente hacia mejores regiones del espacio de búsqueda mediante procesos probabilísticos de selección de los individuos más adaptados en la población (a mayor grado de adaptación mayor probabilidad de dejar descendencia) y de modificación por recombinación y/o mutación de la descendencia de los individuos seleccionados [67].

Los algoritmos genéticos son una subclase de algoritmos evolutivos donde los elementos del espacio de búsqueda son cadenas binarias o matrices de otros tipos elementales. En los algoritmos genéticos, las búsquedas basadas en técnicas computacionales, se fundamentan analógicamente en los mecanismos genéticos de organismos biológicos que se han adaptado y florecido ante entornos cambiantes altamente competitivos.

La última década ha sido testigo de muchos avances interesantes en el uso de algoritmos genéticos para resolver problemas de optimización en sistemas de control de procesos. Representan la solución para la optimización de problemas difíciles de forma rápida, confiable y precisa [65].

### **Algoritmos Genéticos**

La viabilidad de los algoritmos genéticos se basa en la teoría de supervivencia Darwiniana del más apto. Los algoritmos genéticos pueden contener un cromosoma, un

gen, un conjunto de población, estado físico, función de estado físico, reproducción, mutación y selección. Los algoritmos genéticos comienzan con un conjunto de soluciones representadas por cromosomas, llamado población. Se toman soluciones de una población y se utilizan para formar una nueva población, cuyo objetivo es la posibilidad de que la nueva población sea mejor que la anterior. Además, las soluciones son seleccionadas de acuerdo con su aptitud para formar nuevas soluciones, es decir, descendientes. El proceso anterior se repite hasta que se cumple alguna condición. Las principales diferencias de los algoritmos evolutivos con los métodos tradicionales de búsqueda (p.ej. métodos basados en gradientes, aleatorios, exhaustivos, etc.) son las siguientes [67]:

- Codificación del conjunto de parámetros (en general no se evolucionan los parámetros directamente sino que su codificación).
- Búsqueda en paralelo con una población de puntos.
- Uso de función de adaptación directamente (no se requiere de derivadas).
- Reglas de transición probabilísticas entre una iteración y otra.

Mediante un diagrama de flujo, el algoritmo genético básico, se describe en la Ilustración 8:

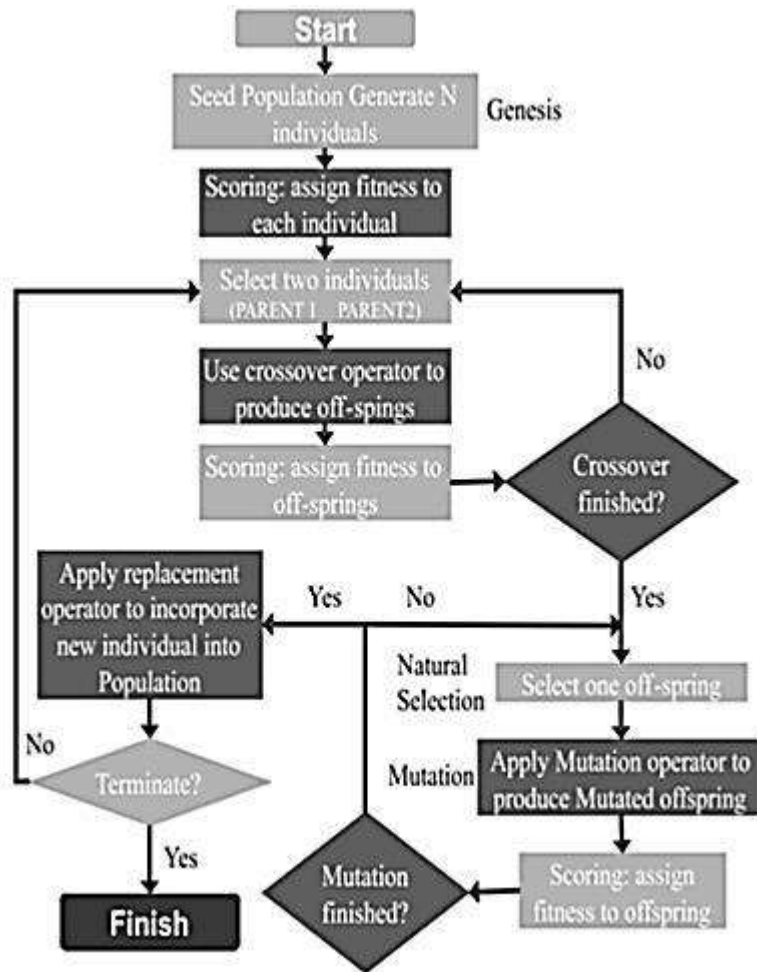


Ilustración 8. Diagrama de Flujo de un Algoritmo Genético. Fuente: [65].

Las técnicas de codificación en algoritmos genéticos son específicas del problema, lo que transforma la solución del problema en cromosomas. Existen varias técnicas de codificación utilizadas en algoritmos genéticos, entre ellas: codificación binaria, codificación de permutación, codificación de valor y codificación de árbol [65].

### Operador de cruzamiento

El operador de cruzamiento (crossover) corresponde al proceso en el cual los genes se seleccionan de los cromosomas parentales y se crea una nueva descendencia. Este es un operador sexuado que intercambia el material genético de un par de padres produciendo descendientes que normalmente difieren de sus padres. La idea central es que segmentos

distintos de padres diferentes con alta adaptación deberían combinarse en nuevos individuos que tomen ventaja de esta combinación.

El algoritmo genético explota las regiones con mayor adaptación, ya que generaciones sucesivas de selección y recombinación producen un número creciente de puntos en estas regiones [67].

El operador de cruzamiento opera con probabilidad  $p_c$  (esto permite que en algunos casos no haya recombinación y se mantengan los padres). Dados los padres  $p$  y  $q$  de un largo de  $n$  bit, se escoge aleatoriamente un punto  $k \in \{1, \dots, n - 1\}$  y se intercambian los bits a la derecha de esa posición entre ambos individuos, obteniéndose los descendientes  $s$  y  $v$ , como se indica a continuación:

$$\vec{p} = \{p_1, \dots, p_{k-1}, p_k, \dots, p_n\}$$

$$\vec{q} = \{q_1, \dots, q_{k-1}, q_k, \dots, q_n\}$$

$$\vec{s} = \{p_1, \dots, p_{k-1}, q_k, \dots, q_n\}$$

$$\vec{v} = \{q_1, \dots, q_{k-1}, p_k, \dots, p_n\}$$

El operador crossover de un punto, sufre de un sesgo posicional ya que un bit cercano al extremo derecho de la tira tiene una alta probabilidad de intercambio, mientras que un bit en el extremo izquierdo tiene una baja probabilidad de intercambio. El operador crossover binomial corrige este sesgo, intercambiando bits entre padres sobre una base bit a bit, con probabilidad aleatoria  $p \in [0,1]$ , distinta para cada posición [67].

### **Operador de mutación**

En el algoritmo genético simple, el operador de mutación juega un papel secundario, invirtiendo ocasionalmente un bit. Tasas de mutación pequeñas garantizan que un individuo no difiera mucho de sus padres en el genotipo (tira binaria). La mutación sirve para evitar la pérdida de diversidad producto de bits que han convergido a un cierto valor para toda la población y que, por tanto, no pueden ser recuperados por el operador de recombinación. El operador de mutación invierte cada bit de la tira binaria sobre una base bit a bit con probabilidad  $p_m$ ,  $\hat{m}_{\{p_m\}}(q_1, q_2, \dots, q_n) = (\hat{q}_1, \hat{q}_2, \dots, \hat{q}_n), \forall i \in [1, \dots, n]$ , en donde  $\hat{q}_i = q_i si(r > p_m)$ ,  $\hat{q}_i = 1 - q_i si(r \leq p_m)$  y  $r \in [0,1]$  uniformemente aleatorio, distinto para cada bit  $q_i$  [67].

## Operador Gradiente

El operador gradiente es una herramienta ampliamente utilizada para detectar las variaciones en los niveles de intensidad que pudieran corresponder a los contornos de interés en la clasificación de imágenes [5]. En los algoritmos genéticos esta herramienta no es necesaria, ya que para estructurar las posibles soluciones a un problema, aquellos utilizan una función de evaluación o adaptación [67]. Sin embargo, en problemas puntuales para la extracción de contornos en imágenes ruidosas o con objetos con niveles de intensidad no uniforme, por ejemplo, se realiza típicamente una detección de bordes utilizando el operador gradiente, seguida por algún esquema de procesamiento global como puede ser un algoritmo genético [5].

El operador gradiente determina los cambios bruscos de intensidad  $I = (x, y)$  de los objetos que conforman la imagen. Matemáticamente se expresa de la siguiente manera:

$$\nabla I(x, y) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x, y)}{\partial x} \\ \frac{\partial I(x, y)}{\partial y} \end{bmatrix}$$

En muchas ocasiones resulta conveniente discriminar el gradiente escalar  $\Delta = \sqrt{G_x^2 + G_y^2}$  que abstrae la dirección del gradiente propiamente dicho [5]. Los operadores  $G_x$  y  $G_y$  representan una implementación genérica de las derivadas direccionales en la imagen digital, pudiéndose implementar por medio de las máscaras de Roberts, Prewitt, o Sobel.

Entonces la magnitud del vector gradiente es:

$$M(x, y) \approx |G_x| + |G_y|$$

Y la dirección del vector gradiente:

$$\alpha(x, y) = \tan^{-1} \left( \frac{G_y}{G_x} \right)$$

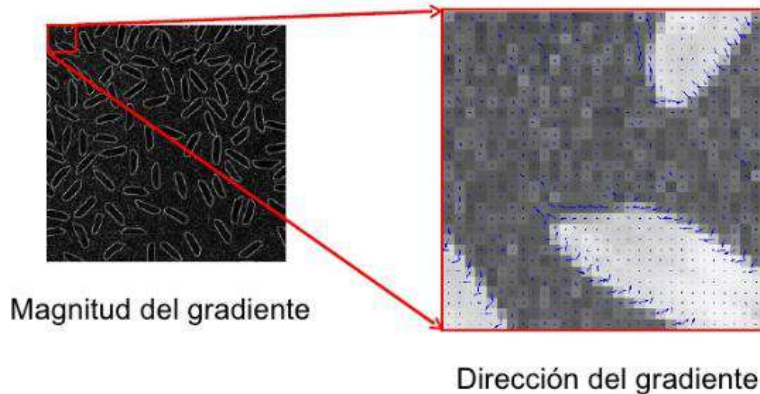


Ilustración 9. Representación de la aplicación del operador gradiente.

### **Combinación de metodologías para la extracción de contornos**

A partir de una población inicial de soluciones posibles del contorno, el sistema evoluciona mediante operaciones de mutación y cruzamiento, que inducen soluciones del contorno paulatinamente mejores. La selección se realiza evaluando la función de adaptación (fitness) de las soluciones mediante el operador  $\nabla$ , y el resultado esperado se detalla a continuación [5]:

**Población Inicial:** Sin dudas el contorno óptimo buscado estaría ubicado en el límite externo del objeto de interés. Luego es posible generar una población inicial de soluciones adecuada para disparar un algoritmo evolutivo que incluya este límite. Un ejemplo de esto se efectúa posicionando manualmente dos círculos concéntricos, o bien, determinando automáticamente estos radios. Posteriormente se genera aleatoriamente en la zona comprendida entre ambos radios una cantidad  $N$  de contornos, conformada por igual cantidad  $M$  de puntos, espaciados en ángulos uniformes a lo largo del anillo y en posiciones aleatorias de forma radial. Es posible en este punto configurar tanto la cantidad de contornos a generar como la cantidad y el espaciamiento de los puntos que determinan cada uno de ellos [5].

**Mutación y Cruzamiento:** La operación de mutación implementada realiza, para cada uno de los contornos generados, la selección aleatoria de uno de sus puntos. Luego modifica su posición con mayor incidencia sobre componente radial para una misma ubicación angular. De esa manera se obtiene un contorno modificado a partir del original. La operación de cruzamiento (crossover) se efectúa tomando los contornos de a pares.

Luego se elige una posición aleatoria y se intercambian los puntos de ambas curvas desde dicho punto hasta el final del contorno, obteniendo dos nuevas curvas [5].

**Fitness y selección:** La evaluación del fitness de un contorno debe cuantificar la proximidad del mismo al supuesto valor óptimo. El objetivo es detectar la poligonal de “costo mínimo”, es decir, que minimice el costo local acumulado a lo largo de la misma. El costo local en cada uno de los puntos disminuye cuando se encuentra en una zona de la imagen donde el gradiente es alto. La función de costo mínimo en un pixel  $(x, y)$  entonces es [5]:

$$k(x, y) = 9 - \sum_{i=-1}^1 \sum_{j=-1}^1 \nabla(x + 1, y + 1)$$

De esta manera el costo local en un punto  $c_i^j$  del contorno  $C_i$  puede expresarse como  $k = (c_i^j)$  y para evaluar la función de adaptación o fitness  $f(C_i)$  del contorno  $i$ -ésimo simplemente es necesario acumular el costo local de cada uno de los  $M$  puntos que conforman  $f(C_i) = \sum_{j=1}^M k(c_i^j)$ .

La estrategia propuesta facilita la utilización y evaluación de otras alternativas en los operadores de mutación y cruzamiento, como también la posibilidad de adaptar diferentes esquemas para efectuar el cómputo del fitness, manteniendo sin cambios el resto del algoritmo. La naturaleza “adaptativa” del sistema evolutivo permite, además, la potencial extensión del sistema ampliando su campo de aplicación para soportar condiciones dinámicas de las imágenes de entrada [5].

### **Análisis de imágenes biomédicas**

La interpretación de imágenes médicas facilita el proceso de diagnóstico de casi todas las enfermedades, desde el cáncer de pulmón hasta una enfermedad cardíaca. El análisis digital de imágenes de patología y microscopía se usa frecuentemente para estudios completos de la morfología celular o de la estructura de los tejidos, por cuanto la evaluación manual suele requerir mucho trabajo y es propensa a variaciones entre observadores. Los métodos asistidos por computadora, que pueden mejorar significativamente la objetividad y la reproducibilidad, han despertado un gran interés en las publicaciones recientes [68].

Un algoritmo automático de clasificación, segmentación y detección de imágenes patológicas o de microscopía puede jugar un papel importante en la toma de decisiones para el diagnóstico de enfermedades y, por consiguiente, ayudar a garantizar la mejor opción de tratamiento posible. En la actualidad hay diferentes modelos DCNN que se han aplicado



con éxito en la patología computacional y se están volviendo muy populares en el campo, debido al rendimiento de vanguardia en muchas aplicaciones diferentes [69].

Uno de los problemas puntuales de interpretación de imágenes de microscopía se presenta en la detección y segmentación de núcleos celulares. Ello ocurre básicamente por dos razones, primero, la morfología de los núcleos es un componente clave en la mayoría de las clasificaciones de grado de cáncer por ejemplo. Segundo, las técnicas eficientes de segmentación de núcleos pueden reducir significativamente el esfuerzo humano para el análisis a nivel celular [69].

Una revisión de las publicaciones recientes referidas al tema, considera que la detección del núcleo de la célula resuelve la ubicación del objeto sin delinear con precisión los límites, y generalmente se conoce como búsqueda de marcador o semilla cerca de los centroides del objeto, uno por núcleo. El marcador será un solo punto o una pequeña región dentro del objeto. Claramente, la detección de marcadores puede proporcionar soporte para el conteo, seguimiento y segmentación de núcleos [68].

Aunque diferentes tipos de imágenes de microscopía o imágenes de tinción exhiben variaciones significativas en las características de la imagen, de modo que cada una puede requerir un algoritmo específico para la detección del núcleo celular, los principales métodos presentados en la literatura reciente pueden clasificarse ampliamente en varios grupos según sus algoritmos subyacentes: distancia transformación, operación de morfología, transformación H-mínima-máxima, filtrado laplaciano de Gauss (LoG), detección de región extrema máximamente estable (MSER), transformación de Hough, votación basada en simetría radial y aprendizaje supervisado [68]. La siguiente ilustración resume a modo gráfico, algunas publicaciones que contienen detección de núcleos celulares. Las columnas de resultados DET y resultados SEG, informan la precisión de detección y segmentación, respectivamente, si existe una cuantificación específica informada en las publicaciones; de lo contrario, solo se proporcionan las métricas [68].

data	DET	DET results	SEG	SEG results
over 500 multifocus z-stack muscle images	SVM, mean shift	precision/recall/F <sub>1</sub> : 93%/96%/94%	-	-
10 bladder inverted papilloma images	SVM	USR: 0.2%, OSR: 0.4%	watershed	MAD: 0.85
6 CHO, 5 L929, and 5 Sf21 cell line images	random forest, hierarchical clustering	precision: 77.7%±8% (CHO), 82.8%±4.4% (L929), 97.3%±0.9% (Sf21), recall: 92.9%±3% (CHO), 92.6%±2.9% (L929), 96.4%±3.2% (Sf21)	-	-
21 cervical images	multiscale CNN, graph cut	-	watershed	Dice: 95%±18% (cytoplasm), F <sub>1</sub> : 99%±1% (nuclei)
48 U2OS and 49 NIH3T3 cell images	PCA, template matching	-	nonrigid registration	Rand index: 95% (U2OS), 91% (NIH3T3), Jaccard index: 2.5 (U2OS), 2.72 (NIH3T3)
30 rolling leukocyte images	Bayesian classifier	detection rate/false alarm rate: 78.6%/13.1%	-	-
50 breast histology images	sparse representation classification	precision/recall/F <sub>1</sub> : 84.98%/87.18%/86.07%	-	-
30 brain tumor (BT), 51 pancreatic neuroendocrine tumor (NET), and 35 breast cancer (BC) images	CNN	precision/recall/F <sub>1</sub> : 72%±20%/88%±15%/77%±16% (BT), 84%±8%/93%±5%/88%±4% (NET), 71%±14%/88%±9%/78%±8% (BC)	active contour	Dice/HD/MAD: 85%±11%/5.06±5.26/3.26±2.89 (BT), 92%±9%/2.41±2.99/1.58±1.75 (NET), 80%±15%/8.6±6.77/6.24±4.90 (BC)

Ilustración 10. Resumen de publicaciones de revistas basadas en los algoritmos subyacentes de los métodos de detección y segmentación. Fuente: [68].

Como se puede observar en la ilustración anterior, los dataset analizados resultan ser variados y compuestos por distintas líneas celulares o tejidos humanos. El biomonitorio humano se ha convertido en una herramienta central tanto en la medicina ambiental como la ocupacional y la investigación en la identificación, control y prevención de exposiciones de la población a compuestos potencialmente dañinos [70].

El análisis de micronúcleos (MN) es un estándar de enfoque para la evaluación del daño cromosómico en poblaciones humanas expuestas a diferentes factores ambientales, ocupacionales o de estilo de vida y para pruebas de genotoxicidad *in vitro* [12].

La puntuación o conteo de MN generalmente se realiza en los linfocitos de sangre periférica, pero también se puede calificar relativamente fácil en otros tipos de células relevantes para el biomonitoreo humano, como los fibroblastos, las células epiteliales exfoliadas (de las células bucales, de la mucosa nasal o de la vejiga en la orina) y en eritrocitos [71].

Algunas publicaciones recientes acerca de la automatización del análisis de imágenes para la puntuación *in vitro*, describen los procedimientos como de escaneo automático de portaobjetos, detección de células, núcleos y MN, aunque se requiere una validación visual de estos últimos. Si bien, las frecuencias absolutas de MN encontradas con puntuación automatizada fueron menores en comparación con las obtenidas por puntuación visual, el sistema de puntuación automatizado es capaz de producir resultados biológicamente relevantes y confiables, con baja variabilidad entre observadores en el paso de validación visual, y por lo tanto que puede usarse para biomonitoreo [72].

La técnica de Micronúcleos que se presenta en el estudio de caso abordado se realizó de acuerdo con el protocolo desarrollado por Hooftman y De Raat [80], con algunas modificaciones. Las células empleadas fueron extraídas de moluscos y peces, observándose marcadas y notables diferencias con las células y tejidos humanos.

La revisión de la literatura detallada sugiere la posibilidad de investigar la viabilidad de las soluciones automatizadas empleadas en células humanas, adaptándolas a las células animales.

## **Capítulo III**

### *Presentación del Estudio de Caso*

## **Procesamiento de Imágenes en el estudio de caso abordado**

### **Procesamiento de Imágenes Digitales**

La unidad primitiva de representación más común de la información, que puede ser una imagen, es el píxel; es decir que cada píxel representa un valor discreto correspondiente a una señal analógica registrada o medida por un periférico. Los píxeles de una imagen se almacenan en la memoria de manera consecutiva empleando un número de bits para representar el valor de cada píxel. Podríamos decir que un píxel típicamente representa intensidad luminosa (niveles de gris o color). La calidad de una imagen digital, depende de la resolución (número total de píxeles de la imagen), del número de bits por píxel, y lógicamente del sensor del periférico encargado de medir la señal luminosa [73]. La representación discreta (o imagen digital) siempre es una aproximación, dado que se tiene un número finito de muestras.

El proceso de digitalización de imágenes puede realizarse por medio de la utilización de un escáner o una cámara digital. Las imágenes son, comúnmente, digitalizadas en el formato RGB (Red, Green, Blue), con 8 bits por píxel y cuentan con 256 posibles valores para cada canal, lo que significa más de 16 millones de posibles combinaciones de los colores primarios. En ocasiones, las imágenes RGB con 8 bits por canal se denominan imágenes de 24 bits ( $8 \text{ bits} * 3 \text{ canales} = 24 \text{ bits de datos por píxel}$ ); y este es un modelo sensorial más orientado a los equipos electrónicos. Los colores se representan en coordenadas cartesianas, en donde cada color es un vector de origen y la diagonal principal corresponde a la escala de grises [74].

### **Espacio RGB**

El espacio RGB está basado en el sistema de coordenadas Cartesiano en donde los colores son puntos definidos por vectores que se extienden desde el origen, en donde el negro está en el origen y el blanco está ubicado en la esquina opuesta al origen [74]. En la Ilustración 11 se puede clarificar el concepto.



Ilustración 11. Espacio de color RGB. Fuente:[74].

Dado un pixel  $p$  la composición de su color es la combinación lineal de los vectores base verde, rojo y azul [74]:

$$\phi_p = r_p \hat{i} + g_p \hat{j} + b_p \hat{k}$$

En donde  $r_p$ ,  $g_p$  y  $b_p$  son los componentes de rojo, verde y azul, respectivamente. La orientación y magnitud de un vector de color define la cromaticidad e intensidad del color, respectivamente. A tal fin, se entiende por cromaticidad al grado de diferencia existente entre un color y un gris de su misma luminosidad y claridad, que se corresponde con la saturación del color percibido. Cabe aclarar que este espacio es sensible a la iluminación; es decir, a pesar de que dos vectores tengan la misma cromaticidad, estos representan diferentes colores si sus intensidades son diferentes[75].

### **Espacio HSV**

La representación de color en el espacio HSV, representada en la Ilustración 12, emula la percepción humana del color ya que la cromaticidad es desacoplada de la intensidad [74]. En este espacio el color de un pixel  $p$  se representa por sus componentes de tono ( $h$ ), saturación ( $s$ ) e intensidad ( $v$ ) [75]:

$$\varphi_p = [h_p, s_p, v_p]$$

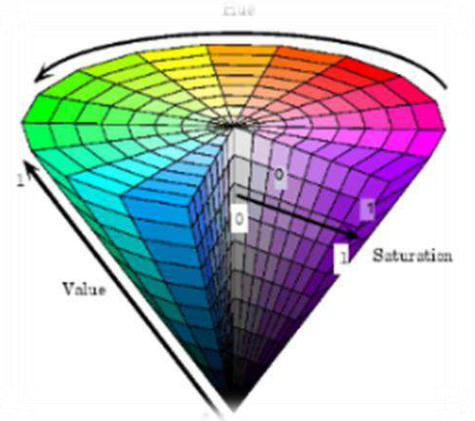


Ilustración 12. Espacio HSV. Fuente:[75].

El negro y el blanco no tienen una cromaticidad definida. El blanco se obtiene cuando la saturación de un color es bajo, es decir, cuando  $s \approx 0$ ; por otra parte, el negro se obtiene cuando la intensidad del color es baja, esto es, cuando  $v \approx 0$ .

En el escenario actual del estudio de caso presentado, la percepción humana del color determina no sólo la cantidad de células por individuo que serán investigadas a través de la técnica de Micronúcleos, sino también la clasificación de las muestras según el daño que presenten.

### **Especificaciones del Estudio de Caso**

El Instituto de Biología Subtropical (IBS) es una unidad ejecutora de investigaciones científicas y tecnológicas de doble dependencia. Depende de la Universidad Nacional de Misiones (UNaM) y del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET). En la sede de la Ciudad de Posadas, que funciona en las instalaciones de la Facultad de Ciencias Exactas, Químicas y Naturales, y específicamente en el área de investigación en Mutagénesis y Monitoreo Ambiental de aguas de ríos y arroyos de la Provincia de Misiones, se aplican a las muestras entre otras técnicas, la técnica de Micronúcleos [14].

Las aguas superficiales, como lagos, ríos y mares contienen mezclas complejas de contaminantes como resultado de la acción antropogénica, incluidos los compuestos genotóxicos que causan efectos adversos sobre la salud pública y los ecosistemas acuáticos [76]. El biomonitoreo, empleando organismos acuáticos, es esencial para indicar calidad y

sostenibilidad del medio ambiente [10]. En este sentido, los peces y moluscos bivalvos son adecuados como bioindicadores de condiciones ambientales debido a su posición en la cadena trófica y su sensibilidad a bajas concentraciones de sustancias genotóxicas [77].

La prueba de Micronúcleos es una de las técnicas citogenéticas más empleadas para la evaluación de genotoxicidad y mutagenicidad. Los Micronúcleos surgen a partir de cromosomas completos o fragmentos acéntricos que se retrasan durante la anafase, ya sea por la falta de centrómeros o por daño al huso mitótico [1]. Además, los cambios en la morfología normal de los núcleos, también se consideran indicadores de daño genotóxico y son descritos como anormalidades nucleares que constituyen un análisis complementario al recuento de Micronúcleos [7].

## **Materiales y Métodos**

### **Muestras**

Se realizaron muestreos estacionales sobre el río Paraná y arroyos tributarios durante los años 2009-2017, donde se colectaron ejemplares de peces nativos y moluscos bivalvos pertenecientes a diferentes géneros y especies. Los sitios muestreados abarcaron desde el norte de la provincia de Misiones en la ciudad de Puerto Iguazú, hasta su extremo sur en la ciudad de Posadas, a los fines de analizar el impacto generado por diferentes fuentes de contaminación antrópica sobre los organismos y su material genético. Se detalla a continuación [78]:

- Puerto Iguazú: ciudad ubicada en el extremo norte de la provincia y próxima a los límites internacionales con las Repúblicas de Paraguay y Brasil. En el año 2011 se habilitó en las afueras de la ciudad, la red y planta de tratamiento de líquidos cloacales desde donde los residuos se dirigen en última instancia al Río Paraná, distante 4,6 kilómetros aguas abajo del sitio de muestreo seleccionado. Anteriormente, los líquidos cloacales eran evacuados de forma cruda al parque denominado “Honorio Rolón”, con salida directa al Río Iguazú, distante 1,5 kilómetros de la desembocadura en el Río Paraná. El tramo comprendido desde el sitio de descarga y el sitio elegido para la toma de muestras sobre el Río Paraná, es de 3,5 kilómetros aguas abajo. (Coordenadas: 26°28’46,6”S, 54°35’33,1”O) [78].
- Puerto Piray: sitio receptor de efluentes de una industria papelería dedicada a la fabricación de papel y pulpa al sulfito, cuya tecnología de blanqueo utiliza cloro elemental. Al presente, dicha industria, se encuentra prácticamente en estado obsoleto y con grandes períodos de inactividad, principalmente



asociados a cuestiones legales relacionadas a infracciones a la Ley 24.051 de residuos tóxicos. (Coordenadas: 26°28'19,6"S, 54°43'35,6"O) [78].

- Puerto Mineral: sitio perteneciente al Municipio de Capióví y ubicado a 2,3 kilómetros aguas debajo de una industria papelera dedicada a la fabricación del papel y pulpa kraft, que utiliza para el blanqueo tecnología totalmente libre de cloro (TCF). Los efluentes industriales son evacuados a un pequeño arroyo tributario, a pocos centenares de metros de su desembocadura en el Río Paraná. (Coordenadas: 26°57'16,1"S, 55°08'17,1"O) [78].
- Posadas: sitio correspondiente a la capital de la provincia y ubicado en el extremo sur. Recibe importantes desechos fluviales y urbanos. El lugar escogido para la toma de muestras, denominado Club Luz y Fuerza, se encuentra 1 kilómetro aguas arriba del emisario de descarga de la planta de tratamiento de residuos cloacales de la ciudad [78]. Adicionalmente, se realizaron muestreos estacionales en los siguientes arroyos urbanos [79]: Antonica, Itá, Mártires, Pindapoy y Zaimán. En todos los casos, los cursos de agua reciben diferentes tipos de desechos urbanos y están rodeados en general por asentamientos poblacionales de distinta magnitud. El arroyo Antonica, ubicado dentro de la ciudad (27°21'S, 55°56'O), se caracteriza por la presencia de asentamientos urbanos precarios en donde el agua es utilizada para diferentes actividades tales como: riego de cultivos, consumo y pesca de subsistencia. El Itá es también un arroyo urbano (27°21'S, 55°54'O) y en cercanías a su desembocadura en el río Paraná, se realizan actividades antrópicas muy variadas como navegación, remo, pesca tanto de subsistencia como deportiva, entre otras. Un detalle que debe ser destacado es el hecho de que en él se ubica una reserva urbana, considerada un espacio para la conservación de la flora y la fauna. El arroyo Mártires, considerado en la actualidad como subembalse debido a las modificaciones sufridas luego del llenado del embalse principal de la represa Yacyretá (INA/EBY, 2005), se encuentra en la periferia de la ciudad de Posadas (27°22'S, 55°57'O). En este sitio se realizan actividades recreativas de tipo balnearias y es, no obstante, el principal receptor de los desechos residuales de la planta de tratamientos de efluentes cloacales de la ciudad. El arroyo Pindapoy se posiciona en el municipio de Garupá y está ubicado dentro del departamento capital (27°26'S, 55°44'O). En este sitio se observa la presencia de balnearios en los que se realizan diversos tipos de actividades recreativas con contacto directo. Aguas arriba del punto de muestreo se ubican criaderos comerciales de peces y corrales de engorde o "feed lot" de ganado bovino que aportan una gran cantidad de nutrientes y coliformes a este arroyo. Además, sobre el curso alto del arroyo Pindapoy se encuentra la planta de residuos sólidos urbanos de la ciudad de Posadas y alrededores. Finalmente, el arroyo Zaimán (27°25'S, 55°54'O) se encuentra sujeto a la construcción

de obras viales como parte del plan de finalización de obras complementarias de la Entidad Binacional Yacyretá (EBY). La población que se asienta sobre las márgenes de este arroyo, utiliza sus aguas para actividades asociadas al riego y la pesca [79].

- Candelaria: El Anselmo, es una zona costera deshabitada cerca del municipio de Candelaria sin influencia antrópica directa. Un total de veintitrés muestreos se realizaron durante el período 2007-2010 que comprende los meses de Febrero, Marzo, Abril, Junio, Agosto, Octubre y Noviembre [7].

Los peces se recolectaron utilizando redes de espera (tamaños de malla: 4, 5 y 6 cm) y redes de arrastre; mientras que los bivalvos fueron obtenidos manualmente. Cada muestra fue registrada en una base de datos con un código numérico que señala la especie, fecha y lugar de recolección, frecuencias de Micronúcleos y anomalías nucleares. Las capturas de ejemplares y extracción de muestras biológicas fueron autorizadas por el Ministerio de Ecología y Recursos Naturales Renovables de la provincia de Misiones [7].

### **Técnica de Micronúcleos**

La técnica de Micronúcleos (MN) se realizó de acuerdo con el protocolo desarrollado por Hooftman y De Raat [80], con algunas modificaciones. Se obtuvieron muestras de sangre periférica de los peces mediante punción de la vena caudal con jeringas previamente heparinizadas y de hemolinfa en los moluscos bivalvos a partir del músculo posterior aductor. Para cada ejemplar, se realizaron frotis sanguíneos por duplicado sobre portaobjetos y los extendidos se dejaron secar a temperatura ambiente durante toda la noche. Los preparados se fijaron en metanol absoluto durante 15 minutos y fueron teñidos con una solución de Giemsa al 10% durante 10 minutos. Se analizaron un total de 2000 eritrocitos nucleados por espécimen en el caso de los peces y de 1000 hemocitos (granulocitos) en los bivalvos, contabilizando 1000 y 500 células por cada portaobjeto duplicado, respectivamente. Los análisis se llevaron a cabo en microscopio óptico de luz blanca bajo inmersión, con un aumento de 100X [7]. La contabilización de las células se realizó mediante observación directa, por medio de la percepción del ojo entrenado del investigador.

Los siguientes criterios de inclusión se utilizaron para el recuento de Micronúcleos:

- 1) debían estar separados y con la misma tinción del núcleo principal,
- 2) presentar tamaño más pequeño que 1/3 del núcleo principal,
- 3) no mostrar refringencia.

Las anomalías nucleares (NA) se distinguieron en muescas, lobuladas y núcleos globulares, según la clasificación propuesta por Carrasco [81]. Además, dos tipos de anomalías nucleares fueron encontradas en los individuos estudiados, que han sido denominadas “núcleos en forma de ocho” y “núcleos con gemaciones” [7] que se ven plasmadas en las Ilustraciones 13 y 14.

La clasificación, según los criterios antes detallados, también fue llevada a cabo por la percepción humana. La presente investigación propone una herramienta bioinformática capaz de realizar tanto la contabilización de las células para cada muestra como la clasificación de las mismas según los criterios de inclusión que se utilizan en las investigaciones llevadas adelante por el Laboratorio.

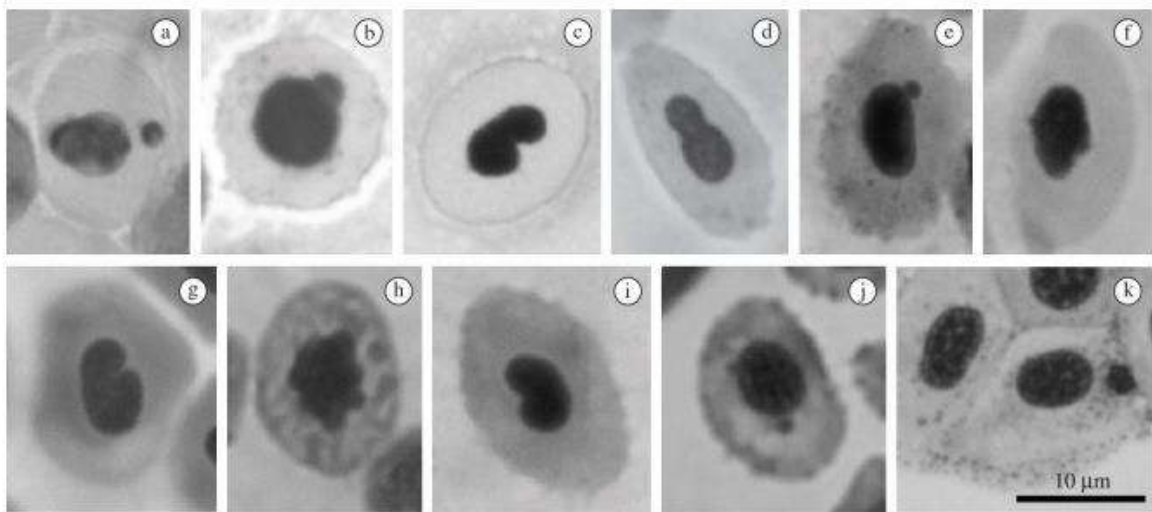


Ilustración 13. Fotomicrografías de MN y NA en eritrocitos de las especies de peces analizadas. Fuente: [7].

En la Ilustración 13 se detallan las siguientes referencias:

- a) Eritrocito con MN en la especie *Steindachnerina brevipinna*.
- b) Núcleo con gemación en la misma especie de a).
- c) Núcleo en forma arriñonada en la especie *Astyanax asuncionensis*.
- d) Núcleo en forma de ocho en eritrocito de la misma especie que en c).
- e) Eritrocito con micronúcleo en la especie *Astyanax schubarti*.
- f) Núcleo globular o con muesca en la especie *Megaleporinus obtusidens*.
- g) Núcleo en forma arriñonada en la especie *Schizodon nasutus*.
- h) Núcleo lobulado en la especie *Apareiodon affinis*.
- i) Núcleo en forma arriñonada en la especie *Crenicichla niederleinii*.
- j) Eritrocito con micronúcleo en la especie *Pimelodella cf. griffini*.

k) Eritrocito con micronúcleo en la especie *Loricaria simillima*.

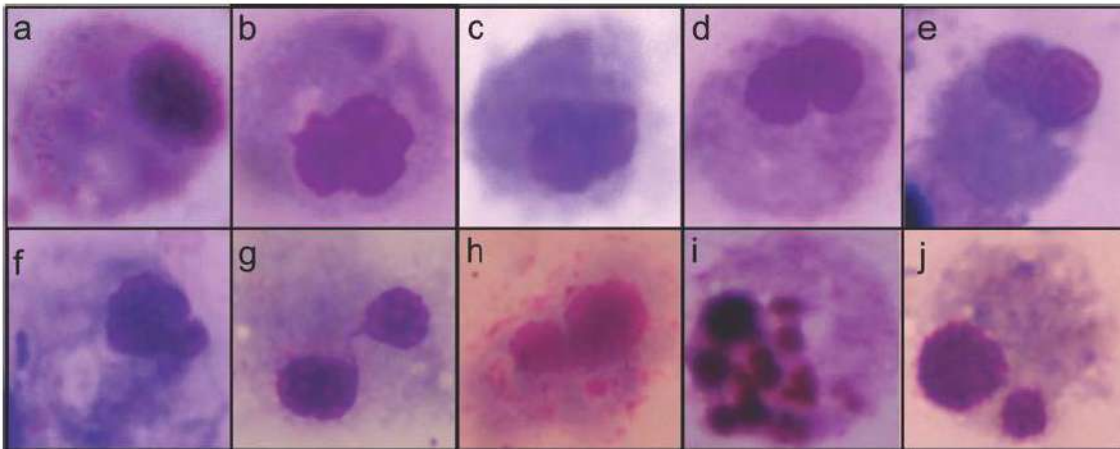


Ilustración 14. Fotomicrografías de MN y NA en hemocitos de las especies de moluscos bivalvos analizadas. Fuente:[7].

En la Ilustración 14 se pueden describir: células en extendidos de hemolinfa de moluscos bivalvos:

- a) hemocito normal;
- b) hemocito con núcleo lobulado;
- c) hemocito con núcleo globular;
- d) y h) hemocito con núcleo en forma arriñonada;
- e) hemocito con núcleo en forma de ocho;
- f) hemocito con gemación nuclear;
- g) hemocito con puente cromatínico nuclear y célula binucleada;
- h) hemocito con núcleo fragmentado (apoptótico); y,
- i) hemocito con micronúcleo.

### **Clasificación de Imágenes**

Para el entrenamiento de la herramienta bioinformática, las imágenes de entrada se clasifican de acuerdo a la muestra, si pertenecen a moluscos o peces. La grilla de guía que corresponde a un molusco, es la Ilustración 15, donde están representadas las células en extendidos de hemolinfa de *Limnoperna fortunei* [78].

- A. Granulocito normal.
- B. Granulocito con núcleo lobulado.

- C. Granulocito con núcleo globular.
- D. Granulocito con núcleo en forma arriñonada.
- E. Granulocito con núcleo en forma de ocho.
- F. Granulocito con gemación nuclear.
- G. Granulocito binucleado con puente de cromatina.
- H. Granulocito con Micronúcleo.
- I. Granulocito con Micronúcleo.
- J. Granulocito rodeado de bacterias.
- K. Hialinocito Normal.
- L. Hialinocito Binucleado.
- M. Hialinocito Trinucleado.
- N. Hialinocito Tetranucleado.
- O. Hialinocito con Micronúcleo.

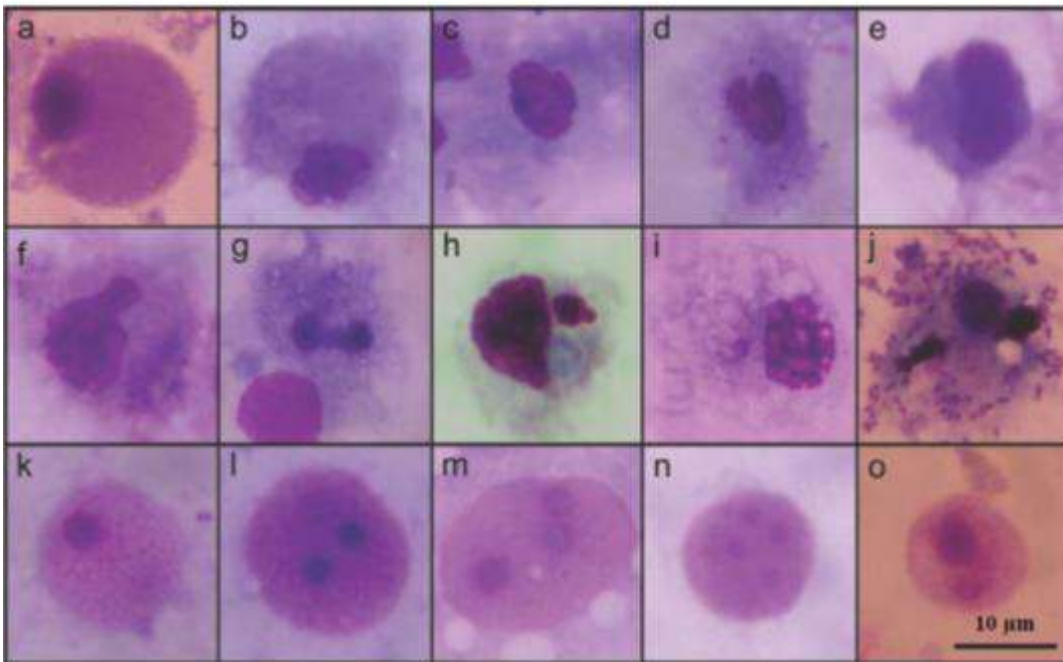


Ilustración 15. *Limnoperna fortunei*, especie de molusco bivalvo .Fuente [78].

Para la clasificación de las imágenes que corresponden a las células de peces, se utiliza la Ilustración 16 como datos de entrada. La misma hace referencia a las alteraciones nucleares obtenidas durante el Test de MN y AN en eritrocitos de *Steindachnerina brevipinna* [78].

- A. Eritrocito Normal.

- B. Núcleo lobulado.
- C. Núcleo Globular.
- D. Núcleo en forma arriñonada.
- E. Núcleo en forma arriñonada.
- F. Núcleo en forma de ocho.
- G. Núcleo en forma de ocho.
- H. Gemaciones Nucleares.
- I. Gemaciones Nucleares.
- J. Puente Cromatínico.
- K. Eritrocitos Binucleados.
- L. Eritrocitos Binucleados.
- M. Eritrocitos con Micronúcleos.
- N. Eritrocitos con Micronúcleos.
- O. Eritrocitos con Micronúcleos.
- P. Eritrocitos con Micronúcleos.

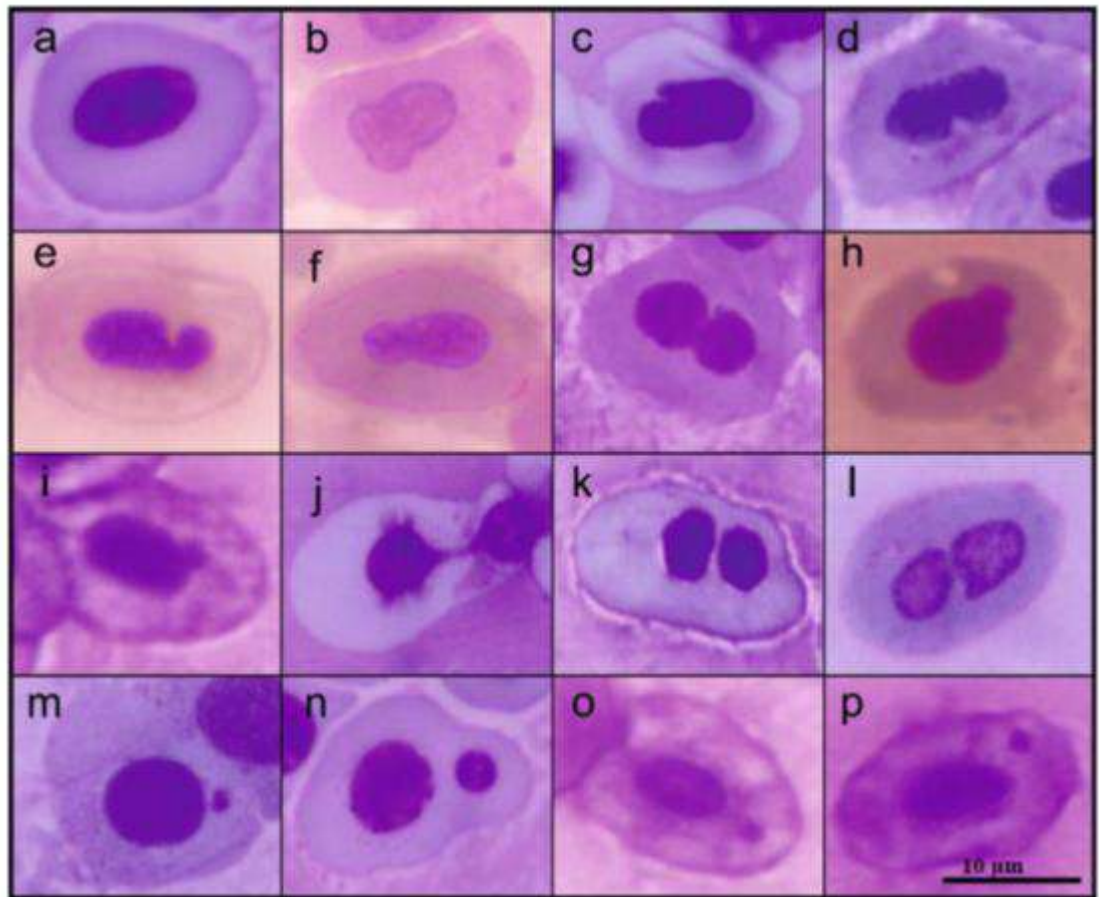


Ilustración 16. *S. brevipinna*, especie de pez. Fuente: [78].

## **Hardware para digitalización de imágenes**

Todas las imágenes de los Micronúcleos en el área de investigación en Mutagénesis y Monitoreo Ambiental de aguas de ríos y arroyos de la Provincia de Misiones, fueron tomadas a través de cámaras fotográficas compatibles con los microscopios ópticos (MO). Específicamente, se utilizan los siguientes recursos:

- MO Olympus con un tubo adaptado en la parte superior para encastrar la cámara de fotos (modelo Olympus C-5000 ZOOM). Tiene una resolución de 5 megapíxel y 3X de zoom óptico.
- MO marca MOTIC presenta una Tablet incorporada. Se anexa una cámara digital de resolución de 5 megapíxel que se incorpora al MO.

Para la estimación de las frecuencias de Micronúcleos (MN) y de anomalías nucleares (NA) tanto en peces nativos del río Paraná (Argentina) como de moluscos bivalvos antes citada [7], se utilizaron tanto el MO marca Olympus como el MO marca MOTIC y la cámara adaptada. Todas las células se fotografiaron en el mayor aumento del microscopio (100X, con aceite de inmersión) y el mayor aumento de zoom de la cámara. Estas imágenes digitales están alojadas en una base de datos local y son las que se utilizan como datos de entrada a los prototipos descritos en el presente trabajo.

ASUS I7-X450L Series, es el computador utilizado para el desarrollo de los prototipos y cuenta con las siguientes características técnicas:

- Procesador: Intel Core i7 4500U (1800 MHz - 3000 MHz)
- RAM: 8GB DDR3 (1600 MHz)
- Pantalla: LED 14.0" (1366x768)
- Batería: 4 celdas (37000 mWh)
- Almacenamiento: HDD 1TB (5400rpm)
- Tarjetas de video: Intel HD Graphics 4400 (Integrada). NVIDIA GeForce 840M (2GB).

## **Tareas desarrolladas**

Fase inicial:

- Estado del arte en cuanto a reconocimiento de patrones en imágenes digitales de microscopía.
- Elección de dos líneas de investigación: la primera referida a la posibilidad de utilizar la metodología del algoritmo DeepMask [49] junto con una CNN [34] para la detección y contabilización de MN y NA [7]. La segunda línea de investigación, corresponde a la utilización de un algoritmo evolutivo con la incorporación de un operador gradiente en la etapa de adaptación [5] para cumplimentar el mismo objetivo.

Segunda fase:

- Elaboración y evaluación de la eficacia en el cumplimiento del objetivo de la tesis, creando un prototipo que tiene como estructura la categorización de máscaras de segmentación a partir de las especificaciones de detección y, por consiguiente, de contabilización propias de MN y NA [7]. Los datos de aprendizaje para la CNN [34] son las muestras tomadas por el área de investigación en Mutagénesis y Monitoreo Ambiental del IBS (UNaM-CONICET) [14].

Tercera fase:

- Elaboración de un informe acerca de los resultados obtenidos.





## Capítulo IV

### *Herramienta Bioinformática*

## Ciclo de Vida del Software en base a Metodologías Ágiles

Se recurre a una metodología ágil [82], para el desarrollo del prototipo basado en la categorización de máscaras de segmentación, construidas a partir de las especificaciones de detección y de contabilización propias de MN y NA [7].

La metodología Scrum [83], seleccionada para ello, se define como un marco para la gestión de proyectos exitoso durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Esencialmente sus características principales son dos y se ejemplifican en la Ilustración 17:

- El desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días. El resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente [83].
- La segunda característica importante son las reuniones a lo largo del proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración [82].

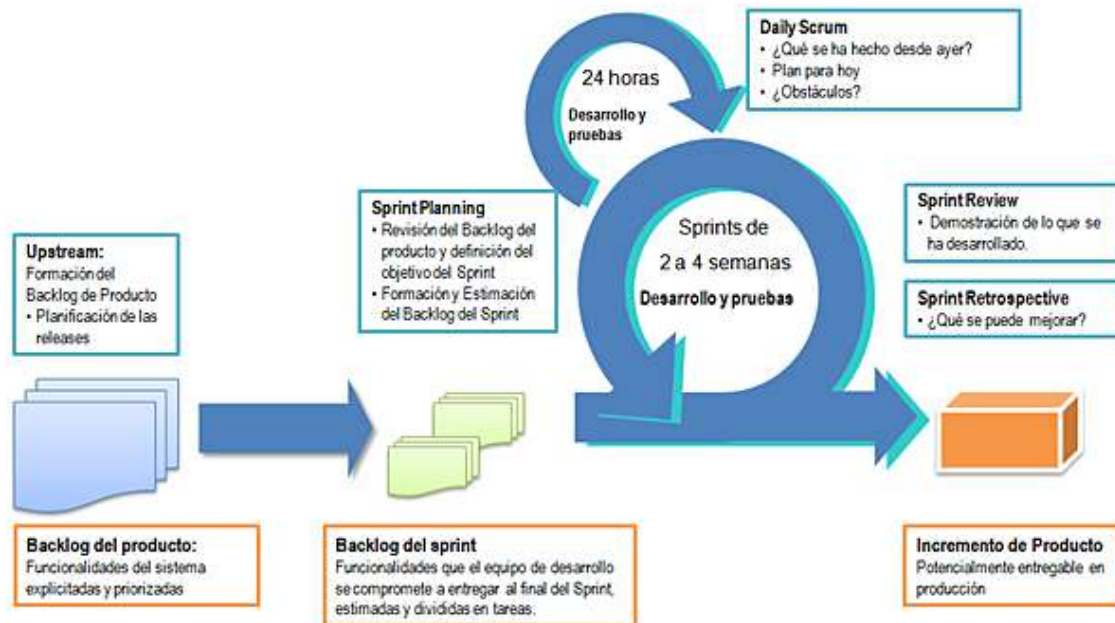


Ilustración 17. Esquema Scrum Fuente:[83].

En el estudio de caso abordado los requisitos del software en formato *sprint* son necesarios para adaptar las características específicas que necesita el investigador en las imágenes de microscopía. En la Ilustración 18 se puede observar la representación gráfica del diseño de software llevado a cabo.

El tablero de actividades realizadas, la hoja de ruta y el esquema de arquitectura de software entre otros, está disponible en el enlace [Micronúcleos Jira](#).

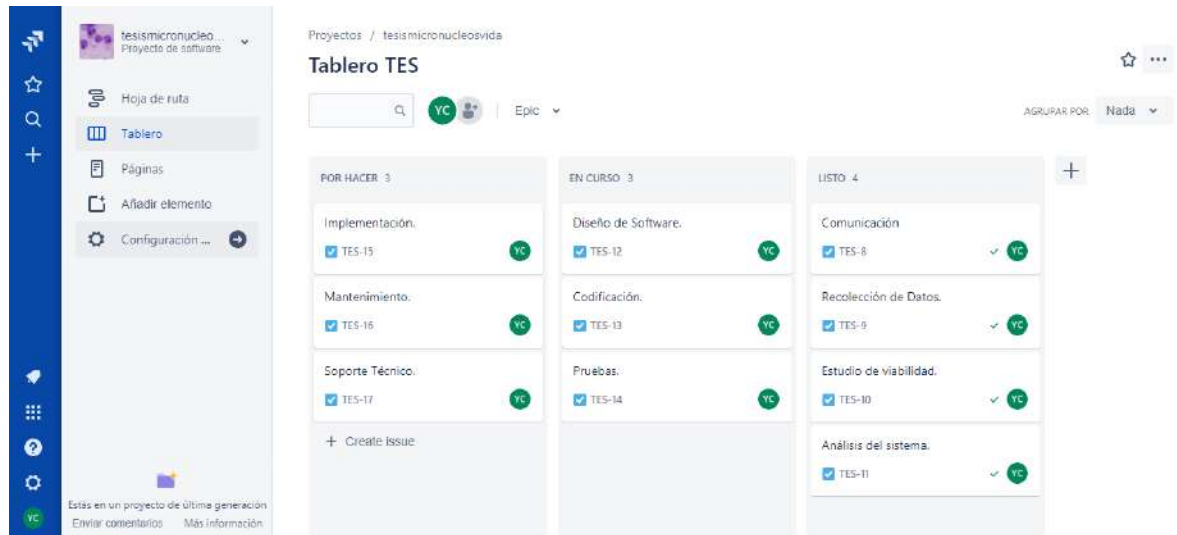


Ilustración 18. Tablero Jira

## **Especificaciones de Funcionalidad**

Los datos de aprendizaje de la CNN [34] son las muestras tomadas por el área de investigación en Mutagénesis y Monitoreo Ambiental del IBS (UNaM-CONICET) [14]. La herramienta bioinformática denominada “VGG16\_TesisMicro”, se encuentra sobre una arquitectura VGG16 [84] cuya funcionalidad principal es la capacidad de identificar tres regiones dentro de una imagen:

- Anormalidades Nucleares. (Etiqueta AN)
- Eritrocitos de Peces. (Etiqueta Peces)
- Hemocitos de Moluscos Bivalvos. (Etiqueta Moluscos)

```
In [20]: #Librerías
import sys
import os
from tensorflow.python.keras.preprocessing.image import ImageDataGenerator
from tensorflow.python.keras import optimizers
from tensorflow.python.keras.models import Sequential
from tensorflow.python.keras.layers import Dropout, Flatten, Dense, Activation
from tensorflow.python.keras.layers import Convolution2D, MaxPooling2D
from tensorflow.python.keras import backend as K
from tensorflow.python.keras import applications

In [21]: #Creación VGG16
vgg= applications.vgg16.VGG16()
```

Ilustración 19. Librerías de inicio

En la Ilustración 19 se representa el esquema de las librerías de inicio necesarias para trabajar con la herramienta diseñada en Lenguaje Python 3.8 bajo un entorno Anaconda. Entre ellas se utiliza Tensorflow [85] para el enfoque de entrenamiento e inferencia en redes neuronales profundas. De Keras [86] se importan los optimizadores, más un modelo secuencial que ordena las capas de una red neuronal y otras librerías que colaboran con el desempeño de la arquitectura VGG16 [59] incluida en la última línea de código de la Ilustración.

En la herramienta VGG16\_TesisMicro, se modifica la última capa de predicción del modelo VGG16, acordando solamente las 3 clases para la clasificación de las imágenes de entrenamiento. Las demás capas son consideradas necesarias, dado su poder computacional elevado, ya que se encuentran pre entrenadas en detección y segmentación de imágenes. De ellas se utilizan los pesos, y la activación de la nueva capa se realiza a través de Softmax como se demuestra en la Ilustración 20.

```
In [10]: #Las capas ya entrenadas de La VGG16 no vuelven a entrenarse para reutilizar el entrenamiento en la segmentación y detección de
for layer in cnn_micro.layers:
    layer.trainable=False

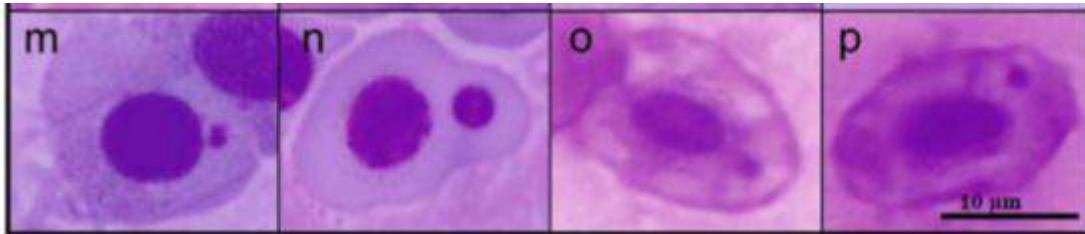
In [11]: cnn_micro.add(Dense(3,activation='softmax')) #3 clases, moluscos, peces, y anomalías nucleares.
```

Ilustración 20. Activación Softmax.

Se requieren dos carpetas contenedoras de datos con las 3 clases previamente establecidas y etiquetadas. La primera de ellas presenta los datos de entrenamiento que corresponden a las imágenes determinadas por el investigador y la segunda los datos de validación. Siendo que:

La Etiqueta Peces debe cumplir las siguientes condiciones según las especificaciones del investigador del Laboratorio:

- 1) Deben estar separados y con la misma tinción del núcleo principal,
- 2) presentar tamaño más pequeño que  $1/3$  del núcleo principal,
- 3) no mostrar refringencia.



*Ilustración 21. Eritroblastos y eritrocitos de peces para entrenamiento.*

Las imágenes *m* y *n* en la Ilustración 21, corresponden a eritroblastos o eritrocitos en proceso de maduración, que también se utilizan para el recuento de MN. Sus características fundamentales se deben a la coloración del citoplasma (más claro con respecto a los eritrocitos maduros); siendo que también difieren de éstos en orden al tamaño y forma del núcleo, presentando formas más redondeadas [78]. La Ilustración 22 representa los datos de entrenamiento para la Etiqueta Peces.

```
In [80]: fig, ax = plt.subplots(figsize=(10.6666, 10.6666))
image_peces = './data/entrenamiento/Peces/Blebed.tif'
image = io.imread(image_peces)
ax.imshow(image)

Out[80]: <matplotlib.image.AxesImage at 0x5b2afcae08>
```



Ilustración 22. Pre-entrenamiento datos “Peces”.

En el pre-procesamiento de los datos de tipo “peces” que representa la Ilustración 21, el método de segmentación propuesto para la posterior clasificación de la imagen es el método de inundación (*watershed*) con un sigma de 7 [87], cuyo objetivo es la localización de los micronúcleos que se presentan separados y con la misma tinción del núcleo principal. La detección de la variación del color es propuesta a través de la librería *Scikit-Image* [88], siendo:

El vector de color  $\phi_p$  del pixel  $p$  representado en el espacio RGB.

- El vector de color  $\phi_p$  se mapea al espacio HSV obteniendo:
 
$$\varphi_p = [h_p, s_p, v_p]$$
- Verifica si el pixel es de color “negro”: si  $v_p \leq \delta_v$  entonces  $v'_p = 0$  y  $s'_p = 0$ .
- Verifica si el pixel es de color “blanco”: si  $s_p \leq \delta_s$  entonces  $v'_p = 191$  y  $s'_p = 0$ .
- Si  $v_p > \delta_v$  y  $s_p > \delta_s$  entonces calcular el vector  $\psi_p$  siempre que sus valores oscilen de  $v_p \geq 55 \leq 95$ ,  $s_p \geq 71 \leq 95$  y  $h_p \geq 195 \leq 288$ .

En Python parametrizando *skimage.color.deltaE\_cie76 (lab1, lab2)* los colores celeste y violeta característicos de las muestras según lo determina el investigador.

- Se obtiene el vector de peso de la neurona ganadora  $w_i = [w_{i,1}, w_{i,2}]$  y se etiqueta al pixel con el número  $i$ .

Los valores  $\delta_s$  y  $\delta_v$  son umbrales para saturación e intensidad, respectivamente. Dada la naturaleza difusa del color, no hay valores específicos para decidir exactamente cuando un color es blanco o negro; de forma experimental los mejores umbrales son  $\delta_s = \mu_s - \sigma_s$  y  $\delta_v = \mu_v - \sigma_v$ ; en donde  $\mu_s$  y  $\mu_v$  son la media de saturación e intensidad de la imagen, respectivamente; siendo  $\sigma_s$  y  $\sigma_v$  la desviación estándar de la saturación e intensidad de la imagen [75].

La Etiqueta Moluscos cumple con las siguientes condiciones:

- 1) Los MN deben estar separados y con la misma tinción del núcleo principal,
- 2) presentar tamaño más pequeño que 1/3 del núcleo principal,
- 3) no mostrar refringencia,
- 4) núcleo principal en posición céntrica o hacia la periferia de la célula.



Ilustración 23. Pre-entrenamiento datos “Moluscos”.

La Ilustración 23 representa Etiqueta Moluscos. Para la segmentación se utiliza nuevamente el método por inundación (*watershed*) con un sigma de 16 [87], y el tratamiento de la morfología de la imagen desde la máscara de umbral basada en el cálculo píxeles vecinos. Siendo necesario parametrizarse: *skimage.filters.threshold\_li(image, \[\*[, ...]]*), *skimage.filters.threshold\_local(image, ...)*. Y la detección de los bordes a través de *skimage.filters.laplace(image[, ksize, mask])*.



La Etiqueta AN se demuestra cuando NO se cumplen las condiciones necesarias para que la imagen sea segmentada y caracterizada según la Etiqueta Peces y la Etiqueta Moluscos.

Para la validación, la herramienta VGG16\_TesisMicro, se utilizan las mismas imágenes del entrenamiento.

Las redes neuronales con pocas neuronas funcionan mejor en imágenes que tienen pocos colores; mientras que las redes neuronales con un número grande de colores tienen mejor desempeño con imágenes de varios colores [75]. Visto que la segmentación de instancias es la tarea de identificar contornos de objetos a nivel de píxel, tal como se propone en los modelos descritos del capítulo anterior [50][57] [58], y según los datos de entrada correspondiente a una imagen de pez, se obtiene a modo de ejemplo la Ilustración 24:

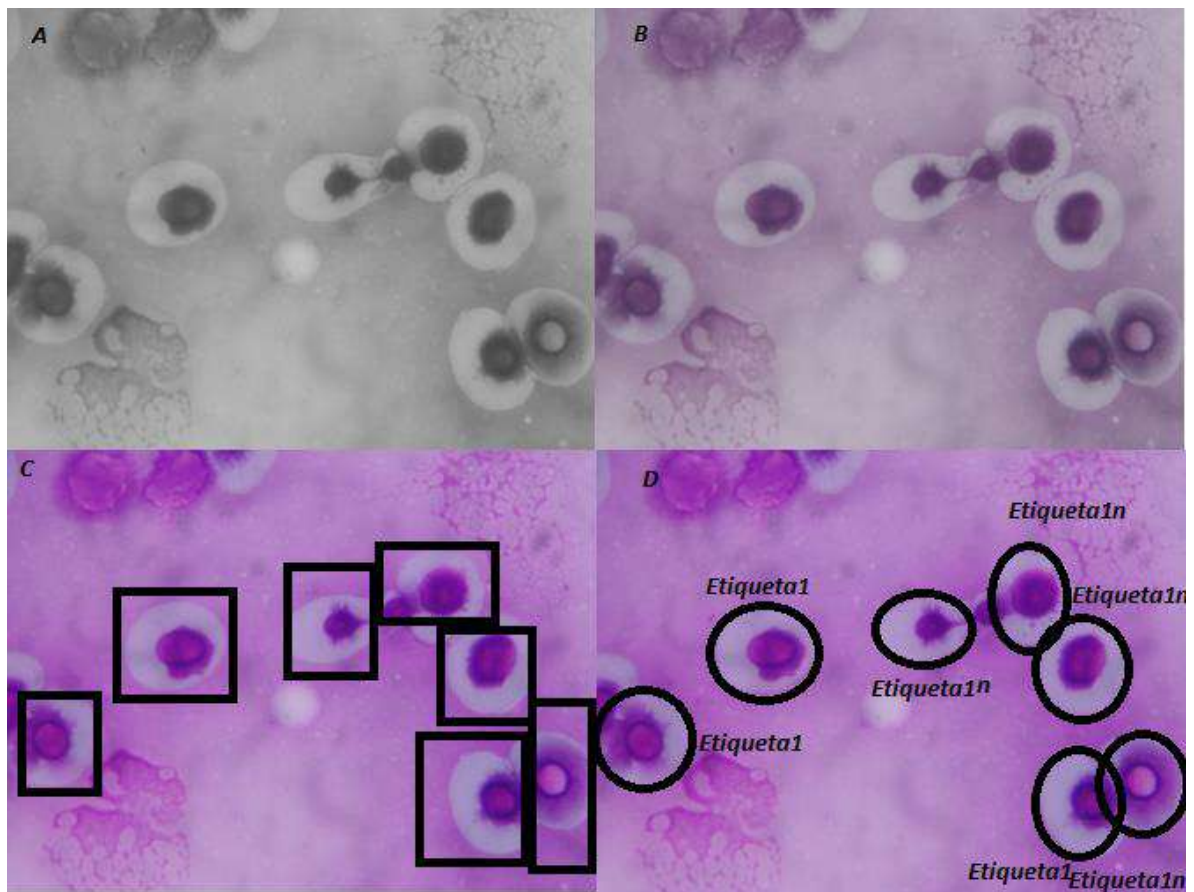


Ilustración 24. Instancia de segmentación, eritrocitos de peces.

- Clasificación según la Ilustración 17 A: la imagen presenta objetos denominados eritrocitos.
- Segmentación semántica según la Ilustración 17 B: estos son todos los píxeles que representan las imágenes de los objetos denominados eritrocitos.

- Detección de objetos según la Ilustración 17 C: son siete los objetos denominados eritrocitos en esta imagen y en estos específicos lugares.
- Segmentación de instancias según la Ilustración 17 D: son siete los objetos denominados eritrocitos en estas ubicaciones, y estos son los píxeles que pertenecen a cada uno; siendo etiquetados según los parámetros de entrenamiento dónde “EtiquetaIn” corresponde a eritroblastos designados de tipo “n”.

## Requerimientos Computacionales

- Lenguaje Python >= 3.6 [88].
- Tensorflow [85].
- Numpy.
- Keras [86].
- VGG16 [59].
- Colaboratory - Google Colab. Para la utilización de una GPU K80 [89].

## Arquitectura de Software

La Ilustración 25 constituye el diagrama estructural de la arquitectura de software:

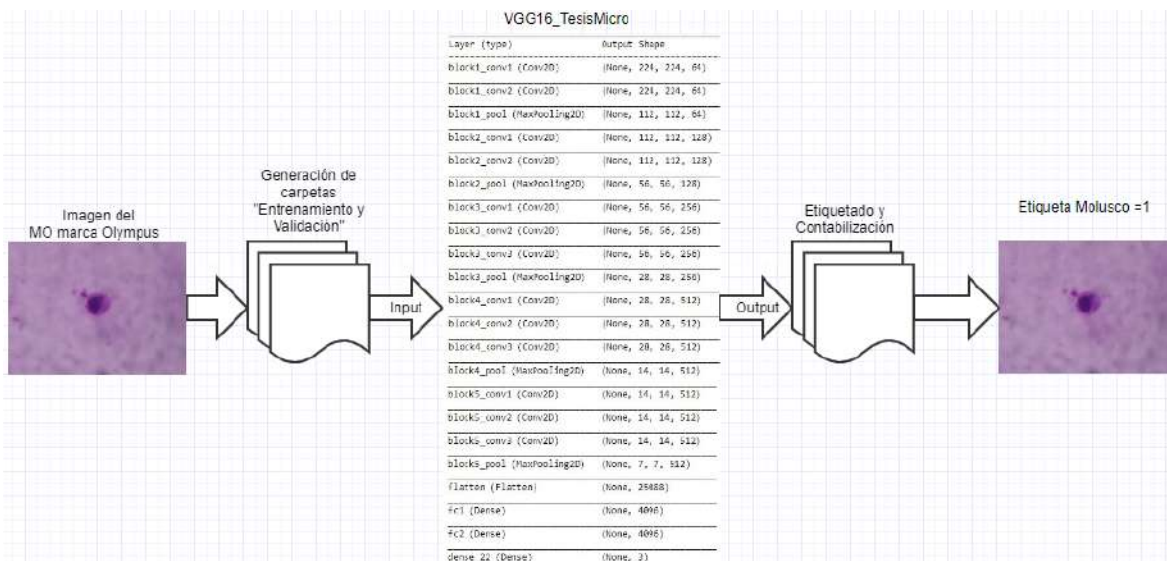


Ilustración 25. Arquitectura de Software

Esencialmente el software recibe una imagen de entrada, segmenta las zonas de interés determinando si representan a un micronúcleo aceptado por la Técnica de Micronúcleos [1]. Reconoce si la imagen es de tipo “Etiqueta Peces”, “Etiqueta Moluscos” y “Etiqueta AN” ingresando los resultados a las carpetas de Entrenamiento y Validación de la VGG16\_MicroTesis. La salida de la CNN es la contabilización de los individuos que pertenecen a cada clase parametrizada, y el aprendizaje en su detección.

### **Entrenamiento de la DCNN**

El *Broad Bioimage Benchmark Collection* (BBBC), es una colección de conjuntos de imágenes de microscopía de descarga gratuita. El BBBC facilita la posibilidad de utilizar las imágenes como puntos de referencia al desarrollar, probar y publicar nuevos algoritmos de análisis de imágenes para las ciencias de la vida [90]. La funcionalidad del BBBC representa una comprensión de qué métodos son mejores para diversas aplicaciones de análisis de imágenes biológicas. Sin embargo la misma carece de imágenes de microscopía similares a las utilizadas por el Laboratorio de Citogenética General y Monitoreo Ambiental [14], y/o parametrizadas según la Técnica de Micronúcleos [1].

Por otro lado, debido al alto coste computacional y la gran cantidad de imágenes que supone el entrenamiento de una DCNN por primera vez, se acuerda la utilización de la arquitectura VGG-16 en modo pre-entrenada a partir de un modelo de datos existente. El paquete Pycocotools, compuesto por el conjunto de datos COCO [51] podría ser una alternativa de solución, sin embargo carece de datos de referencia para el entrenamiento necesario de la herramienta VGG16MicroTesis, como ocurre con los *dataset* del BBBC.

### **Resultados obtenidos**

La posibilidad de clasificar el contenido de una imagen en bruto, del estudio de caso planteado en la presente tesis, se ve condicionado por la utilización de varios *pipeline* para el proceso de segmentación. Cada uno de éstos debe detallar metodologías y requerimientos específicos de una imagen dada.

Durante el desarrollo del software los *sprints* proporcionados al Laboratorio, generaron detalles que derivaron en modificaciones de las funcionalidades iniciales del software o *backlog del producto* [83]. Sin embargo al ingresar una nueva imagen, la segmentación se vuelve específica al ojo del investigador e imposible al *pipeline* explícito inicialmente, instruyendo así a una nueva programación y reajuste en los parámetros.

En otro orden de ideas, la carencia de *dataset* similares a los necesarios para el entrenamiento de la VGG16MicroTesis, imposibilita la elaboración de un diseño que se encargue de clasificar la imagen en “Etiqueta Peces”, “Etiqueta Moluscos”, “Etiqueta AN”.

Vale destacar que la “Etiqueta AN” debe ser considerada un *outlier* y tener un tratamiento más específico dada la extensa descripción de los parámetros que abarca, y no cómo se plantea en la actual tesis.

A su vez, tal como está desarrollada, en <https://github.com/YaninaCaffetti/TesisMicro.git> la herramienta VGG16MicroTesis, y contando con un entrenamiento de las imágenes del conjunto de datos COCO [51], puede generar etiquetas y clasificaciones de las mismas, teniéndose presente que éstas deben ubicarse en la carpeta DATA.

En la mayoría de las imágenes del *dataset* utilizado en esta tesis, el método por inundación (*watershed*) [87] no logra eficientemente la segmentación necesaria para su posterior clasificación. Razón por la cual, la segmentación inicial no define un parámetro para que el *pipeline* determine la etiqueta de clasificación necesaria en el pre-procesamiento de datos. Los filtros de morfología y detección de colores tampoco obtienen los resultados deseados para la correcta identificación de los MN.



# Capítulo V

## *Conclusiones*

## **Resultados**

Se desarrolló una revisión sistemática de la literatura científica de los actuales avances registrados en cuanto a detección y reconocimiento de patrones en imágenes, haciendo foco en los algoritmos de Inteligencia Artificial de Facebook, algoritmos evolutivos, algoritmos genéticos, y redes neuronales convolucionales. Con esta base, se creó una herramienta bioinformática que cumple con los requisitos de reconocimiento y clasificación de patrones en imágenes, pero no de imágenes de microscopía que se presentan en el estudio de caso abordado, es decir, parametrizadas según la Técnica de Micronúcleos.

Se utilizó transferencia de aprendizaje de la arquitectura VGG16 prevista en Keras [86]. Mediante modificaciones en las últimas capas, se redujo el sobre-entrenamiento y adaptó el modelo al algoritmo creado para la detección de objetos en una imagen dada.

El aprendizaje es un proceso de modificación de los pesos en respuesta a los estímulos presentes en la capa de entrada de una red neuronal, el *pipeline* que se creó no alcanza a extraer puntualmente las características particulares de las imágenes de MN y AN por lo tanto, no se logró la adaptación correcta de los pesos. El fallo fundamental está en la construcción de un *pipeline* específico por clase de muestra. Sin embargo para probar su funcionalidad, se entrenó en paralelo al modelo VGG16MicroTesis con el conjunto de datos COCO [51], modificando las imágenes de entrada en la carpeta DATA. Con las modificaciones, se logró la clasificación eficiente de objetos con las etiquetas “árbol”, “persona”, “caballo” por ejemplo.

Los siguientes pasos para avanzar con el proyecto son, sin lugar a dudas, dar mayor difusión a lo realizado a fin de que, con la exposición pública y a través de publicaciones científicas, se logre crear un *dataset* con los *pipeline* correctos y necesarios para detectar los MN y AN en las imágenes de microscopía.

La presente investigación generó en la actualidad, dos publicaciones científicas, una de ellas en la XX Workshop de Investigadores en Ciencias de la Computación (WICC 2018, Universidad Nacional del Nordeste) disponible en <http://sedici.unlp.edu.ar/handle/10915/67490> y la otra en I Congreso Internacional de Ciencia, Diseño y Tecnología de la Universidad Autónoma de Encarnación (UNAE) disponible en [Academia\\_FACAT2019](#).

### **Futuras líneas de investigación**

Se presenta como una posible vía de superación ante los inconvenientes señalados, la eventual contribución a la creación de un *dataset* específico de MN y AN, en el *Broad Bioimage Benchmark Collection*, alternativa que se encuentra perfectamente prevista en su portal web a modo de contribución.

En ese sentido, la herramienta de Inteligencia Artificial de Facebook denominada Dectectron2, modificada en <https://colab.research.google.com/drive/13OMU6yy8yadkSbaAdJ5y3MCHJWOXb4oq> representa una opción más, tendiente a corregir el inconveniente de detección y segmentación en las imágenes de MN y AN. Con ese objetivo la presente investigación ha sido postulada a la *Solicitud de propuestas de sistemas para aprendizaje automático* extendida por Facebook IA en <https://research.fb.com/programs/research-awards/proposals/systems-for-machine-learning-request-for-proposals/>.

Por último, las publicaciones recientes acerca de investigaciones llevadas a cabo de imágenes de microscopía utilizando aprendizaje profundo, sugieren la posibilidad de adaptar la metodología de análisis al *dataset* de la presente investigación. Sin embargo, hasta el momento no se encuentran propuestas similares dedicadas al estudio de imágenes de MN en peces y moluscos.





# Bibliografía

- [1] W. Schmid, "The micronucleus test," *Mutat. Res. Mutagen. Relat. Subj.*, vol. 31, no. 1, pp. 9–15, Feb. 1975.
- [2] Z. Al Aghbari and R. Al-Haj, "Hill-manipulation: An effective algorithm for color image segmentation," *Image Vis. Comput.*, vol. 24, no. 8, pp. 894–903, Aug. 2006.
- [3] B. Somayeh Mousavi, F. Soleymani, and N. Razmjooy, "Color image segmentation using neuro-fuzzy system in a novel optimized color space," *Neural Comput. Appl.*, vol. 23, no. 5, pp. 1513–1520, Oct. 2013.
- [4] L. Wang and M. Dong, "Multi-level Low-rank Approximation-based Spectral Clustering for image segmentation," *Pattern Recognit. Lett.*, vol. 33, no. 16, pp. 2206–2215, Dec. 2012.
- [5] R. Katz and C. Delrieux, "Estrategias evolutivas para la detección de contornos en imágenes digitales," *V Work. Investig. en Ciencias la Comput.*, pp. 251–255, 2003.
- [6] C. A. Arango Londoño, A. M. Hualpa Zúñiga, D. K. Reina Moreno, and M. N. Roncancio Ávila, "Construcción E Implementación De Un Estudio De Caso Como Herramienta De Apoyo a Los Procesos De Aprendizaje En La Asignatura De Producción Desde Un Enfoque Constructivista," *Ingeniare*, no. 21, p. 109, Dec. 2016.
- [7] G. N. A. Furnus, J. D. Caffetti, E. M. García, M. F. Benítez, M. C. Pastori, and A. S. Fenocchio, "Frequências basais de micronúcleos e anormalidades nucleares em peixes nativos do Rio Paraná (Argentina)," *Brazilian J. Biol.*, vol. 74, no. 1, pp. 217–221, Feb. 2014.
- [8] J. D. Caffetti, M. S. Mantovani, M. C. Pastori, and A. S. Fernocchio, "First genotoxicity study of Paraná River water from Argentina using cells from the clam *Corbicula fluminea* (veneroida corbiculidae) and Chinese hamster (Cricetulus griseus Rodentia, Cricetidae) K1 cells in the comet assay," *Genet. Mol. Biol.*, vol. 31, no. 2, pp. 561–565, 2008.
- [9] F. A. Leveroni, J. D. Caffetti, and M. C. Pastori, "Genotoxic response of blood, gill and liver cells of *Piaractus mesopotamicus* after an acute exposure to a glyphosate-based herbicide," *Caryologia*, vol. 70, no. 1, pp. 21–28, Jan. 2017.
- [10] W. A. Ramsdorf, T. Vicari, M. I. M. de Almeida, R. F. Artoni, and M. M. Cestari, "Handling of *Astyanax* sp. for biomonitoring in Cangüiri Farm within a fountainhead (Iraí River Environment Preservation Area) through the use of genetic biomarkers," *Environ. Monit. Assess.*, vol. 184, no. 10, pp. 5841–5849, Oct. 2012.
- [11] A. N. Jha, "Ecotoxicological applications and significance of the comet assay," *Mutagenesis*, vol. 23, no. 3, pp. 207–221, Feb. 2008.
- [12] M. Fenech, "Cytokinesis-block micronucleus cytome assay," *Nat. Protoc.*, vol. 2, no. 5, pp. 1084–1104, May 2007.
- [13] J. Baršienė, A. Rybakovas, T. Lang, L. Andreikėnaitė, and A. Michailovas,

- “Environmental genotoxicity and cytotoxicity levels in fish from the North Sea offshore region and Atlantic coastal waters.,” *Mar. Pollut. Bull.*, vol. 68, no. 1–2, pp. 106–16, Mar. 2013.
- [14] C.-Un. IBS, “Acerca del.” [Online]. Available: <https://ibs.conicet.gov.ar/>. [Accessed: 02-Sep-2019].
- [15] J. A. Heddle, M. Fenech, M. Hayashi, and J. T. MacGregor, “Reflections on the development of micronucleus assays,” *Mutagenesis*, vol. 26, no. 1, pp. 3–10, Jan. 2011.
- [16] C. Díaz Antonio, M. González Sergio, and E. Valente Francisco, “La prueba de micronucleos en sangre como bioindicador de genotoxicos,” *Abanico Vet.*, vol. 2, no. 2, 2012.
- [17] M. Kirsch-Volders, G. Plas, and A. Elhajouji, “The in vitro MN assay in 2011: Origin and fate, biological significance, protocols, high throughput methodologies and toxicological relevance,” *Arch. Toxicol.*, vol. 85, no. 8, pp. 873–899, Aug. 2011.
- [18] A. Karpathy, “CS231n Convolutional Neural Networks for Visual Recognition,” *Stanford University*, 2016. [Online]. Available: <http://cs231n.github.io/>. [Accessed: 11-Sep-2019].
- [19] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Deep, big, simple neural nets for handwritten digit recognition,” *Neural Computation*, vol. 22, no. 12, pp. 3207–3220, Dec-2010.
- [20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.
- [21] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [22] Y. Bengio, “Learning Deep Architectures for AI,” *Found. Trends® Mach. Learn.*, vol. 2, no. 1, pp. 1–127, Nov. 2009.
- [23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *ImageNet Large Scale Visual Recognition Challenge*, “*Int. J. Comput. Vis.*,” vol. 115, no. 3, pp. 211–252, Dec. 2015.
- [24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012, vol. 2, pp. 1097–1105.
- [25] M. D. Zeiler and R. Fergus, “Stochastic Pooling for Regularization of Deep Convolutional Neural Networks,” *arXiv:1301.3557v1*, p. 9, Jan. 2013.
- [26] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, “Multi-scale orderless pooling of deep convolutional activation features,” in *Lecture Notes in Computer Science (including*

- subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*), 2014, vol. 8695 LNCS, no. PART 7, pp. 392–407.
- [27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 2818–2826.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 770–778.
- [29] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical Evaluation of Rectified Activations in Convolutional Network,” *arXiv:1505.00853v2*, May 2015.
- [30] Y. Tang, “Deep Learning using Linear Support Vector Machines,” *arXiv:1306.0239v4*, Jun. 2013.
- [31] Q. Zhao and L. D. Griffin, “Suppressing the Unusual: towards Robust CNNs using Symmetric Activation Functions,” *arXiv:1603.05145v1*, p. 11, Mar. 2016.
- [32] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv:1207.0580*, Jul. 2012.
- [33] Y. Glorot, X., Bordes, A., & Bengio, “Deep Sparse Rectifier Neural Networks,” in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics.*, 2011, pp. 315–323.
- [34] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. S. Kruthiventi, and R. V. Babu, “A taxonomy of deep convolutional neural nets for computer vision,” *Front. Robot. AI*, vol. 2, no. JAN, p. 36, Jan. 2016.
- [35] W. Rawat and Z. Wang, “Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017.
- [36] D. Yu, H. Wang, P. Chen, and Z. Wei, “Mixed pooling for convolutional neural networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8818, pp. 364–375.
- [37] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv:1409.1556*, Sep. 2014.
- [38] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, “Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition,” in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
- [39] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal

- margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*, 1992, pp. 144–152.
- [40] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “High-Performance Neural Networks for Visual Object Classification,” *arXiv:1102.0183*, p. 12, Feb. 2011.
- [41] H. Coates, Adam, Ng, Andrew Y., and Lee, “An Analysis of Single-Layer Networks in Unsupervised Feature Learning,” *Proc. 14th Int. Conf. Artif. Intell. Stat. 2011, Fort Lauderdale, FL, USA. Vol. 15 JMLR W&CP 15.*, 2011.
- [42] R. Collobert and S. Bengio, “A gentle Hessian for efficient gradient descent,” in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. V-517–20.
- [43] G. E. Dahl, M. Ranzato, A. R. Mohamed, and G. Hinton, “Phone recognition with the mean-covariance restricted Boltzmann machine,” in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, 2010.
- [44] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality Reduction by Learning an Invariant Mapping,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, 2006, vol. 2, pp. 1735–1742.
- [45] G. E. Hinton, “Connectionist learning procedures,” *Artif. Intell.*, vol. 40, no. 1–3, pp. 185–234, Sep. 1989.
- [46] T. D. Nguyen, A. Shinya, T. Harada, and R. Thawonmas, “Segmentation Mask Refinement Using Image Transformations,” *IEEE Access*, vol. 5, pp. 26409–26418, 2017.
- [47] J. Dai, K. He, and J. Sun, “Instance-Aware Semantic Segmentation via Multi-task Network Cascades,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3150–3158.
- [48] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, “Simultaneous detection and segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8695 LNCS, no. PART 7, pp. 297–312.
- [49] P. O. Pinheiro, R. Collobert, and P. Dollár, “Learning to segment object candidates,” in *Advances in Neural Information Processing Systems*, 2015, vol. 2015-Janua, pp. 1990–1998.
- [50] P. O. Pinheiro, T. Y. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9905 LNCS, pp. 75–91.
- [51] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár and L.

- Zitnick, “Microsoft COCO: Common objects in context,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693 LNCS, no. PART 5, pp. 740–755.
- [52] D. Eigen and R. Fergus, “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2650–2658.
- [53] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P.V.D. Smagt, D. Cremers and T. Brox, “FlowNet: Learning optical flow with convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, vol. 2015 Inter, pp. 2758–2766.
- [54] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, “Learning Hierarchical Features for Scene Labeling,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.
- [55] J. Hosang, R. Benenson, P. Dollar, and B. Schiele, “What Makes for Effective Detection Proposals?,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 814–830, Apr. 2016.
- [56] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.
- [57] H. Hu, S. Lan, Y. Jiang, Z. Cao, and F. Sha, “FastMask: Segment multi-scale object candidates in one shot,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-Janua, pp. 2280–2288.
- [58] S. Zagoruyko, A. Lerer, T.Y. Lin, P.O. Pinheiro, S. Gross, S. Chintala and P. Dollár, “A multipath network for object detection,” in *British Machine Vision Conference 2016, BMVC 2016*, 2016, vol. 2016-Sept, pp. 15.1-15.12.
- [59] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, vol. 2015 Inter, pp. 1440–1448.
- [60] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” *2009 IEEE Conf. Comput. Vis. Pattern Recognition, Miami, FL.*, pp. 248–255, 2009.
- [61] A. Torralba, “Contextual Priming for Object Detection,” *Int. J. Comput. Vis.*, vol. 53, no. 2, pp. 169–191, 2003.
- [62] S. Gidaris and N. Komodakis, “Object detection via a multi-region and semantic segmentation-aware U model,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 Inter, pp. 1134–1142, May 2015.
- [63] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, “Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks,” in

*Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-Decem, pp. 2874–2883.

- [64] E. Aldabas Rubira, “Introducción al reconocimiento de patrones mediante redes neuronales,” IX Jornades de Conferències d’Enginyeria Electrònica del Campus de Terrassa, Terrassa, España, Dic. 2002.
- [65] R. Malhotra, N. Singh, and Y. Singh, “Genetic Algorithms: Concepts, Design for Optimization of Process Controllers,” *Comput. Inf. Sci.*, vol. 4, no. 2, Feb. 2011.
- [66] K. Sastry, D. Goldberg, and G. Kendall, “Genetic Algorithms,” in *Search Methodologies*, Boston, MA: Springer US, 2005, pp. 97–125.
- [67] P. E. Valencia, “Optimización mediante algoritmos genéticos,” *An. del Inst. Ing. Chile*, pp. 83–92, 1997.
- [68] F. Xing and L. Yang, “Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A comprehensive review,” *IEEE Rev. Biomed. Eng.*, vol. 9, pp. 234–263, 2016.
- [69] M. Z. Alom *et al.*, “Advanced Deep Convolutional Neural Network Approaches for Digital Pathology Image Analysis: a comprehensive evaluation with different use cases,” Apr. 2019.
- [70] I. Decordier, A. Papine, K. Vande Loock, G. Plas, F. Soussaline, and M. Kirsch-Volders, “Automated image analysis of micronuclei by IMSTAR for biomonitoring,” *Mutagenesis*, vol. 26, no. 1, pp. 163–168, Jan. 2011.
- [71] S. Bonassi, F. M Lando, C. Lin, Y.P. Ceppi, M. Chang WP, N. Holland, M. Kirsch-Volders, E. Zeiger, S. Ban, R. Barale, M.P. Bigatti, C. Bolognesi, C. Jia, M. Di Giorgio, L.R. Ferguson, A. Fucic, O.G. Lima, P. Hrelia, A.P. Krishnaja, T.K. Lee, L. Migliore, L. Mikhalevich, E. Mirkova, P. Mosesso, W.U. Müller, Y. Odagiri, M.R. Scarffi, E. Szabova, I. Vorobtsova, A. Vral, A. Zijno, “HUMAN MicroNucleus project: international database comparison for results with the cytokinesis-block micronucleus assay in human lymphocytes: I. Effect of laboratory protocol, scoring criteria, and host factors on the frequency of micronuclei,” *Environ. Mol. Mutagen.*, vol. 37, no. 1, pp. 31–45, 2001.
- [72] I. Decordier, A. Papine, G. Plas, S. Roesems, K. Vande Loock, J. Moreno-Palomo, E. Cemeli, D. Anderson, A. Fucic, R. Marcos, F. Soussaline, M. Kirsch-Volders, “Automated image analysis of cytokinesis-blocked micronuclei: An adapted protocol and a validated scoring procedure for biomonitoring,” *Mutagenesis*, vol. 24, no. 1, pp. 85–93, 2009.
- [73] M. Castrillón Santana, *Libro: Fundamentos de informática y programación para ingeniería: Ejercicios resueltos para C y Matlab. ISBN 9788497328463*. Paraninfo, 2011.
- [74] R. E. Gonzalez, Rafael C. Woods, *Libro: Digital image processing. ISBN 9780201180756*, Prentice H. 2002.



- [75] M. C. Murillo, F. G. Lamont, and U. Autónoma, “Imitando la percepción humana del color Color Image Segmentation by Mimicking the Human Perception of Color,” *Res. Comput. Sci.*, vol. 114, pp. 71–81, 2016.
- [76] T. Ohe, T. Watanabe, and K. Wakabayashi, “Mutagens in surface waters: a review.,” *Mutat. Res.*, vol. 567, no. 2–3, pp. 109–49, Nov. 2004.
- [77] R. van der Oost, J. Beyer, and N. P. E. Vermeulen, “Fish bioaccumulation and biomarkers in environmental risk assessment: a review.,” *Environ. Toxicol. Pharmacol.*, vol. 13, no. 2, pp. 57–149, Feb. 2003.
- [78] J. D. Caffetti, “Estudio Del Potencial Genotóxico De Aguas Del Río Paraná Mediante Ensayo Cometa Y Micronúcleos in Vivo E in Vitro,” Tesis Doctoral .Universidad Nacional de Córdoba, Argentina, 2014.
- [79] M. Maldonado, “Evaluación del potencial citotóxico y genotóxico de muestras de agua de arroyos de la ciudad de Posadas (Misiones, Argentina) mediante los bioindicadores *Allium cepa* y *Phaloceros caudimaculatus*.,” Tesis de Licenciatura en Genética. Universidad Nacional de Misiones, Argentina, 2017.
- [80] R. N. Hooftman and W. K. de Raat, “Induction of nuclear anomalies (micronuclei) in the peripheral blood erythrocytes of the eastern mudminnow *Umbra pygmaea* by ethyl methanesulphonate,” *Mutat. Res. Lett.*, vol. 104, no. 1–3, pp. 147–152, Apr. 1982.
- [81] K. R. Carrasco, K. L. Tilbury, and M. S. Myers, “Assessment of the Piscine Micronucleus Test as an in situ Biological indicator of Chemical Contaminant Effects,” *Can. J. Fish. Aquat. Sci.*, vol. 47, no. 11, pp. 2123–2136, Nov. 1990.
- [82] P. Letelier, C. Penadés, J. Canós, and E. Sánchez, “Metodologías Ágiles en el Desarrollo de Software,” *Val. Val.*, p. 59, 2009.
- [83] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*, vol. 18. 2001.
- [84] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [85] M. Abadi, Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng “TensorFlow: A system for large-scale machine learning,” *ArXiv 1605.08695*, 2016.
- [86] A. Gulli and S. Pal, “Deep Learning with Keras - Antonio Gulli, Sujit Pal - Google Libros,” *Packt Publishing Ltd*, 2017. [Online]. Available: <https://books.google.es/books?hl=es&lr=&id=20EwDwAAQBAJ&oi=fnd&pg=PP1&dq=define+keras&ots=IHfx8jdSV8&sig=uDuBITqXJvtg7C4Klo6FLqex768#v=onepage&q=define+keras&f=false>. [Accessed: 30-Oct-2019].
- [87] N. La, S. Palomino, and L. P. Concepción, “Watershed: un algoritmo eficiente y

flexible para segmentación de imágenes de geles 2-DE,” *Rev. Investig. Sist. E Informática - Univ. Nac. Mayor San Marcos*, vol. 7, no. 2, p. 7, 2010.

- [88] S. Lynch and S. Lynch, “Image Processing with Python,” in *Dynamical Systems with Applications using Python*. ISBN 978-3-319-78144-0, Springer International Publishing, 2018, pp. 471–489.
- [89] E. Bisong and E. Bisong, “Google Colaboratory,” in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. ISBN 978-1-4842-4469-2, Apress, 2019, pp. 59–64.
- [90] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, “Annotated high-throughput microscopy image sets for validation,” *Nature Methods*, vol. 9, no. 7. p. 637, Jul-2012.