

Trabajo Fin de Grado.

Grado en Tecnologías de la Ingeniería de
Telecomunicación.

Caracterización de la Trisomía 21 o Síndrome de
Down mediante descriptores texturales clásicos
aplicados a imágenes de hipocampo murino.

Autor: Rafael Eugenio López Zaragoza

Tutor: Irene Fondón García

María Auxiliadora Sarmiento Vega

Dpto. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2015



Trabajo Fin de Grado
Grado en Tecnologías de la Ingeniería de Telecomunicación

**Caracterización de la Trisomía 21 o Síndrome
de Down mediante descriptores texturales
clásicos aplicados a imágenes de hipocampo
murino.**

Autor:

Rafael Eugenio López Zaragoza

Tutor:

Irene Fondón García

Profesora Contratada Doctora

María Auxiliadora Sarmiento Vega

Profesora Contratada Doctora Interina

Departamento de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2015

Trabajo Fin de Grado: Caracterización de la Trisomía 21 o Síndrome de Down mediante descriptores texturales clásicos aplicados a imágenes de hipocampo murino.

Autor: Rafael Eugenio López Zaragoza

Tutor: Irene Fondón García

María Auxiliadora Sarmiento Vega

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

Agradecimientos

A mi familia y mi novia por el apoyo incondicional en todo momento.

A mis compañeros, porque sin ellos el camino no hubiese sido lo mismo.

Y por supuesto, a mis tutoras Uxi e Irene por confiar en mi cuando nadie lo hizo y aguantar mi carácter durante el desarrollo de este gran proyecto.

Índice

1. Introducción	13
1.1. Contexto	13
1.2. Identificación del problema	13
1.3. Objetivos	14
2. Fundamentos fisiológicos	15
2.1. ¿Qué es la neurona?	15
2.2. ¿Qué es el hipocampo?	16
2.3. Patrones dendríticos	18
3. Estado del arte	21
3.1. Textura	21
3.1.1. Análisis de texturas	22
3.1.1.1. Métodos estadísticos	22
3.1.1.2. Métodos espectrales	26
3.1.1.3. Métodos estructurales	27
3.1.1.4. Local Binary Patterns	28
3.2. Descriptores de texturas en imágenes biomédicas	30
3.3. Clasificadores	31
3.3.1. k-Nearest Neighbor	31
3.3.2. Redes neuronales	34
3.3.3. Support Vector Machine (SVM)	36
3.3.4. Árboles de decisión. Random Forest	39
3.3.4.1. Random Forest	40
4. Algoritmo implementado	41
4.1. Selección de imágenes	42
4.2. Pre-procesado	50
4.3. Extracción de características	58

4.4.	Entrenamiento	60
4.5.	Clasificación	61
5.	Resultados.....	62
5.1.	Grupo entrenamiento de 6 muestras	63
5.2.	Grupo entrenamiento de 8 muestras	71
6.	Conclusiones y líneas futuras	79
8.	Anexo I: ROI de imágenes originales	87
9.	Anexo II: ROI con FI Inpainting.....	91
10.	Anexo III: Imágenes finales	94

Índice de Figuras

Figura 2.1. Partes principales neurona multipolar. [1]	15
Figura 2.2. Ejemplos tinción de Golgi. [3].....	16
Figura 2.3 Localización del hipocampo en cerebro humano [4].	16
Figura 2.4. Imagen de inmunofluorescencia MAP2. División del hipocampo.....	17
Figura 2.5. Patrón dendritico para muestras Wild Type.....	18
Figura 2.6. Patrón dendritico para muestras trisómicas.	19
Figura 3.1. Ejemplos de texturas. (a) Madera. (b) Pared de ladrillos.	21
(c) Metal. (d) Tejido algodón.	21
Figura 3.2. Ejemplo de histograma.	22
Figura 3.3. Distancia y orientación de los pares [5].....	23
Figura 3.4. Matriz de co-ocurrencia [0,1] de I [5].	24
Figure 3.5. Espectro en frecuencia de tejido.....	26
Figura 3.6. Tipos de primitivas: (a) de repetición aleatoria (b) de estructura regular (c) de repetición regular	27
Figura 3.7. Ejemplo de geometría fractales.	27
Figura 3.8. Ejemplo de algoritmo LBP para una vecindad 3x3 [11].	28
Figura 3.9 Ejemplos de vecindarios multiescala [12].....	29
Figura 3.10. Ejemplo de código LBP en multiescala [13].	29
Figura 3.11. Ejemplo espacio de muestras k-NN [31].....	32
Figura 3.12. Red neuronal [32].	34
Figura 3.13. Hiperplano y margen caso 2D [35].	36
Figura 3.14. Ejemplo de transformación polinomial [36].....	38
Figura 3.15. Ejemplo de transformación radial [37].	38
Figura 3.16. Árbol decisión: condición contratación seguro de automóvil.	39
Figura 4.1. Esquema de formación de clasificador.....	41
Figura 4.2. Muestra hipocampo para el plano Z=1.	43

Figura 4.3. Muestra hipocampo para el plano Z=2.....	43
Figura 4.4. Muestra hipocampo para el plano Z=3.....	44
Figura 4.5. Muestra hipocampo para el plano Z=4.....	44
Figura 4.6. Muestra hipocampo para el plano Z=5.....	45
Figura 4.7. Muestra hipocampo para el plano Z=6.....	45
Figura 4.8. Muestra hipocampo para el plano Z=7.....	46
Figura 4.9. Muestra hipocampo para el plano Z=8.....	46
Figura 4.10. Muestra hipocampo para el plano Z=9.....	47
Figura 4.11. Muestra hipocampo para el plano Z=10.....	47
Figura 4.12. Muestra hipocampo para el plano Z=11.....	48
Figura 4.13. Muestra hipocampo para el plano Z=12.....	48
Figura 4.14. Ejemplo de imagen coronal de hipocampo.....	49
Figura 4.15. Imagen descartada por falta de nitidez.....	49
Figura 4.16. Ejemplo de diferencia de luminosidad entre muestras.....	50
Figura 4.17. Ejemplo de muestras con cruce de vasos sanguíneos.....	50
Figura 4.18. Esquema pre-procesado.....	51
Figura 4.19. Aproximación a la región de interés.....	52
Figura 4.20. ROI trisómica.....	53
Figura 4.21. ROI wild type.....	53
Figura 4.22. Aplicación Fast Image Inpainting I [42].....	53
Figura 4.23. Aplicación Fast Image Inpainting II [42].....	53
Figura 4.24. Efecto vasos sanguíneos.....	54
.Figura 4.25. Resultados Fast Image Inpainting.....	55
Figura 4.27. Ejemplo de ecualización de histograma [43].....	56
Figura 4.26. Ejemplo limitación del contraste en CLAHE [45].....	56
Figura 4.27. ROI trisómica final.....	57
Figura 4.28. ROI wild type final.....	57
Figura 4.29. Ordenación [46].....	59
Figura 5.1 Modelo tabla de confusión.....	62
Figura 5.2. Curva ROC clase TR.....	69
Figura 5.3. Curva ROC clase WT.....	70
Figura 5.4. Curva ROC clase TR.....	77
Figura 5.5. Curva ROC clase WT.....	78

1. Introducción

1.1. Contexto

Este proyecto se encuadra dentro de la colaboración entre las profesoras del Departamento de Teoría de la Señal y Comunicaciones, Irene Fondón y M^a Auxiliadora Sarmiento, y el grupo científico encabezado por M^a Luz Montesinos.

M^a Luz Montesinos es profesora titular del departamento de Fisiología Médica y Biofísica y responsable del Laboratorio de Traducción Local Sináptica.

Este grupo de investigación está especializado en el estudio de la traducción local sináptica y en 2010 patentaron el uso de rapamicina para el tratamiento del déficit cognitivo asociado al Síndrome de Down.

1.2. Identificación del problema

La discapacidad intelectual y la neurodegeneración se relacionan con defectos de morfología neuronal, presentes en el Síndrome de Down.

El Síndrome de Down es una alteración genética que se produce por la presencia de un cromosoma. Las células del cuerpo humano tienen 46 cromosomas distribuidos en 23 pares. Uno de estos pares determina el sexo del individuo, los otros 22 se numeran del 1 al 22 en función de su tamaño decreciente. Las personas con síndrome de Down tienen tres cromosomas en el par 21 en lugar de los dos que existen habitualmente; por ello, este síndrome también se conoce como trisomía 21.

El síndrome de Down es la principal causa de discapacidad intelectual y la alteración genética humana más común. Tiene lugar de forma espontánea, sin que exista una causa aparente sobre la que se pueda actuar para impedirlo. Se produce en todas las etnias, en todos los países, con una incidencia de una por cada 600-700 concepciones en el mundo.

Según la Encuesta de Discapacidad, Autonomía Personal y Situaciones de Dependencia (EDAD 2008) realizada por el Instituto Nacional de Estadística (INE), se puede estimar que hasta el año 2008 en España existían 34.000 personas con Síndrome de Down.

La motivación de este proyecto reside en encontrar una característica matemática que sea capaz de cuantificar las alteraciones morfológicas neuronales y que permita de esta forma contribuir en el progreso de investigaciones que en un futuro pueden servir a mejorar la calidad de vida de las personas con Síndrome de Down.

1.3. Objetivos

El grupo científico tenía la necesidad de transformar los patrones neuronales en un modelo matemático para automatizar su clasificación. Gracias a su larga experiencia en el procesamiento de imágenes, Irene Fondón y M^a Auxiliadora Sarmiento llegaron a la conclusión de que podían aplicar métodos de tratamiento digital en esas imágenes.

El objetivo de este proyecto es:

- El estudio de descriptores texturales clásicos en imágenes coronales de hipocampo murino.
- El estudio de clasificadores para la clasificación de arborización dendrítica.
- La clasificación automática de patrones dendríticos.

2. Fundamentos fisiológicos

2.1. ¿Qué es la neurona?

La neurona es la unidad elemental de procesamiento y transmisión de la información en el sistema nervioso. Constan de cuatro regiones diferenciadas: soma, dendritas, axón y botones terminales.

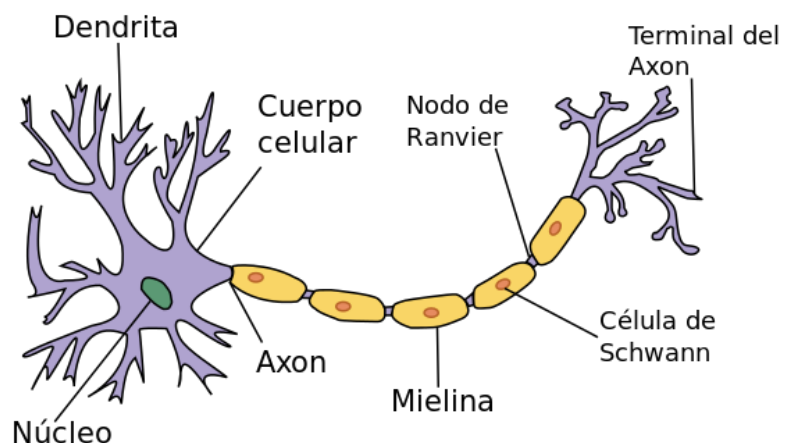


Figura 2.1. Partes principales neurona multipolar. [1]

El soma contiene el núcleo y gran parte de la maquinaria que posibilita los procesos vitales de la célula. Las dendritas son receptores en la comunicación entre neuronas y el axón se encarga de transmitir la información desde el soma hasta los botones terminales. [2]

En ocasiones, se realizan estudios de arborización dendrítica analizando individualmente varias neuronas de un mismo sujeto. Este proceso se puede llevar a cabo mediante la tinción de Golgi en tejidos neuronales y la visualización de cultivos neuronales.

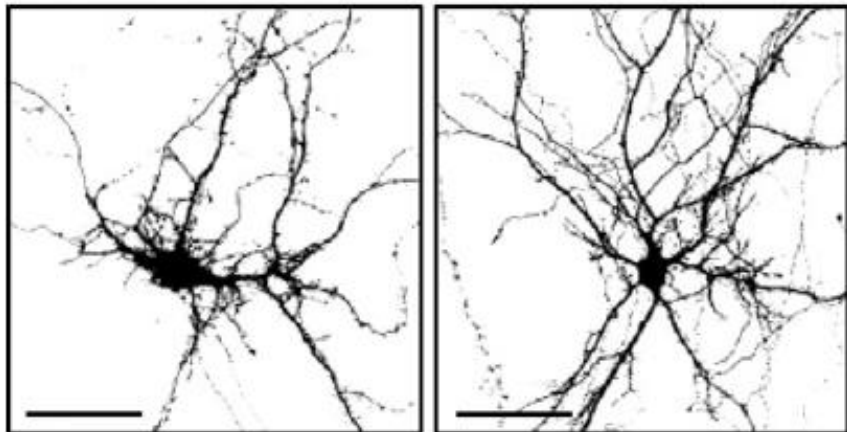


Figura 2.2. Ejemplos tinción de Golgi. [3]

Se desconoce el motivo, pero la tinción de Golgi sólo aísla un reducido número de neuronas (en torno al 5-10%). Esta característica puede tener ventajas en estudios individuales, pero no en nuestro caso. De este modo, los cultivos neuronales son modelos simplificados de lo que realmente ocurre con las neuronas en el sistema nervioso.

2.2. ¿Qué es el hipocampo?

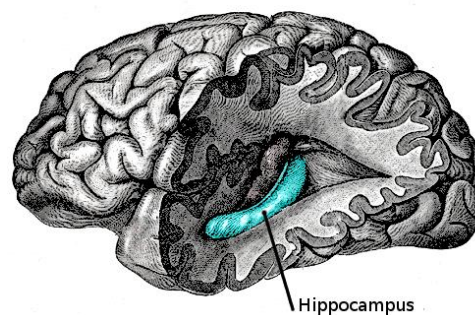


Figura 2.3 Localización del hipocampo en cerebro humano [4].

El hipocampo es una estructura prosencefálica del lóbulo temporal que constituye una parte importante del sistema límbico. Incluye al propio hipocampo, el giro dentado y el subículo. Se conoce que algunas regiones del sistema límbico, sobre todo la formación hipocámpal, están implicadas en el aprendizaje y la memoria [2].

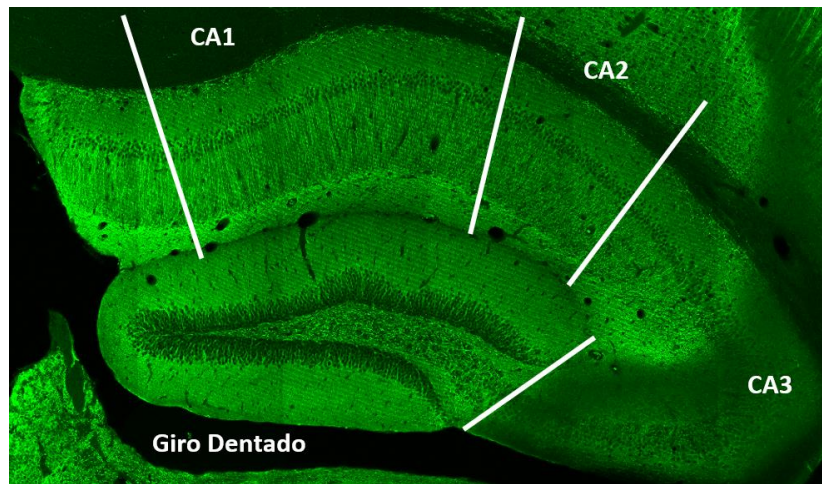


Figura 2.4. Imagen de inmunofluorescencia MAP2. División del hipocampo.

Como se puede ver en la figura 2.4, el hipocampo se divide en cuatro zonas fundamentales: los campos CA1, CA2, CA3 y CA4. El campo CA3 recibe aferencias del giro dentado y las proyecta al campo CA1. El campo CA1 las recibe y las proyecta fuera del hipocampo a través del subículo [2].

Para nuestro proyecto, las imágenes utilizadas se obtienen mediante la técnica de inmunofluorescencia. Esta técnica se basa en la unión de anticuerpos a una sustancia fluorescente que indica la presencia de una determinada molécula para el estudio de tejidos.

Para ello los ratones son anestesiados y perfundidos con paraformaldehído para fijar el tejido cerebral. A continuación cada cerebro es seccionado en rodajas utilizando un vibratomo. Estas rodajas se someten a marcaje de las estructuras somatodendríticas neuronales mediante inmunofluorescencia, utilizando anticuerpos específicos contra la proteína MAP2. Finalmente, se toman imágenes de microscopía confocal de hipocampo.

Para la captación de imágenes con este tipo de preparación existen dos métodos: la captación mediante un microscopio confocal laser o mediante un microscopio de epifluorescencia.

Si una muestra gruesa es observada al microscopio de epifluorescencia, la imagen que se enfoca se ve contaminada por la superposición de los elementos del tejido que están

fuera de foco, tanto por encima como por debajo del plano enfocado; la imagen enfocada se deteriora a causa de las estructuras superpuestas borrosas o no enfocadas. Además, tiene la dificultad de que el tejido debe cortarse finamente para ser examinado y mientras más delgado sea, más nítida será la imagen; pero con este método se pierde la información tridimensional durante el corte.

El microscopio confocal es un microscopio laser que permite realizar cortes ópticos finos a muestras de tejidos y realizar reconstrucciones en tres dimensiones a partir de las secciones. Al estar siempre dentro de foco, las imágenes obtenidas son de alta calidad debido a que no existe ruido de fondo.

2.3. *Patrones dendríticos.*

La información que se transmite a través de las dendritas tiene una alta implicación en el proceso de aprendizaje y memorización. Por eso, el estado del conjunto dendrítico es tan importante.

Se ha determinado que el Síndrome de Down en humanos conlleva defectos de arborización dendrítica. El equipo científico detectó, a raíz de los estudios preliminares, que en sus modelos de ratones sucedía un fenómeno similar.

A modo de ejemplo, si pudiésemos visualizar una única neurona en un tejido mediante tinción de Golgi y posteriormente multiplicarla para agruparla formando un conjunto dendrítico tendríamos los casos de la figura 2.5 y 2.6.

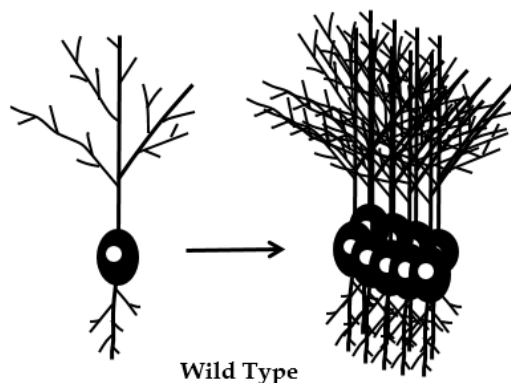


Figura 2.5. Patrón dendrítico para muestras Wild Type.

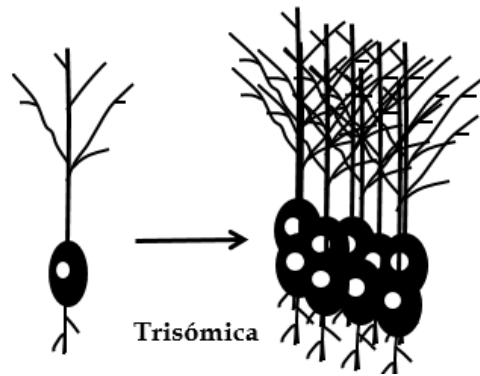


Figura 2.6. Patrón dendrítico para muestras trisómicas.

Generalmente, se nombra a los ratones de control, sin problemas neurológicos, como *wild type* y a los ratones con Síndrome de Down, *trisómico*.

Como se puede observar, la agrupación neuronal clasificada como *wild type* tiene como resultado una arborización dendrítica más densa. Esto implica un mayor conexionado entre neuronas y por tanto un mayor desarrollo. No es así en las muestras trisómicas, donde el entrelazado de interconexiones es más pobre. Este último patrón es el encontrado en los sujetos con Síndrome de Down o trisomía 21.

3. Estado del arte

3.1. Textura

La textura es una propiedad subjetiva del objeto que da información acerca de su estructura superficial. Es una característica dependiente de la percepción humana, ya que nuestra forma de medirlas es a través del tacto o la vista.

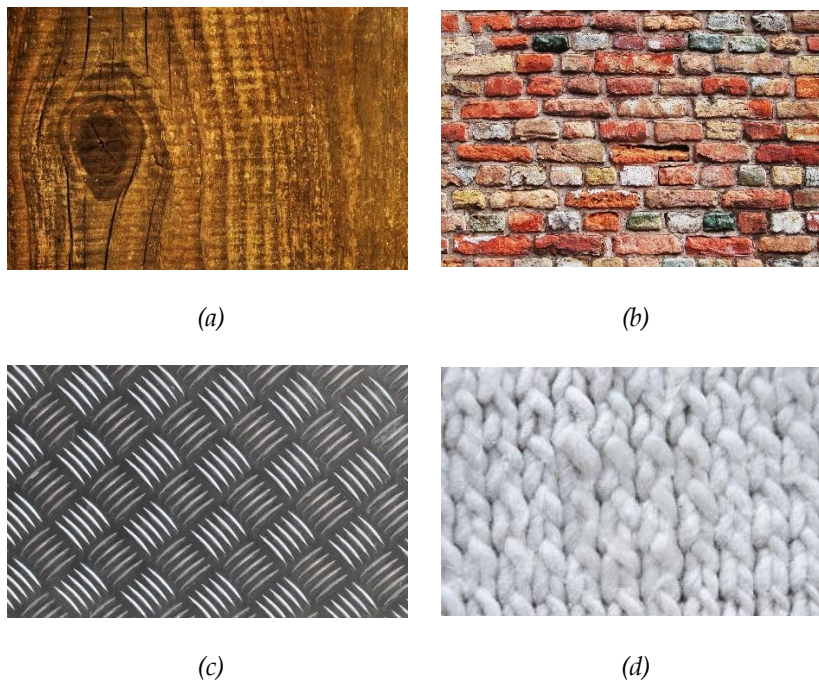


Figura 3.1. Ejemplos de texturas. (a) Madera. (b) Pared de ladrillos.
(c) Metal. (d) Tejido algodón.

En procesamiento de imágenes la textura es compleja de medir, tanto es así que existe una rama dedicada a su investigación denominada *pattern recognition*.

3.1.1. Análisis de texturas

En procesamiento de imagen cuantificamos la textura mediante descriptores de texturas. Los descriptores de textura proporcionan una medida matemática de las características visuales o táctiles de una superficie.

A continuación se detallaran los métodos que describen a una textura.

3.1.1.1. Métodos estadísticos

Analizan la distribución estadística de alguna propiedad para cada uno de los píxeles de la imagen. Dentro de los métodos estadísticos existen dos clasificaciones: métodos de primer y segundo orden.

Los métodos estadísticos de primer orden se basan en el histograma de la imagen. Aunque también es posible aplicarlos a los valores de una imagen. Un histograma es una representación gráfica que aporta información sobre la frecuencia de los valores de una imagen.

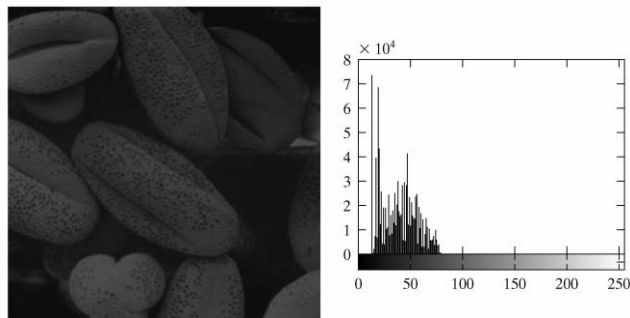


Figura 3.2. Ejemplo de histograma.

Sea

$$h(i) = \frac{\text{número de píxeles con intensidad } i}{\text{número total de píxeles}} \quad (3.1)$$

Entonces:

$$\text{Media: } \mu = \sum_{i=1}^n ih(i) \quad (3.2)$$

$$\text{Varianza: } \sigma^2 = \sum_{i=1}^n (i - \mu)^2 h(i) \quad (3.3)$$

Coficiente de asimetría:

$$\mu_3 = \frac{1}{\sigma^3} \sum_{i=1}^n (i - \mu)^3 h(i) \quad (3.4)$$

$$\text{Kurtosis: } \mu_4 = \frac{1}{\sigma^4} \sum_{i=1}^n (i - \mu)^4 h(i) \quad (3.5)$$

$$\text{Entropía: } H = - \sum_{i=1}^n h(i) \log h(i) \quad (3.6)$$

Los métodos estadísticos de segundo orden se basan en la matriz de co-ocurrencia. La matriz de co-ocurrencia considera la relación espacial entre los niveles de gris de una imagen contabilizando cuantos pares de píxeles con unos valores determinados existen.

La regla con la que se forma la matriz de co-ocurrencia puede variar dependiendo de la distancia y la orientación al pixel (i,j):

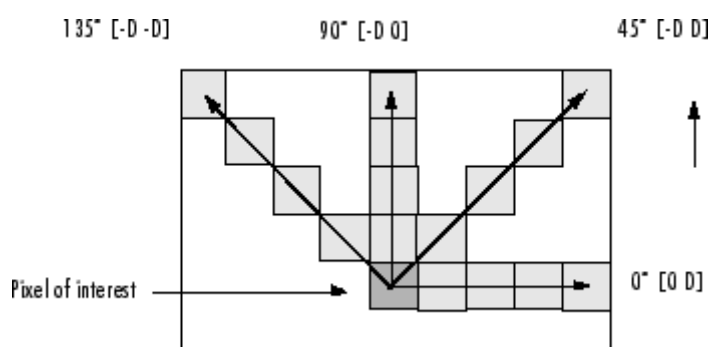


Figura 3.3. Distancia y orientación de los pares [5].

Así, la matriz de co-ocurrencia de I para [0,1] sería:

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Figura 3.4. Matriz de co-ocurrencia [0,1] de I [5].

Sea $p(i, j)$ la matriz de co-ocurrencia normaliza, Haralick [6] en 1973 definió una serie de características que se podían obtener a partir de la matriz de co-ocurrencia:

$$\text{Segundo momento angular: } f_1 = \sum_i \sum_j p(i, j)^2 \quad (3.7)$$

$$\text{Contraste: } f_2 = \sum_{n=0}^{Ng-1} n^2 \left\{ \sum_{\substack{i=1 \\ |i-j|=n}}^{Ng} \sum_{j=1}^{Ng} p(i, j) \right\} \quad (3.8)$$

$$\text{Correlación: } f_3 = \frac{\sum_i \sum_j (i, j) P(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (3.9)$$

$$\text{Varianza: } f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j) \quad (3.10)$$

$$\text{Momento Diferencia Inversa: } f_5 = \sum_i \sum_j \frac{p(i, j)}{1 + (i - j)^2} \quad (3.11)$$

$$\text{Suma promedio: } f_6 = \sum_{i=2}^{2Ng} i p_{x+y}(i) \quad (3.12)$$

$$\text{Suma varianza: } f_7 = \sum_{i=2}^{2Ng} (i - f_6)^2 p_{x+y}(i) \quad (3.13)$$

$$\text{Suma entropía: } f_8 = - \sum_{i=2}^{2Ng} \log\{p_{x+y}(i)\} p_{x+y}(i) \quad (3.14)$$

$$\text{Entropía: } f_9 = - \sum_i \sum_j \log(p(i,j))p(i,j) \quad (3.15)$$

$$\text{Diferencia de la varianza: } f_{10} = \text{variance of } p_{x-y} \quad (3.16)$$

$$\text{Diferencia de entropía: } f_{11} = - \sum_{i=0}^{Ng-i} \log\{p_{x-y}(i)\}p_{x+y}(i) \quad (3.17)$$

Correlación:

$$f_{12} = \frac{HXY - HXY1}{\max\{HX, HY\}} \quad (3.18)$$

$$f_{13} = (1 - \exp[-2(HXY2 - HXY)])^{\frac{1}{2}} \quad (3.19)$$

$$\text{donde: } HXY = - \sum_i \sum_j \log(p(i,j))p(i,j) \quad (3.20)$$

$$HXY1 = - \sum_i \sum_j \log(p_x(i)p_y(j))p(i,j) \quad (3.21)$$

$$HXY2 = - \sum_i \sum_j \log(p_x(i)p_y(j))p_x(i)p_y(j) \quad (3.22)$$

Coefficiente máximo de correlación:

$$f_{14} = (\text{segundo mayor autovalor de } Q)^{\frac{1}{2}} \quad (3.23)$$

$$\text{donde } Q(i,j) = \sum_k \frac{p(i,k)p(j,k)}{p_x(i)p_y(j)} \quad (3.24)$$

3.1.1.2. Métodos espectrales

Los métodos espectrales se basan en la transformada de Fourier 2D.

El espectro de la transformada ofrece características como el periodo espacial y la dirección de los patrones de la textura observando únicamente picos prominentes en el espectro:

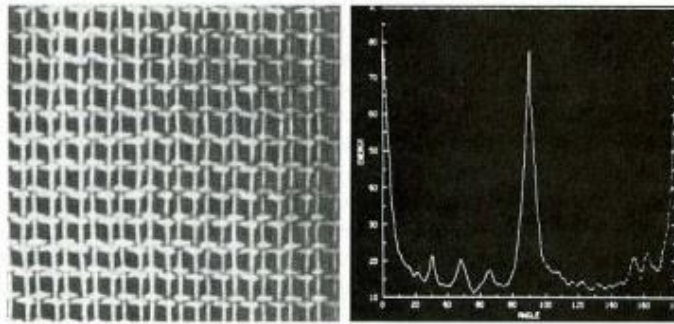


Figure 3.5. Espectro en frecuencia de tejido.

Merece la pena mencionar los filtros de Gabor [7]. Se trata de un filtro lineal cuya respuesta al impulso está definida por una onda sinusoidal multiplicada por una función gaussiana:

$$h(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} \cos(2\pi u_0 x) \quad (3.25)$$

Por la propiedad de multiplicación de la transformada de Fourier, la transformada de Fourier de la respuesta al impulso es la convolución de la transformada de Fourier de la función armónica y la transformada de Fourier de la función gaussiana:

$$H(u, v) = e^{-2\pi^2[\sigma_x^2(u-u_0)^2 + \sigma_y^2 v^2]} + e^{-2\pi^2[\sigma_x^2(u+u_0)^2 + \sigma_y^2 v^2]} \quad (3.26)$$

El interés reside en que las funciones de Gabor están localizadas en dominio espacial y frecuencial, característica muy apropiada para la representación de texturas.

3.1.1.3. Métodos estructurales

Los métodos estructurales parten de la base de que la textura está formada por una estructura que se repite. A esta estructura se le denomina primitiva. Son efectivos cuando los patrones son grandes y sencillos.

Las primitivas de texturas pueden ser:

- De repetición aleatoria: se forman emplazando elementos similares de manera aleatoria.
- De estructura regular: sobre un patrón regular se disponen varios elementos distintos.
- De repetición regular: sobre un patrón similar se disponen elementos similares.

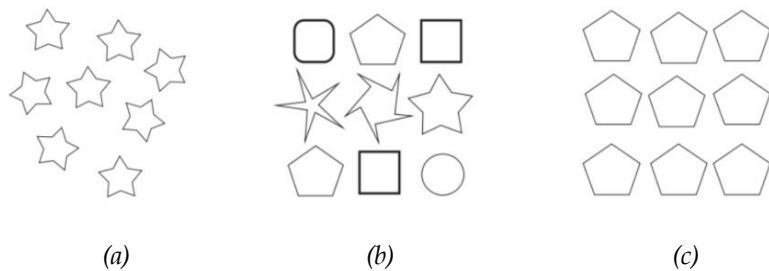


Figura 3.6. Tipos de primitivas: (a) de repetición aleatoria (b) de estructura regular (c) de repetición regular

Destacar también, la medida de la dimensión fractal [8]. Un fractal es un objeto geométrico que puede ser dividido en partes, donde cada una de ellas es una copia reducida del total.

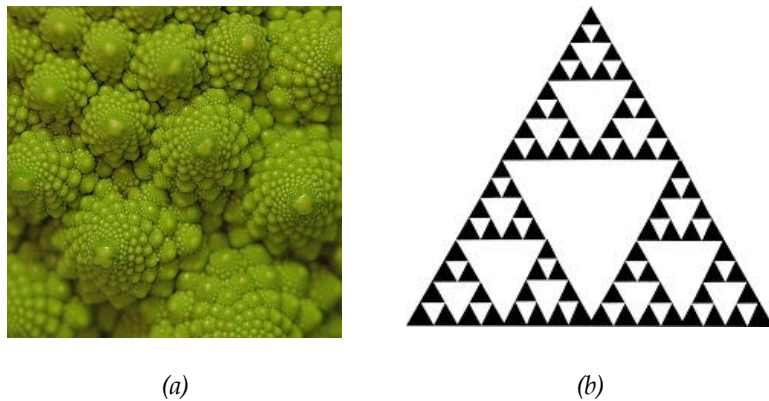


Figura 3.7. Ejemplo de geometría fractales.

La dimensión fractal es una medida que generaliza el concepto de dimensión ordinaria para objetos geométricos que no admiten espacio tangente.

La dimensión de Hausdorff [9] es una caracterización de la dimensión fractal que se basa en considerar círculos abiertos (para un fractal contenido en un plano euclídeo) o esferas (para un fractal contenido en el espacio euclídeo tridimensional). Se define el contenido de Hausdorff como:

$$H_\delta^s(F) = \inf\left\{\sum_{i=1}^{\infty} |U_i|^s, |U_i| = \text{diam}(U_i) < \delta\right\} \quad (3.27)$$

Y por tanto, la medida exterior de Hausdorff del conjunto F:

$$H^s(F) = \lim_{\delta \rightarrow 0} H_\delta^s(F) \quad (3.28)$$

3.1.1.4. Local Binary Patterns

Local Binary Patterns (LBP) es un descriptor de textura basado en histograma propuesto por T. Ojala en 1994 [10]. En su algoritmo, se recorre la imagen pixel por pixel y se evalúa si la intensidad de cada pixel de la vecindad es mayor o menor que el pixel central. En caso de ser mayor, se le asigna un 1, sino un 0. Finalmente, a cada cadena binaria se le asigna una codificación.

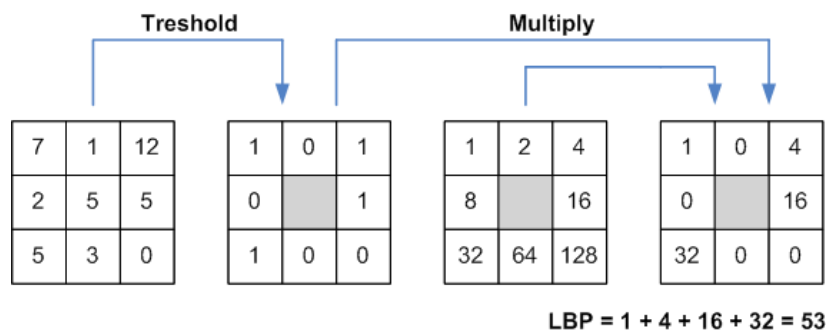


Figura 3.8. Ejemplo de algoritmo LBP para una vecindad 3x3 [11].

La vecindad puede tener una escala mayor, como muestra la figura 3.9.

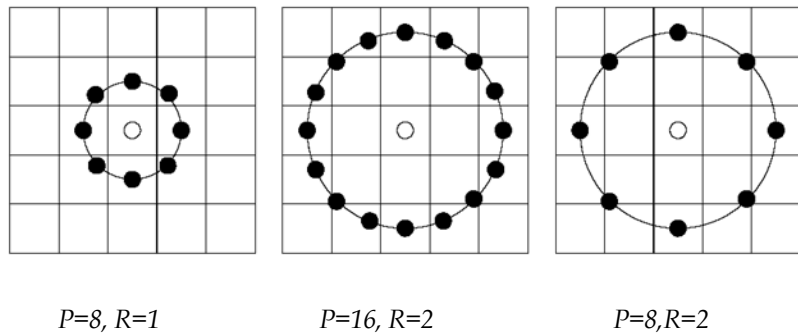
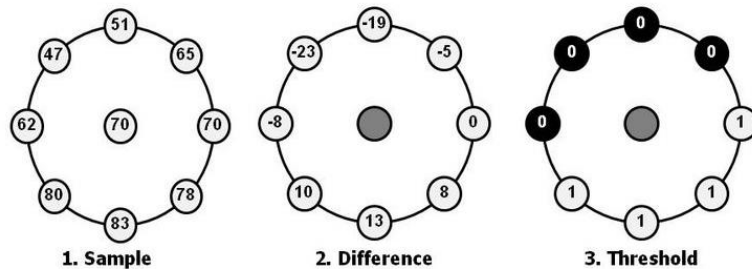


Figura 3.9 Ejemplos de vecindarios multiescala [12].

Así, el valor del código LBP de un pixel (x_c, y_c) es:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (3.29)$$

$$\text{donde } s(x) = \begin{cases} 1, & \text{si } x \geq 0; \\ 0, & \text{en otro caso} \end{cases}$$



$$1*1 + 1*2 + 1*4 + 1*8 + 0*16 + 0*32 + 0*64 + 0*128 = 15$$

Figura 3.10. Ejemplo de código LBP en multiescala [13].

Existen multitud de implementaciones a partir del código LBP básico: invariantes a la rotación, invariantes al escalado, teniendo en cuenta la información de fase (LPQ [14]) o a partir de la transformada discreta de Fourier (LBP-HF [15]).

3.2. Descriptores de texturas en imágenes biomédicas

Entre los descriptores estadísticos, Haralick [6] es uno de los descriptores de texturas más conocidos: se pueden encontrar estudios para modelar la textura de los núcleos de células HI-60 [16], clasificación de meniangiomas [17] y análisis morfológicos en pacientes de Alzheimer [18]. También destacar sus usos en segmentación 2D [19] y 3D [20, 21].

Recientemente, se han implementado nuevos algoritmos de mejora de GLCM [22], caracterizando la textura en base a la matriz de tamaño zonal de niveles de gris (GLSZM) para la clasificación de células HEp-2.

Los descriptores basados en histograma, como el LBP, tienen una gran aceptación en la actualidad. Varios estudios se basan en este algoritmo para la segmentación del cerebro en imágenes de resonancia [23]; segmentación 3D del anillo fibroso y núcleo fibroso del disco intervertebral [24] o clasificación de patologías como la demencia [25] en imágenes de resonancia magnética.

Los filtros de Gabor, además de las aplicaciones en reconocimiento de rostros [26], también tienen aplicaciones en el análisis de características de texturas [26, 28].

Todos los métodos aplican reconocimiento de texturas a neuronas individuales o como ayuda para la segmentación de regiones. Sin embargo, en la actualidad existen pocas investigaciones [29] que analicen la textura en conjuntos de neuronas con el fin de caracterizar enfermedades. Hasta nuestro conocimiento, no existe ningún estudio dedicado a la detección de patrones o defectos morfológicos en neuronas.

3.3. Clasificadores

El objetivo de un clasificador es obtener un sistema que sea capaz de predecir un tipo de clase a partir de un conjunto de muestras conocidas. Existen dos tipos de clasificadores: los supervisados y los no supervisados. En el entrenamiento de los clasificadores supervisados es necesario indicar que clase es cada vector de características, mientras que los no supervisados lo hacen por sí mismos. Nosotros nos centraremos en los supervisados.

En general, para la creación del clasificador es necesario tener un conjunto de imágenes para entrenamiento con sus respectivas etiquetas de clase, y otro conjunto para clasificación. Ninguna imagen del conjunto de entrenamiento puede estar en presente en el conjunto de clasificación y viceversa. Además, se recomienda que todas las imágenes estén normalizadas.

Así, una proporción de imágenes se utiliza para enseñar al clasificador, y el restante, para comprobar la fiabilidad del mismo. Si el clasificador tiene un porcentaje de acierto aceptable, en un futuro, será capaz de clasificar imágenes de las cuales se desconoce su clase con una fiabilidad significativa.

A continuación se explicaran los más importantes.

3.3.1. *k*-Nearest Neighbor

k-NN [30] es un método de clasificación que estima la probabilidad a posteriori de que un elemento x pertenezca a la clase C a partir de la información proporcionada por el conjunto de entrenamiento.

Sea el vector de características, con p atributos, perteneciente a la muestra i del conjunto N :

$$x_i = (x_{1i}, x_{2i}, \dots, x_{pi}) \quad i = 1, \dots, N. \quad (3.30)$$

Entonces se toma cada vector de características como las

coordenadas de un espacio de dimensión N. Si deseamos clasificar una nueva muestra entre los k vecinos más cercanos, se posiciona y se cuantifican el número de elementos de las distintas clases que le rodean.

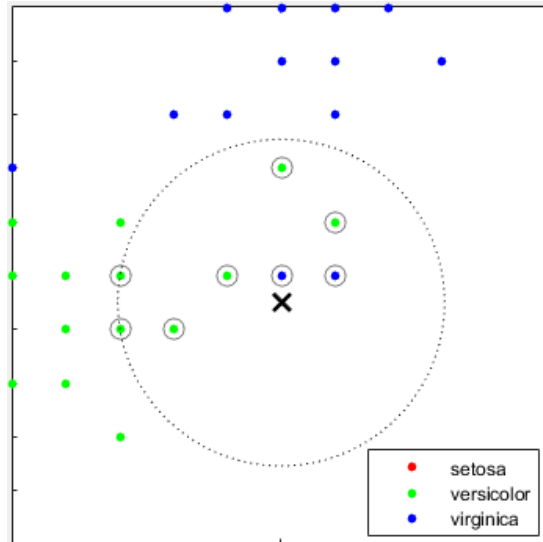


Figura 3.11. Ejemplo espacio de muestras k-NN [31].

En ejemplo que se muestra en la figura 3.11, con $k=8$, se determinaría que la nueva muestra es *versicolor* ya que existen seis muestras *versicolor* y dos *virginica*. Si deseásemos clasificar con $k=3$ la predicción sería *virginica* ya que existe dos muestras *virginicas* y una muestra *versicolor*.

La selección del tipo de distancia del vecino más cercano puede variar la toma de decisión [31]:

- Distancia euclídea:

$$d_{ij} = \sqrt{\sum_{r=1}^P (x_{ri} - y_{rj})^2} \quad (3.31)$$

donde x_{ri} son las coordenadas definidas en (3.29) e y_{rj} la coordenadas del vecino en (r, j)

- Distancia ponderada: se da mayor peso a los vecinos más cercanos. Así, no hay riesgo de permitir a un vecino muy lejano (pero que entre en el conjunto perteneciente a k) contribuir a la clasificación de x.

$$d_{ij} = \sqrt{\sum_{r=1}^P w_i (x_{ri} - y_{rj})} \quad (3.32)$$

donde w_i representa el vector de pesos inverso.

- Distancia Minkowski:

$$d_{ij} = \sqrt[p]{\sum_{r=1}^N |x_{ri} - y_{rj}|^p} \quad (3.33)$$

Casos especiales:

si $p = 1$, entonces d_{ij} es la distancia City Block

si $p = 2$, entonces d_{ij} es la distancia euclídea

si $p = \infty$, entonces d_{ij} es la distancia de Chebychev

- Distancia del coseno:

$$d_{ij} = \left(1 - \frac{x_i y_j'}{\sqrt{(x_i x_i')(y_j y_j')}} \right) \quad (3.34)$$

Donde x es el vector de coordenadas (3.29) e y el vector de coordenadas del vecino que se toma la distancia.

La influencia de la distancia reside en la elección de las muestras que se usan para tomar una decisión. Es posible que entre dos tipos de distancias, se seleccionen muestras que tengan clases distintas, y por tanto, cambiará la decisión que toma el clasificador.

3.3.2. Redes neuronales

Las redes neuronales artificiales (RNA) son una familia de modelos inspirados en las redes neuronales biológicas. Su objetivo es simular las tareas cognitivas del ser humano mediante un algoritmo.

Están formadas por un conjunto de unidades, llamadas neuronas, que están interconectadas entre sí:

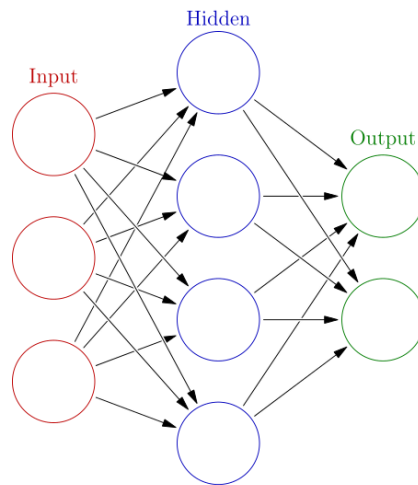


Figura 3.12. Red neuronal [32].

En general, en cualquier red neuronal con i neuronas, existen tres elementos [33]:

- Función o regla de propagación: permite obtener a partir de las entradas y los pesos, el valor potencial postsináptico $h_i(t)$ de la neurona.

$$h_i(t) = \sum_j w_{ij}x_j \quad (3.35)$$

donde w_{ij} representa el peso de la unión (i,j) , x_j el valor de la neurona presináptica j . Dada una entrada positiva, si el peso es positivo, el sistema excitará a la neurona postsináptica, y si es negativo tenderá a inhibirla.

- Función de activación: proporciona el estado de activación actual $a_i(t)$ a partir del potencial postsináptico $h_i(t)$ y del estado de activación anterior $a_i(t - 1)$:

$$a_i(t) = f_i(a_i(t - 1), h_i(t)) \quad (3.36)$$

En ocasiones, se determina que el estado anterior no afecta al estado de activación, dependiendo así, solamente del potencial postsináptico.

Entre las funciones de activación f_i habituales puede seleccionarse la identidad, el escalón, funciones sigmoideas, gaussianas o sinusoidales.

- Función de salida: proporciona la salida de la neurona en función de su estado de activación actual. Normalmente, la función de salida es la identidad, de modo que el estado de activación de la neurona se considera la propia salida.

Durante la fase de entrenamiento, la red neuronal aprende mediante la actualización de los pesos que caracterizan a las conexiones. Los pesos se adaptan de acuerdo a la información extraída de las características de las muestras. Normalmente, se eligen los pesos optimizando cierta función de energía.

Posteriormente, una vez fijados los pesos de la red neuronal, se prueba la capacidad que tiene la red de dar una respuesta correcta ante patrones que no han sido empleados en su entrenamiento.

3.3.3. Support Vector Machine (SVM)

Support Vector Machine (SVM) es un clasificador supervisado desarrollado por V. Vapnik en 1995 [34]. El objetivo de SVM es dividir un espacio de muestras, linealmente separables según su clase, mediante un hiperplano. Durante la clasificación, las muestras que están a un lado del hiperplano se etiquetaran como una clase, y las que están al otro como otra.

Sea el vector de características, con P atributos, perteneciente a la muestra i del conjunto N:

$$x_i = (x_{1i}, x_{2i}, \dots, x_{pi}) \quad i = 1, \dots, N. \quad (3.37)$$

Se toma cada vector de características como las coordenadas de un espacio de dimensión N, denominándolos vectores de soporte. Así, Support Vector Machine clasifica la información de entrada encontrando el mejor hiperplano que separa los puntos de una clase de otra.

El mejor hiperplano es aquel que tiene la mayor distancia, perpendicular al hiperplano, entre dos vectores soporte de clases distintas sin que ningún otro vector soporte pertenezca a ella.

Debido a la dificultad de separabilidad de los datos, en ocasiones se define un margen *suave*, que permite que cierto número de muestras pertenezcan al margen.

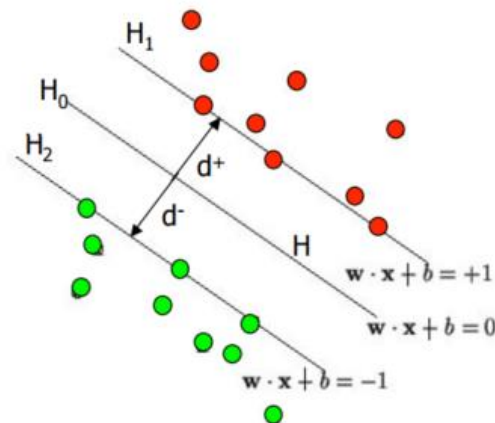


Figura 3.13. Hiperplano y margen caso 2D [35].

Definiendo el hiperplano H tal que [35]:

$$w \cdot x_i + b \geq +1 \text{ cuando } y_i = +1 \quad (3.38)$$

$$w \cdot x_i + b \geq -1 \text{ cuando } y_i = -1 \quad (3.39)$$

Donde w denota el vector normal al hiperplano, b el término independiente y x y y coordenadas en el espacio. Entonces los planos H_1 y H_2 se definen como:

$$w \cdot x_i + b = +1 \quad (3.40)$$

$$w \cdot x_i + b = -1 \quad (3.41)$$

Si el conjunto de entrenamiento es linealmente separable podemos seleccionar dos hiperplanos en el borde de los conjuntos de vectores de soporte maximizando su distancia.

La distancia entre H_0 y H_1 es por tanto:

$$\frac{|w \cdot x + b|}{\|w\|} = \frac{1}{\|w\|} \quad (3.42)$$

Para maximizar el margen, debemos minimizar $\|w\|$. Desarrollando por el método Lagrange el problema a resolver es:

$$\min \frac{1}{2} \|w\|^2 - \sum_i a_i y_i (x_i \cdot w + b) + \sum_i a_i \quad (3.43)$$

donde los términos a_i constituyen una representación del vector de pesos en el conjunto de entrenamiento

Analizando el problema primal y dual obtenemos que la ecuación del plano es:

$$w = \sum_i a_i y_i x_i \quad (3.44)$$

donde los términos a_i constituyen una representación del vector de pesos en el conjunto de entrenamiento.

En la mayoría de los casos, el problema de separación no es lineal. Para ello, se definen unas funciones o kernels que transforman el espacio de entrada en un espacio que sea separable linealmente:

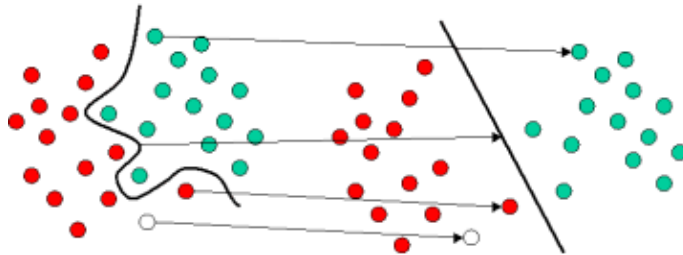


Figura 3.14. Ejemplo de transformación polinomial [36].

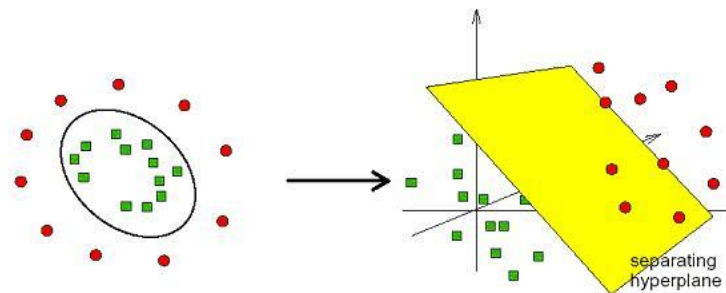


Figura 3.15. Ejemplo de transformación radial [37].

Existen multitud de kernels además del lineal: cuadrático, cúbico, gaussiano o con bases radiales. Dependiendo de la forma que muestre nuestro espacio de muestras, habrá que seleccionar el kernel que mejor se adapte para obtener una separabilidad clara.

3.3.4. Árboles de decisión. Random Forest.

En general, el objetivo de los árboles de decisión es crear un modelo que predice el valor de una variable de salida en función de diversas variables de entrada. Cada hoja es un valor de la variable de salida y cada rama representa la conjunción de características o condiciones que conducen a ese valor de la variable de salida. Si la variable de salida toma valores continuos, se les denomina árboles de regresión.

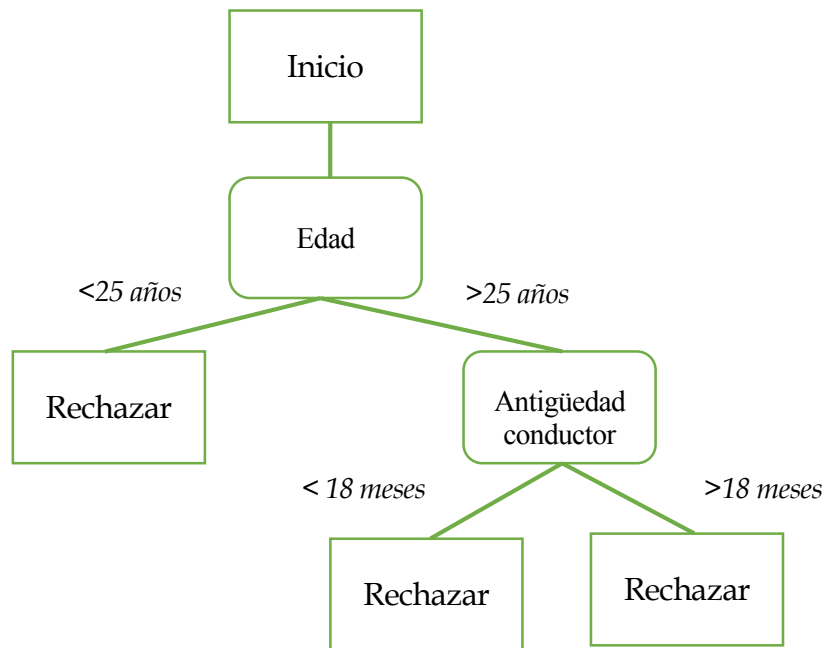


Figura 3.16. Árbol decisión: condición contratación seguro de automóvil.

Particularmente, en la clasificación de imágenes, cada hoja del árbol representa un tipo de clase, y cada rama, un conjunto de características.

Algunas técnicas construyen clasificadores con más de un árbol de decisión. Este es el caso de los métodos:

- Bagging (Breiman, 1996) [38]: construye varios árboles de decisión de gran tamaño muestreando repetidamente la información de entrenamiento y los promedia para obtener el clasificador. Finalmente, clasifica según la clase que más se repita.

- Boosting (Freund & Shapire, 1996) [39]: crea varios árboles de decisión utilizando todos los datos de entrenamiento. Para los casos que fueron erróneamente clasificados, se les aplica un peso para que los siguientes árboles mejoren su respuesta. Existen versiones como AdaBoost [39] (Adaptative Boosting) y LogiBoost [40] (Logistic Adaptation): ambas intentan asignar adaptativamente los pesos de los árboles de decisión para mejorar la respuesta del clasificador.
- Random Forests (Breiman, 1999) [41]: ofrece una mejora sobre Bagged Tree ya que permite eliminar la correlación entre los árboles.

3.3.4.1. *Random Forest*

Sea N el número de casos de prueba, m el número de variables de entrada a ser usado para determinar la decisión de un nodo dado y M el número de variables en el clasificador; m debe ser mucho menor que M . Si se da la condición, para la construcción de cada árbol se usa el siguiente algoritmo:

- Elegir un conjunto de entrenamiento para este árbol y usar el resto de los casos de prueba para estimar el error.
- Para cada nodo del árbol, elegir aleatoriamente m variables en las cuales basar la decisión. Calcular la mejor partición a partir de las m variables del conjunto de entrenamiento.

Para la predicción, se desciende a través del árbol y se le asigna la etiqueta del nodo terminal donde finaliza. Este proceso es iterado por todos los árboles y la etiqueta que más se repita es la que se usa en la predicción.

4. Algoritmo implementado

Llegados a este punto se procederá a detallar todo el proceso para la formación del clasificador. La figura 4.1 ofrece una perspectiva general de todo el proceso:

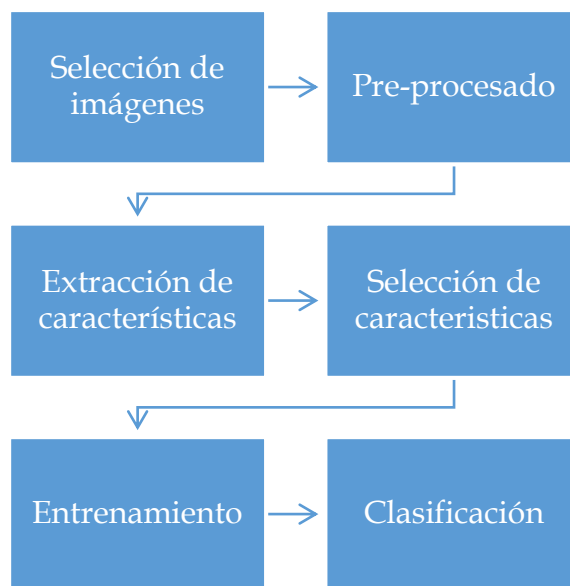


Figura 4.1. Esquema de formación de clasificador

El objetivo de nuestro clasificador es obtener un sistema que sea capaz de predecir un tipo de clase, en nuestro caso wild type o trisómico, a partir de un conjunto de imágenes conocidas. Al final de este capítulo obtendremos una clasificación automática de imágenes de hipocampos murino.

Recalcar que el conjunto de clasificación no puede contener ninguna imagen de entrenamiento, o viceversa. Además, se recomienda que todas las imágenes estén unificadas en cuanto a tamaño, iluminación, etc.

A cada imagen se le calculará un vector de características y se le asignará su correspondiente etiqueta. En el apartado 4.5 entrenaremos al clasificador, y posteriormente, en el apartado 4.6 clasificaremos el resto de muestras para conocer la fiabilidad del mismo.

En siguientes capítulos, se desarrollaran los resultados y pruebas obtenidas.

4.1. Selección de imágenes

El grupo científico se encargó de la captación de las imágenes en el Centro de Investigación, Tecnología e Innovación de la Universidad de Sevilla (CITIUS).

Las imágenes aportadas estaban en formato .czi, un formato desarrollado por la compañía Carl Zeiss para sus componentes ópticos y de microscopía. Carl Zeiss es una compañía especializada en el campo de la óptica y el desarrollo de la tecnología médica.

Cada imagen constaba de doce planos procedentes de un mismo hipocampo. Se decidió transformarlas a un formato que facilitase su tratamiento independientemente del sistema operativo.

Se continuó con la nomenclatura usada por el equipo científico : trX_imY_zZ o wtX_imY_zZ dependiendo del tipo de muestra, donde X e Y representa un índice dentro del tipo de las muestras y Z el plano que se ha tomado.

La figuras 4.2-4.13 exponen el hipocampo de una muestra trisómica con sus respectivos doce planos obtenidos mediante microscopía confocal.

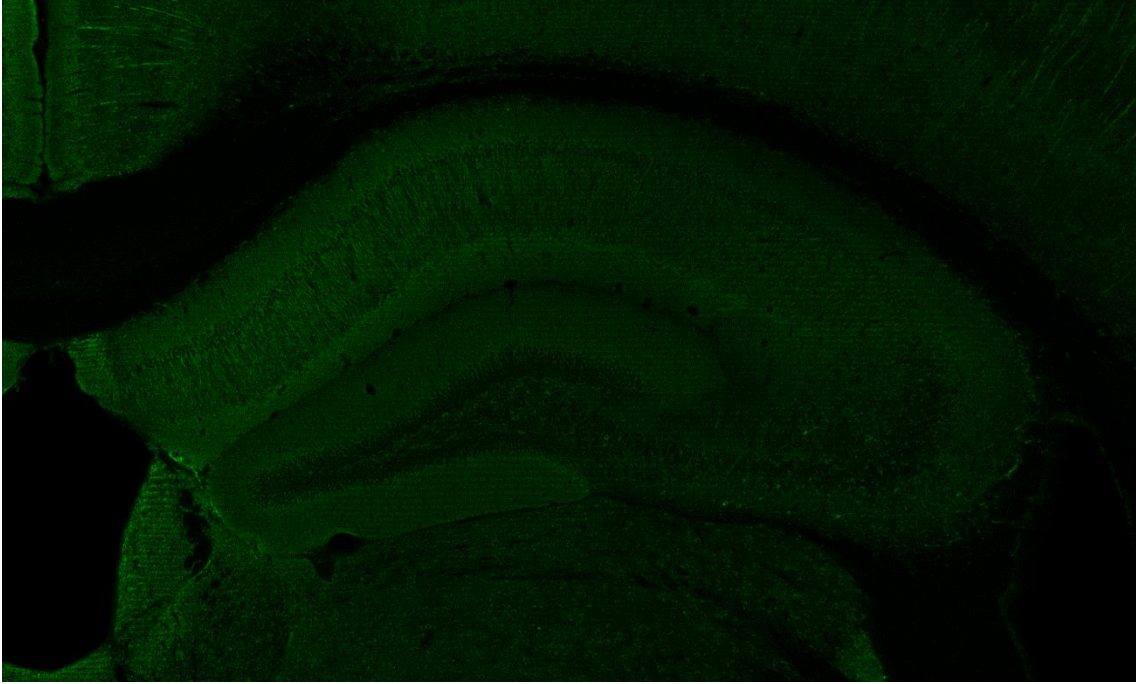


Figura 4.2. Muestra hipocampo para el plano Z=1.

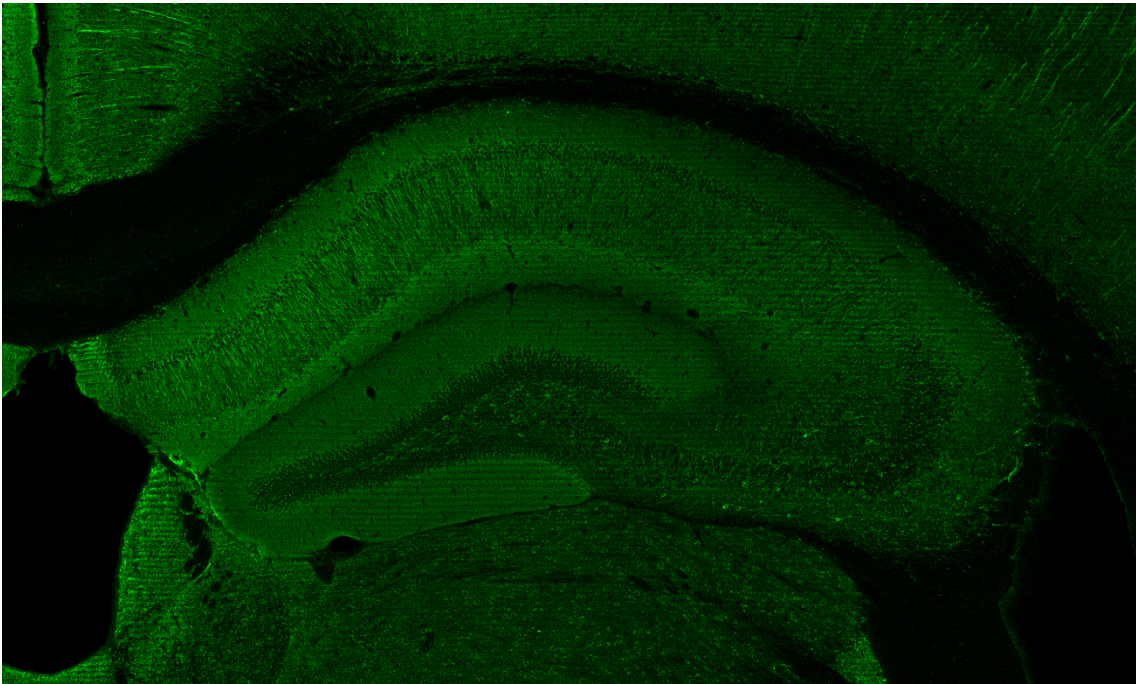


Figura 4.3. Muestra hipocampo para el plano Z=2.

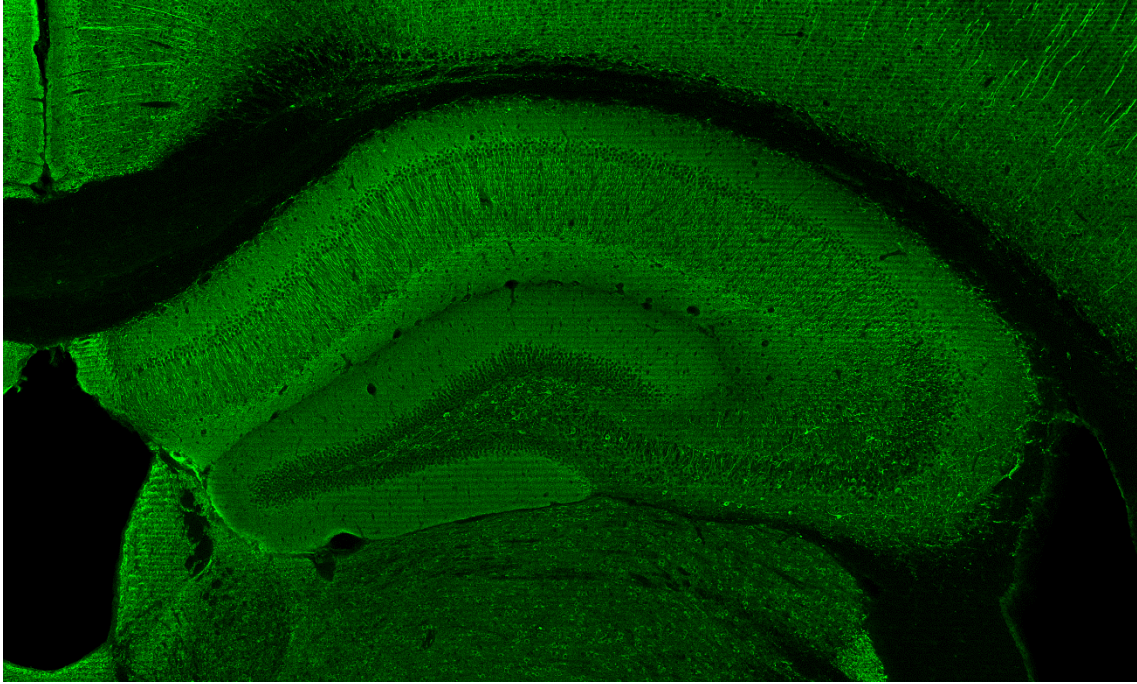


Figura 4.4. Muestra hipocampo para el plano Z=3.

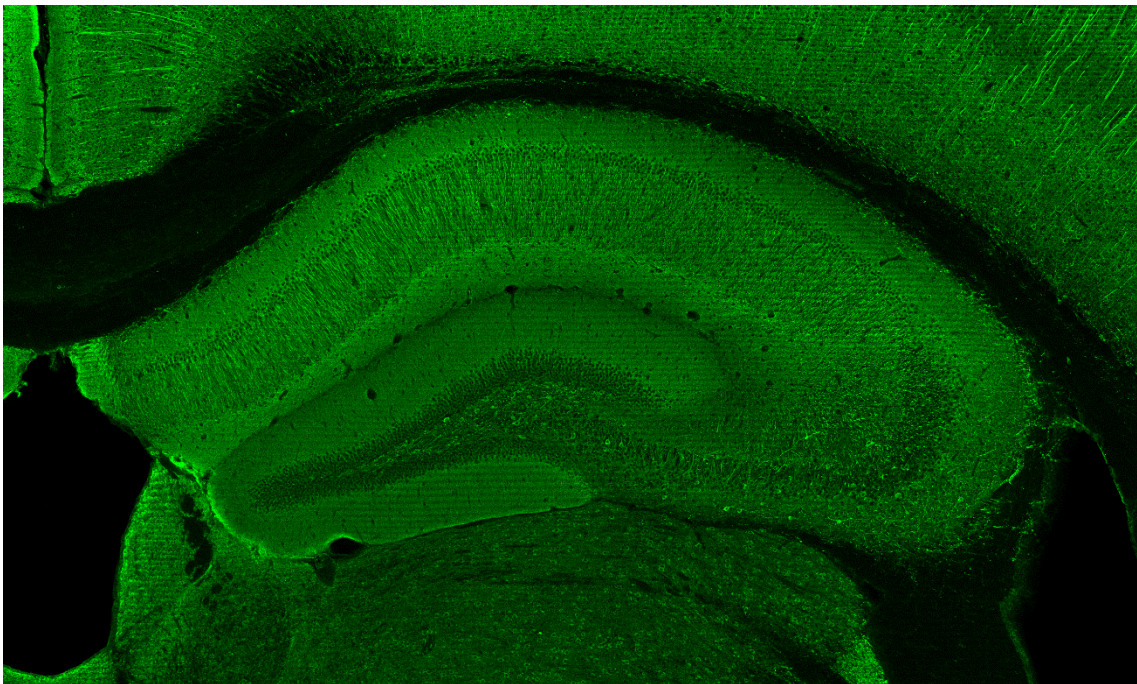


Figura 4.5. Muestra hipocampo para el plano Z=4.

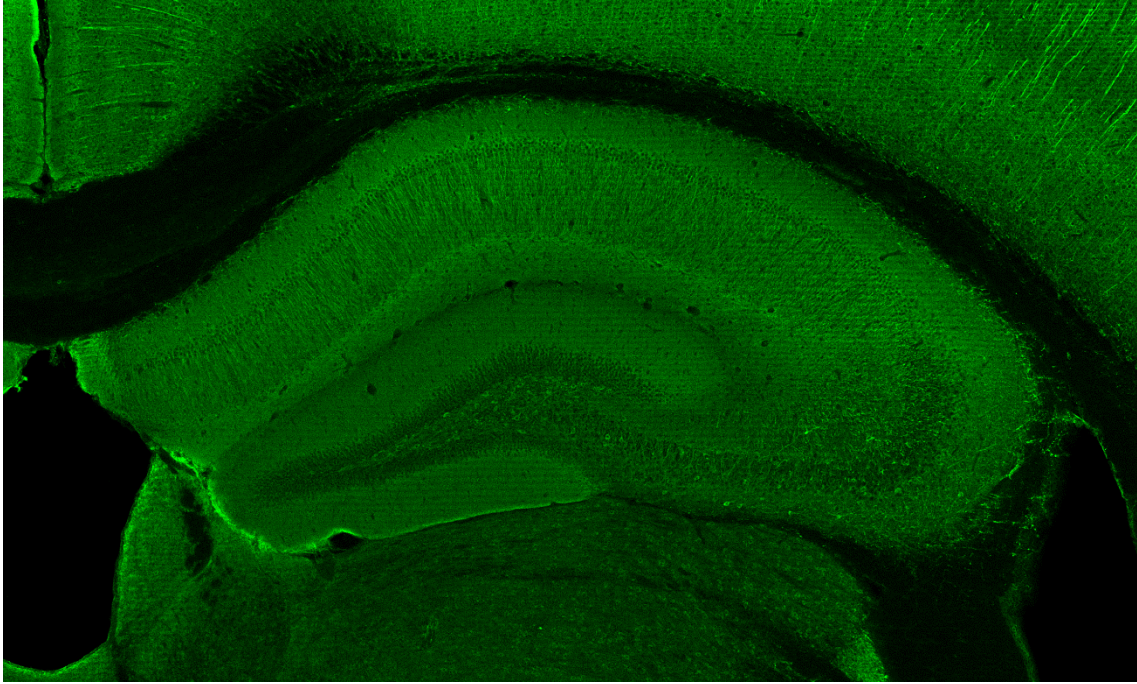


Figura 4.6. Muestra hipocampo para el plano Z=5.

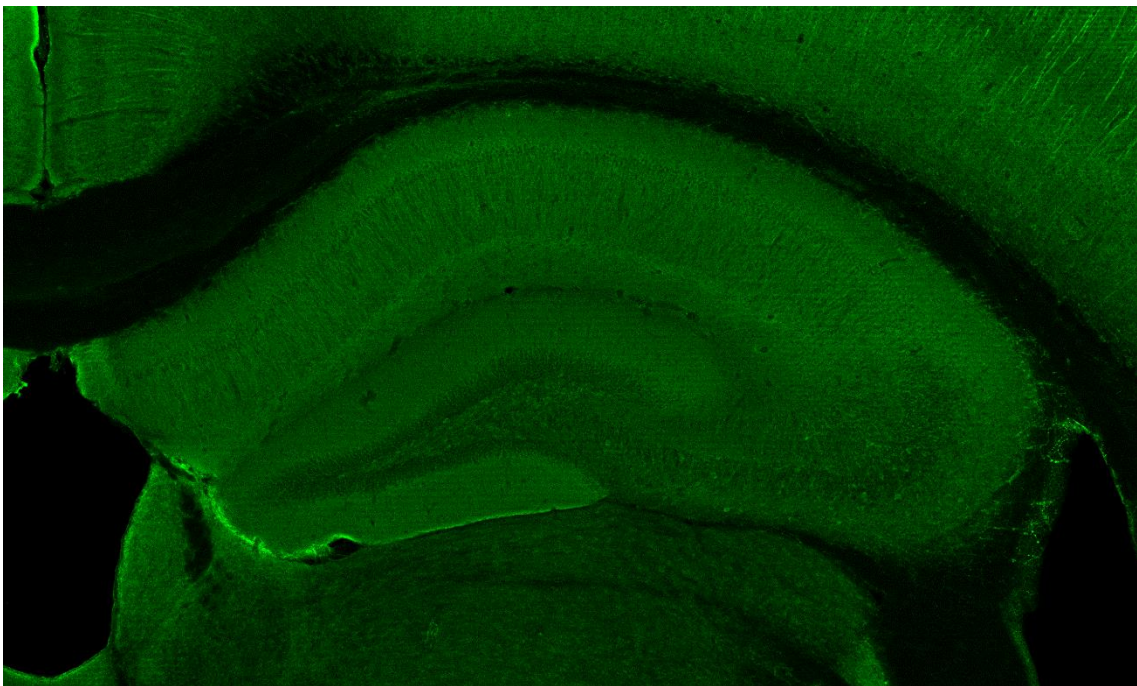


Figura 4.7. Muestra hipocampo para el plano Z=6.

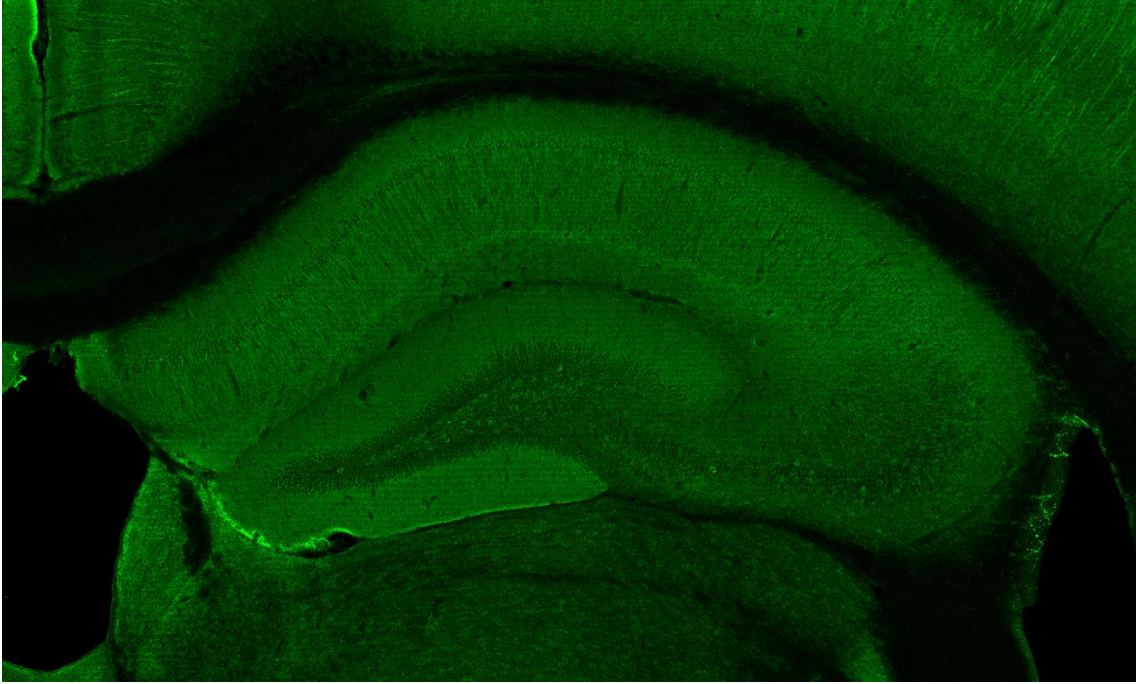


Figura 4.8. Muestra hipocampo para el plano Z=7.

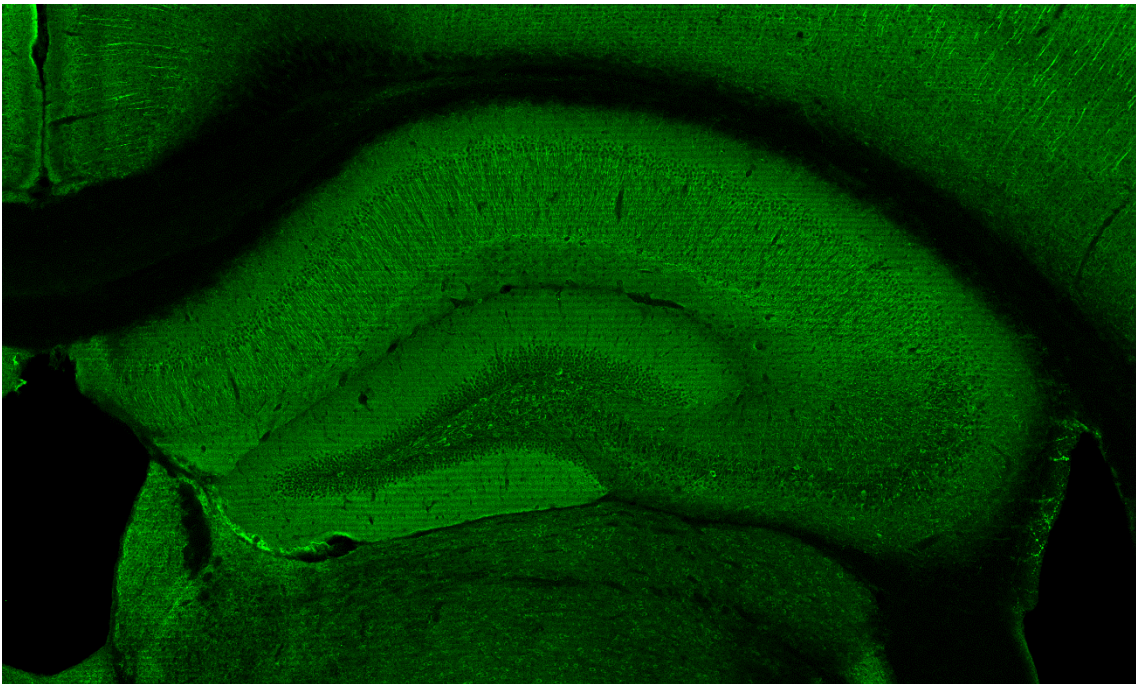


Figura 4.9. Muestra hipocampo para el plano Z=8.

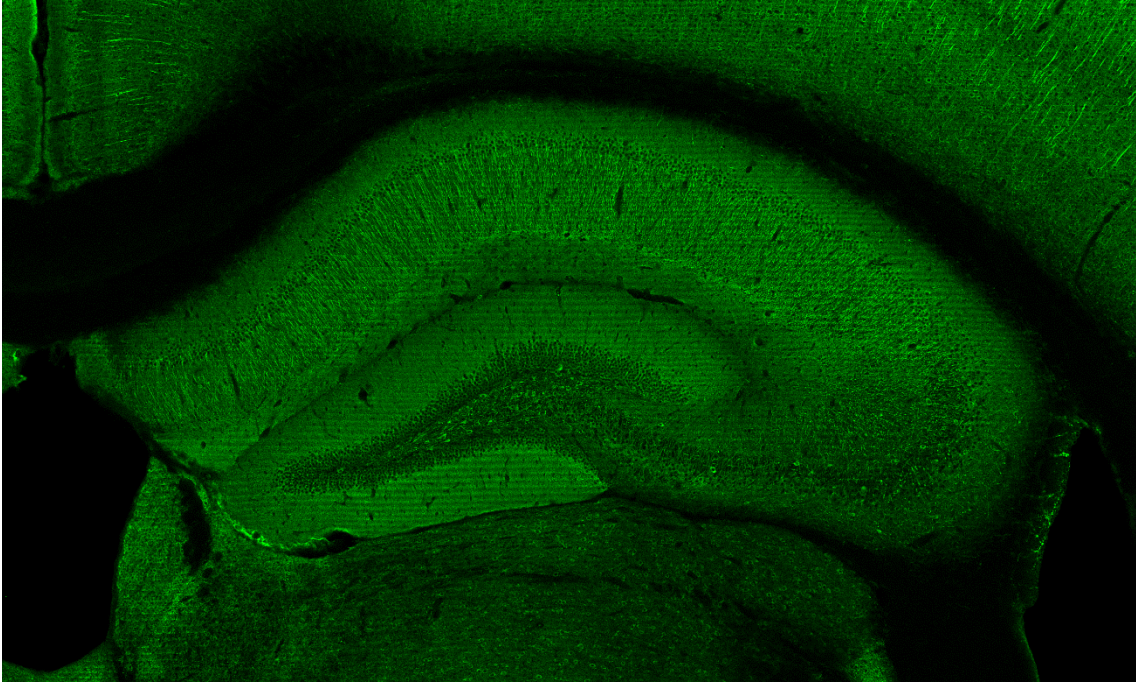


Figura 4.10. Muestra hipocampo para el plano Z=9.

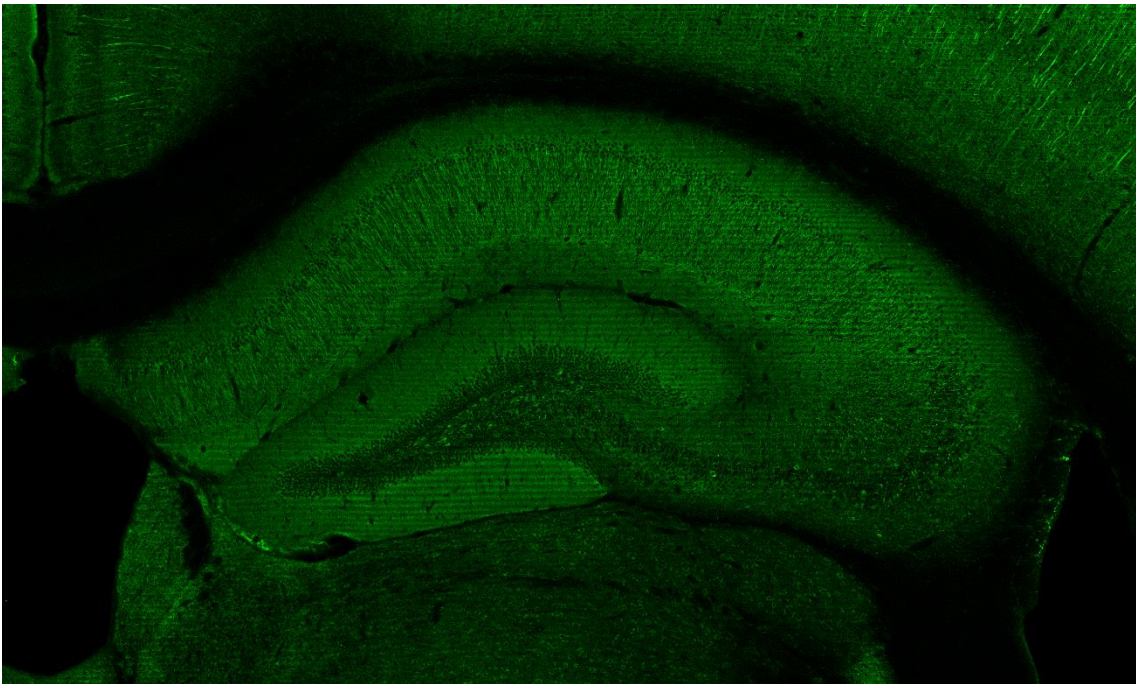


Figura 4.11. Muestra hipocampo para el plano Z=10.

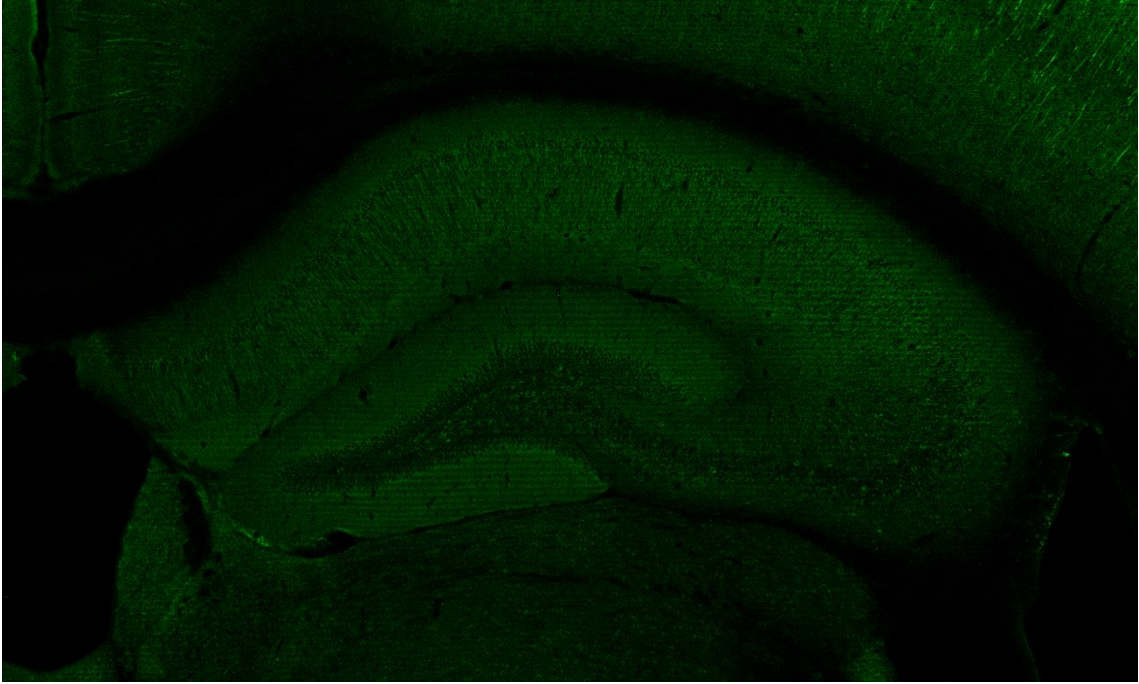


Figura 4.12. Muestra hipocampo para el plano Z=11.

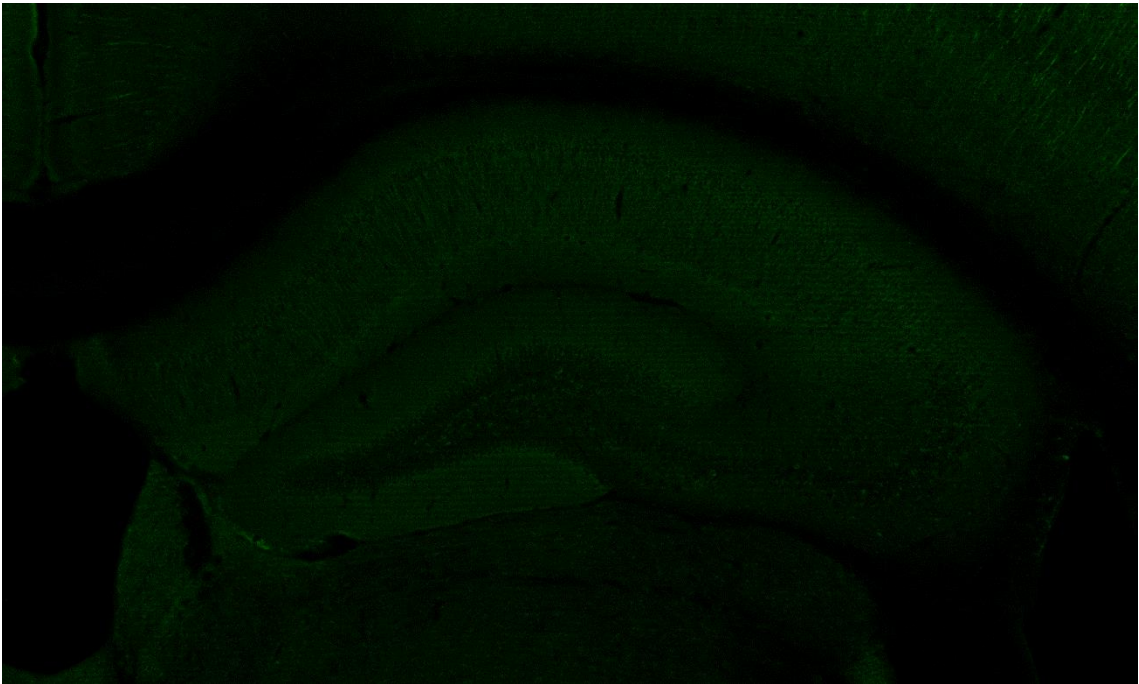


Figura 4.13. Muestra hipocampo para el plano Z=12.

Se proporcionaron quince imágenes de hipocampo. Seis de ellas tenían sección coronal, con lo cual, mostraban dos hipocampos y obteníamos el doble de muestras.



Figura 4.14. Ejemplo de imagen coronal de hipocampo.

Dentro de una muestra de hipocampo se seleccionó el plano que tenía el mejor contraste y nitidez en la región de interés, el resto se descartaron.

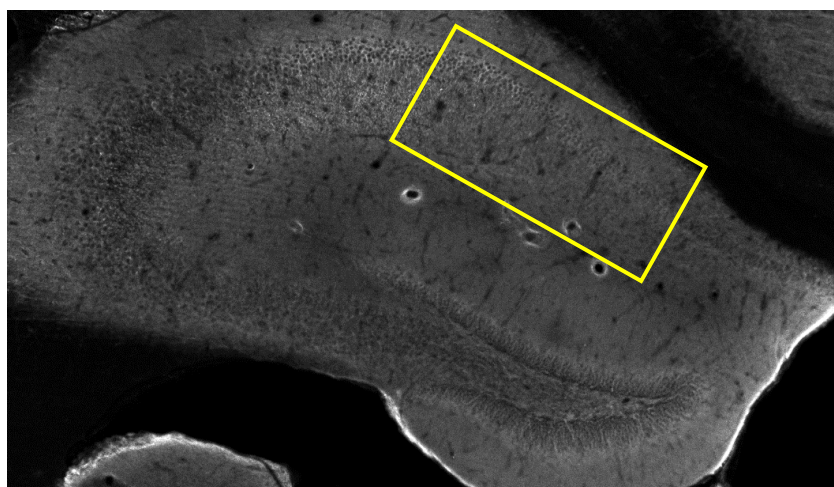


Figura 4.15. Imagen descartada por falta de nitidez.

Por ejemplo, en el conjunto anterior de figuras 4.2 - 4.13 el plano $Z=4$ (Figura 4.5) es el que destaca por su mayor calidad. En algunas imágenes ningún plano cumplía los requisitos y fueron descartadas todas las muestras.

Finalmente, la selección final estaba formada por nueve imágenes, conteniendo un total de diez muestras de hipocampos: cinco de ellas pertenecientes a la clase trisómica y otras cinco a la clase wild type.

4.2. Pre-procesado

El conjunto de imágenes de muestra seleccionado seguía mostrando algunas carencias: diferencias de iluminación, formas no deseadas de vasos sanguíneos y poco contraste, como en las figuras 4.16 y 4.17 respectivamente.

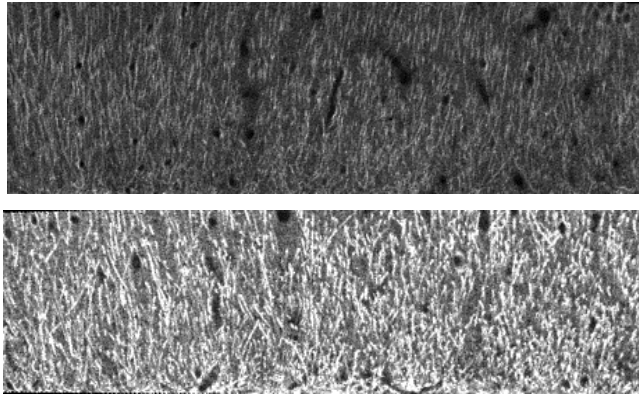
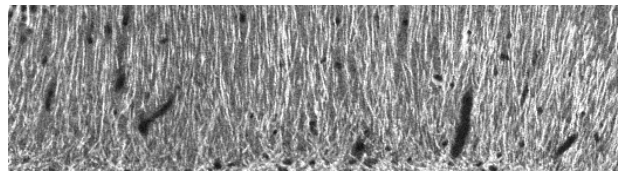
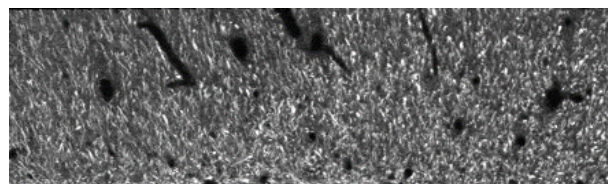


Figura 4.16. Ejemplo de diferencia de luminosidad entre muestras.



(a)



(b)

Figura 4.17. Ejemplo de muestras con cruce de vasos sanguíneos.

En la figura 4.18 se muestra una visión global del pre-procesado que explicaremos a continuación.

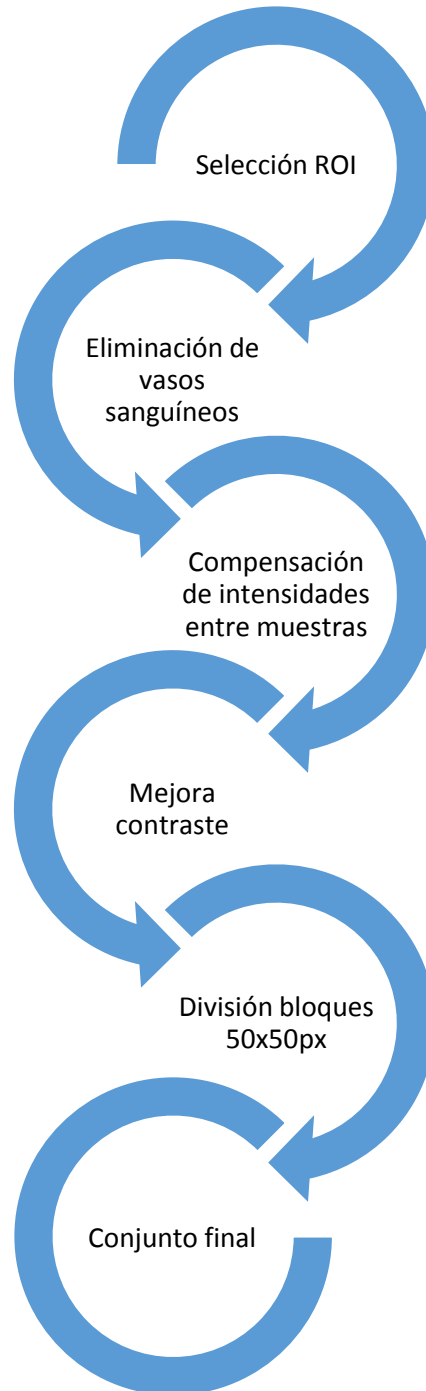


Figura 4.18. Esquema pre-procesado

Inicialmente, las imágenes estaban formadas por un único plano de color en escala de gris y había que seleccionar la región de interés.

Dado que las imágenes de hipocampo fueron tomadas previamente con otros fines, la orientación o posición del campo CA1 era muy variable. Así que, fue necesario su transformación para posicionarla horizontalmente. Para ello, se realizó un estudio individualizado de cada muestra para aplicar una rotación antes de proceder a recortar la ROI.

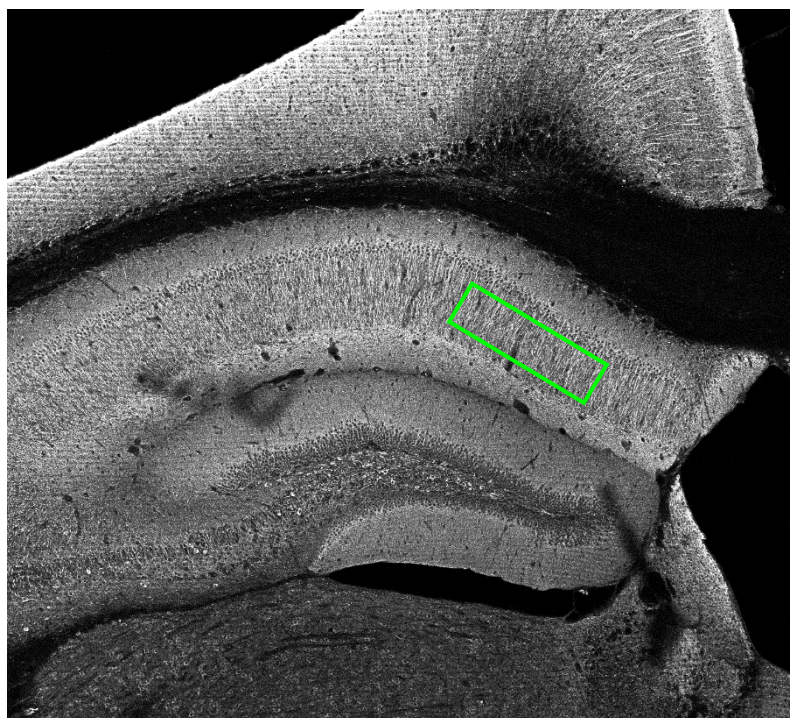


Figura 4.19. Aproximación a la región de interés.

A la hora de recortar las muestras se prestó máxima atención en abarcar la máxima zona dendrítica evitando los somas que se encuentran en la parte superior.

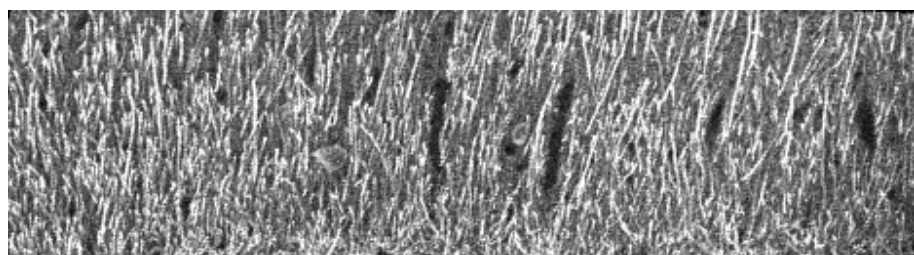


Figura 4.20. ROI trisómica.

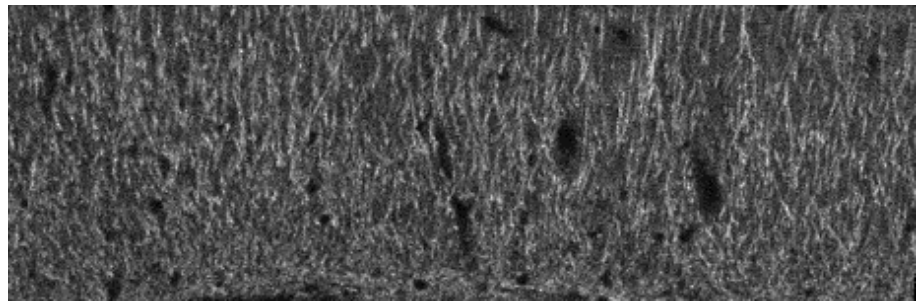


Figura 4.21. ROI wild type.

Como todas las muestras presentaban vasos sanguíneos, se decidió aplicar el algoritmo de Fast Image Inpainting [42]. Este algoritmo, a partir de una imagen y una máscara de puntos no deseados, realiza una interpolación de los valores que tendría la imagen en esos puntos a partir del resto de la imagen. A continuación se muestran algunos ejemplos de los resultados obtenidos en [42]:



Figura 4.22. Aplicación Fast Image Inpainting I [42].

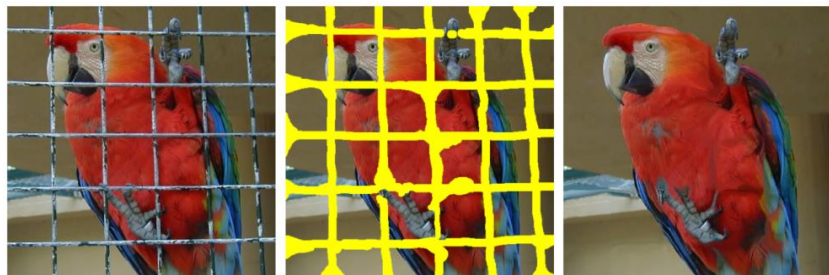
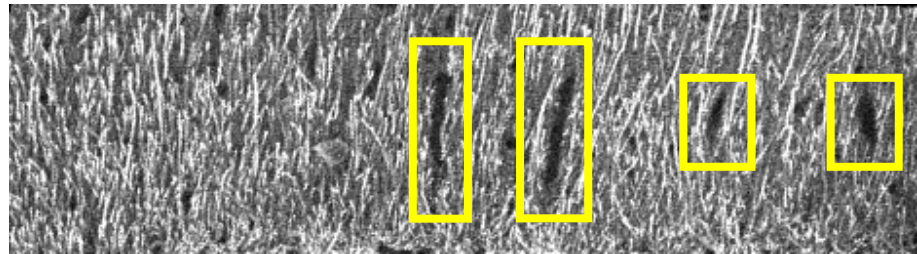
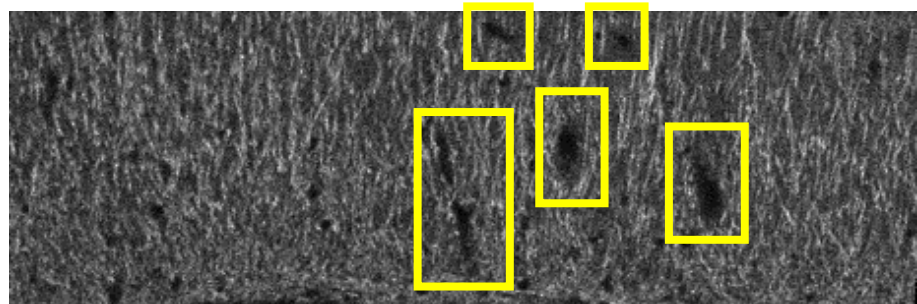


Figura 4.23. Aplicación Fast Image Inpainting II [42].

La figura 4.23 muestra el efecto de los vasos sanguíneos que cruzan el hipocampo. Se han recuadrado los vasos de mayor extensión que dificultarían la toma de decisión al clasificar.



(a)



(b)

Figura 4.24. Efecto vasos sanguíneos

Antes de aplicar el algoritmo, fue necesario umbralizar las imágenes para obtener una máscara con los puntos que formaban parte de los vasos sanguíneos. A continuación, se muestran los resultados obtenidos:

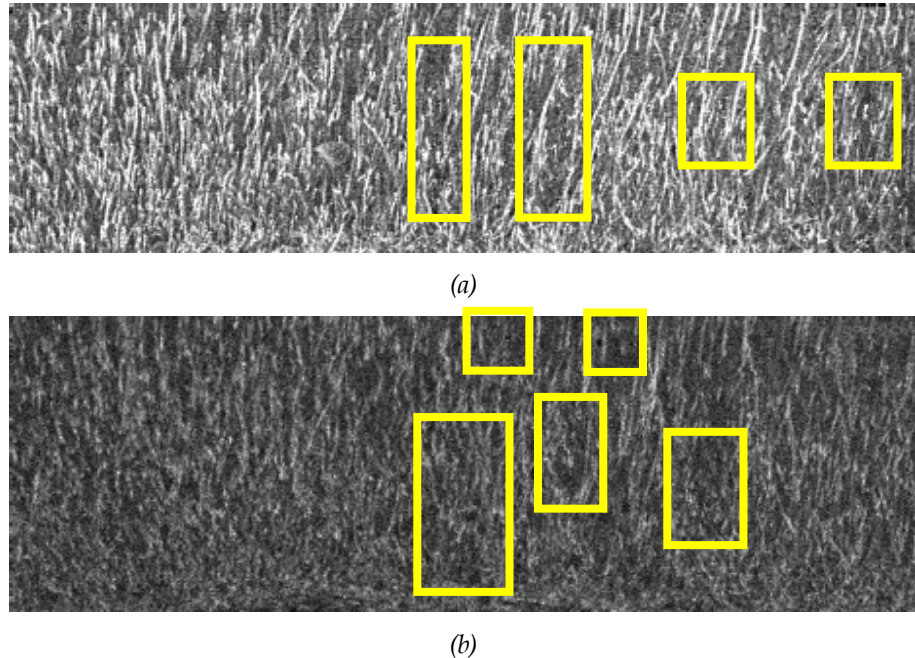


Figura 4.25. Resultados Fast Image Inpainting.

Destacar que se eliminaron sólo los vasos de mayor tamaño. Se buscaba el equilibrio entre la eliminación de los vasos sanguíneos y mantener la imagen tan original como fuese posible. A posteriori, se ha comprobado que los vasos de menor tamaño tienen una influencia mínima en el clasificador.

Una vez obtenidas todas las regiones de interés se procedió a igualar la intensidad y mejorar el contraste de las muestras. Para ello, se utilizó el algoritmo de ecualización adaptativa del histograma con limitación de contraste, por sus siglas en inglés, CLAHE [44].

La ecualización del histograma es un proceso por el cual se transforman los valores de los píxeles de una imagen para que exista el mismo número de píxeles para cada nivel de gris del histograma de una imagen.

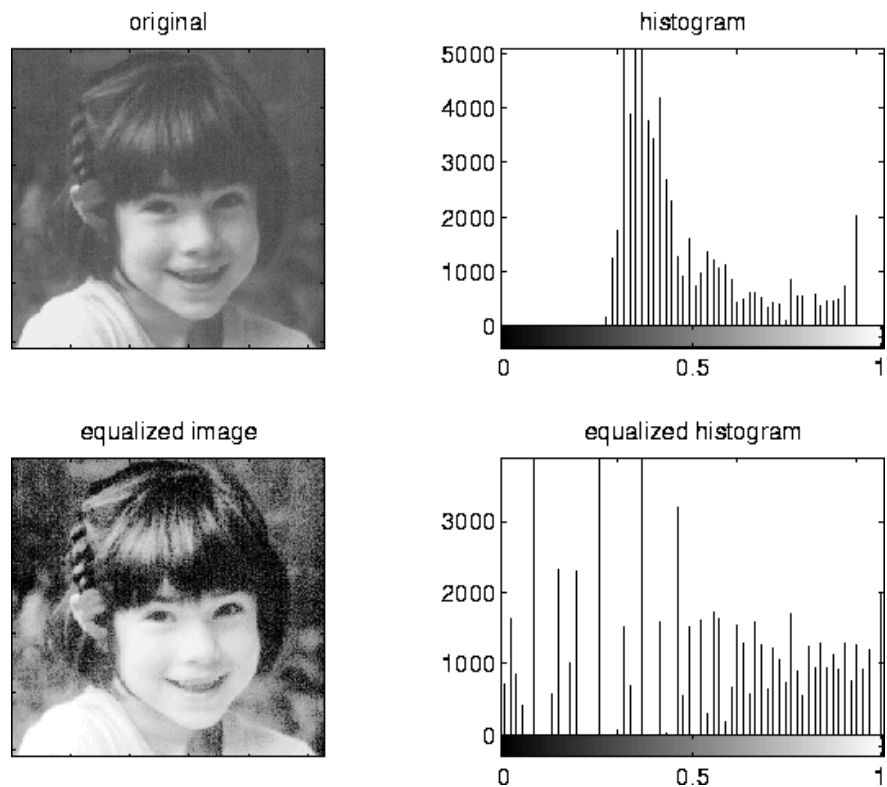


Figura 4.27. Ejemplo de ecualización de histograma [43].

Sin embargo, la ecualización del histograma básica sólo ofrece buenos resultados cuando la distribución de píxeles es similar en una imagen. CLAHE permite realzar los detalles cuando el fondo de la imagen no es homogénea realizando una ecualización del histograma por ventanas que se combina mediante una interpolación bilineal [44]. Además, evita la sobre amplificación del ruido en las zonas homogéneas limitando el contraste que es capaz de modificar.

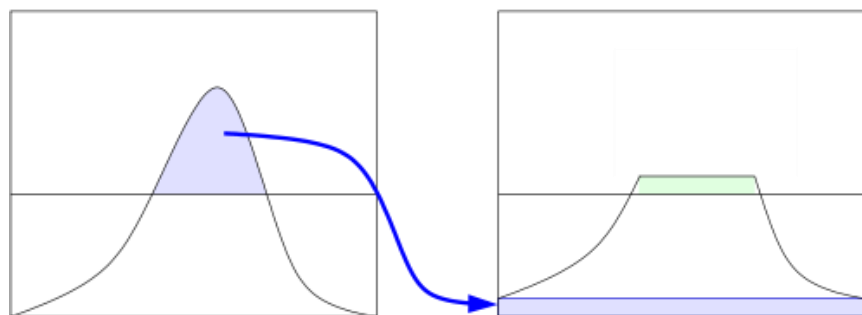


Figura 4.26. Ejemplo limitación del contraste en CLAHE [45].

A continuación se muestra el resultado obtenido después de aplicar el algoritmo CLAHE:

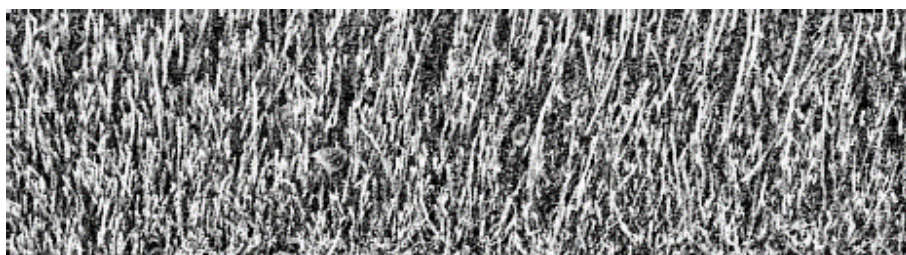


Figura 4.27. ROI trisómica final.

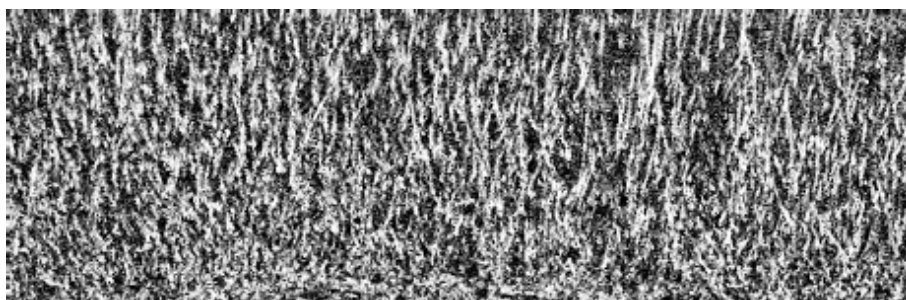


Figura 4.28. ROI wild type final.

El conjunto final de imágenes estaba formado por cinco muestras trisómicas y cinco muestras wild type. Se seleccionaron cuatro imágenes de cada tipo para obtener el conjunto de entrenamiento del clasificador.

Debido al reducido número de imágenes para entrenar y clasificar, se decidió dividir las muestras en bloques cuadrados de cincuenta píxeles sin superposición. Así, por cada imagen se obtuvieron dieciséis bloques, obteniendo finalmente 128 imágenes para entrenamiento y 32 para clasificación.

Recalcar que para el entrenamiento se usaron todos los bloques pertenecientes al conjunto de imágenes de entrenamiento, y respectivamente para la clasificación. Ningún bloque de una imagen de entrenamiento se ha usado en clasificación, o al revés.

4.3. Extracción de características

Dado que este proyecto pretende ser la primera fase de una larga colaboración, se decidió usar descriptores de texturas clásicos como primer estudio. Estos ya han mostrado su gran robustez y contrastado funcionamiento en la clasificación de imágenes.

A partir de las imágenes de entrenamiento se procedió a la extracción de características. De entre los descriptores de texturas estudiados en el capítulo 3.1, los finalmente usados fueron:

- Descriptores estadísticos de primer orden: media, desviación estándar, skewness y curtosis. Anteriormente definidas en las fórmulas (3.1-3.4).
- La particularización de la dimensión fractal denominada dimensión de Hausdorff [9], ya definida en el apartado correspondiente a métodos estructurales.
- Descriptores de texturas de Haralick [6] a partir de la matriz de co-ocurrencia (3.6-3.19).
- Media y varianza del vector textura, descrito a continuación.

El vector textura es un modelo estadístico de textura en el que se extrae de cada píxel de la imagen un vector que representa la textura de la vecindad centrada en dicho píxel [46]. Se opta por un modelo de textura basada en la vecindad detallado por Scharfenberger en [47] que es adecuado para representar texturas locales.

La representación de textura es dispersa y radialmente ordenada. Sea I una imagen en escala de gris, la representación textural de la vecindad centrada en el píxel x es:

$$h(x) = [I(x) \text{ ordenación}_{\uparrow}(I(x_{1,j})) \text{ ordenación}_{\uparrow}(I(x_{2,j})) \dots \text{ ordenación}_{\uparrow}(I(x_{n,j}))]$$

donde $x_{i,j}$ es el conjunto de píxeles que forman la capa radial i sobre el píxel x . Es decir, es el vector formado por el valor del píxel x , seguido de la ordenación ascendente de los píxeles de sus capas radiales, empezando por la más cercana.

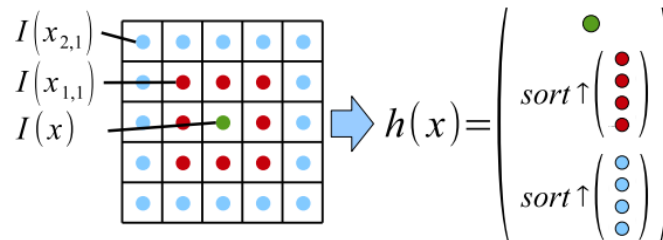


Figura 4.29. Ordenación [46].

Además, con objeto de extraer el vector textura de los píxeles pertenecientes a la periferia de la imagen, se emplea una imagen en la que los bordes son extendidos: se copia el valor de los píxeles de los bordes, primero los horizontales y después los verticales, formándose una imagen de la que es posible extraer tantos vectores textura como píxeles tiene la imagen original.

Eligiendo una vecindad cuadrada 5x5 se obtiene, por tanto, para cada píxel de la imagen un vector textura $h(x)$ formado por 25 elementos. Puesto que cada bloque tiene una dimensión 50x50, estamos trabajando con 2500 vectores textura.

Dada la cantidad y la dimensión de los vectores textura obtenidos, en este paso se reduce el número de componentes de los mismos con el fin de mejorar la eficiencia del algoritmo. Para este fin, se realiza un Análisis de Componentes Principales (PCA) para reducir su dimensión a cinco componentes.

La técnica PCA [48] expresa los datos cuyas dimensiones se desean reducir en términos de sus principales autovectores. Para ello, tras extraer la media a cada vector textura $h(x)$, se calculan los autovectores de su matriz de covarianza. Aquellos autovectores con mayores autovalores representan

los componentes principales de los datos, esto es, los que contienen más información. Si éstos se agrupan formando las filas de una matriz de transformación y ésta se multiplica a cada vector textura sin media, se está realizando una transformación en la que los nuevos vectores textura $h'(x)$ quedan expresados en términos de los autovectores elegidos y su dimensión se ve reducida a cinco componentes.

Finalmente, cada componente se modela como una función gaussiana con media μ_i y desviación típica σ_i . Las características basadas en el vector textura se componen de la media y desviación típica de las cinco componentes.

4.4. *Entrenamiento*

Una vez que se realizó la selección de características se procedió al entrenamiento del clasificador. Inicialmente el clasificador no tiene ningún conocimiento y necesitamos que *aprenda* a clasificar entre las diferentes clases. Para ello, hay que introducirle la matriz de características de los bloques de entrenamiento y una columna de etiquetas que indican el tipo de datos a clasificar, en este caso indicando si son wild type o trisómica.

Internamente, el clasificador analiza el conjunto de datos y reconoce patrones, dividiendo así el espacio de muestras en tantas regiones como tipos de clases.

Entre los clasificadores estudiados en el capítulo 3.3, finalmente se hicieron uso de los siguientes:

- Support Vector Machine (SVM) [34]. En sus versiones de kernel lineal, cuadrático, cúbico y gaussiano.
- K-Nearest Neighbor (KNN). En sus versiones de distancia mínima de coseno y cúbica. [30]
- Complex Tree (un solo árbol de decisión)
- Bagged Tree. [38]
- AdaBoost Tree. [39]
- LogiBoost Tree. [40]
- Random Forests [41].

4.5. *Clasificación*

Una vez que tenemos el clasificador, sólo es necesario obtener la matriz de características de las imágenes del conjunto de clasificación mediante los algoritmos del apartado 4.3.

Recalcar que llegados a este punto tendremos una clasificación automática de hipocampos murino. Los resultados obtenidos se analizarán en el siguiente capítulo.

5. Resultados

Para la valoración de los resultados se hará uso de tablas de confusión. También incluiremos gráficas ROC (Receiver Operating Characteristic). Las tablas de confusión recogen información acerca de las predicciones de la siguiente forma:

Realidad	Clase 1	Verdaderos Positivos (%)	Falsos Positivos (%)	VP FP
	Clase 2	Falsos Negativos (%)	Verdaderos Negativos (%)	FN VN
		Clase 1	Clase 2	VPR/FPR
		Predicción		

Figura 5.1 Modelo tabla de confusión.

Los términos que se usaran en este capítulo son:

- VP: Verdaderos Positivos.
- FP: Falsos Positivos.
- FN: Falsos Negativos.
- VN: Verdaderos Negativos.
- VPR: Ratio Verdaderos Positivos.
- FPR: Ratio Falsos Positivos.

Se presentaran dos simulaciones distintas, cada una con un número de muestras diferentes, para comprobar la fiabilidad de los distintos clasificadores.

En la primera simulación, se utilizarán seis imágenes para el entrenamiento y cuatro imágenes para clasificación, utilizando 96 y 64 muestras respectivamente. En la segunda simulación, sin embargo se utilizarán ocho imágenes para entrenamiento y dos imágenes para clasificación, usando respectivamente 128 y 32 muestras.

5.1. Grupo entrenamiento de 6 muestras

En primer lugar vamos a mostrar los resultados obtenidos entrenando con 6 muestras (96 bloques) y clasificando con 4 muestras (64 bloques). Recordar que hemos hecho un tratamiento por bloques, por tanto los porcentajes hacen referencia al número de bloques acertados/errados:

- Tabla de confusión SVM (Lineal Kernel): Observamos que el clasificador ofrece una probabilidad de acierto del 96.88%, con sólo dos falsos positivos.

Realidad	TR	30 93.75%	2 6.25%	93.75% 6.25%
	WT	0 0%	32 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión SVM (Quadratic Kernel): se duplican los falsos positivos. Aun así, se obtiene un 93.75% de acierto.

Realidad	TR	28 87.5%	4 12.5%	87.5% 12.5%
	WT	0 0%	32 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión SVM (Cubic Kernel): obtenemos los mismos resultados que el kernel cuadrático.

Realidad	TR	28 87.5%	4 12.5%	87.5% 12.5%
	WT	0 0%	32 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión SVM (Gaussian Kernel): Ofrece una probabilidad de acierto del 92.19%. Presenta cuatro falsos positivos y un falso negativo.

Realidad	TR	28 87.5%	4 12.5%	87.5% 12.5%
	WT	1 3.12%	31 96.87%	96.87% 3.12%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión KNN (Minkowski Dimension Metric): aumentan los falsos positivos/negativos, y por tanto, la tasa de error. La probabilidad de acierto disminuye hasta el 75.00%, siendo el peor clasificador de todos.

Realidad	TR	23 71.87%	9 28.12%	71.87% 28.12%
	WT	7 21.87%	25 78.12%	78.12% 21.87%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión KNN (Cosine Dimension Metric): con unos porcentajes cercanos al modelo anterior, aumenta levemente la probabilidad de acierto, alcanzando el 78.13%

Realidad	TR	26 81.2%	6 18.7%	81.2% 18.7%
	WT	8 25%	24 75%	75% 25%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión Complex Tree: junto a los clasificadores KNN es de los que peores resultados ofrece con un 76.56% de acierto.

Realidad	TR	21 65.62%	11 34.37%	65.62% 34.37%
	WT	4 12.5%	28 87.5%	87.5% 12.5%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión Bagged Tree: aunque los falsos positivos son demasiado, sigue clasificando correctamente el 85.94% de las muestras.

Realidad	TR	26 81.2%	6 18.7%	81.2% 18.7%
	WT	3 9.37%	29 90.62%	90.62% 9.37%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión AdaBoost Tree: resultados muy parecidos a Bagged Tree, con un porcentaje de acierto del 87.50%.

Realidad	TR	26 81.2%	6 18.7%	81.2% 18.7%
	WT	2 6.25%	30 93.75%	93.75% 18.75%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión LogiBoost Tree: ofrece los mismos resultados que Bagged Tree, obteniendo una tasa del 85.94% de acierto.

Realidad	TR	26 81.2%	6 18.7%	81.2% 18.7%
	WT	3 9.37%	29 90.62%	90.62% 9.37%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión Random Forests: de los mejores resultados junto a los clasificadores SVM. Probabilidad de acierto: 89.06%

Realidad	TR	28 87.5%	4 12.5%	87.5% 12.5%
	WT	3 9.37%	29 90.62%	90.62% 9.37%
		TR	WT	VPR/FPR
		Predicción		

Como muestra las figura 5.2 y 5.3, podemos determinar que el mejor clasificador, muy cercano al óptimo, es SVM con kernel lineal. Dentro de los clasificadores con buenos resultados están SVM con kernel cuadrático, cúbico y gaussiano. También, podemos incluir en ese grupo a Random Forest. Dado los resultados, podríamos descartar los clasificadores obtenidos mediante los métodos k-NN y árboles de decisión simples.

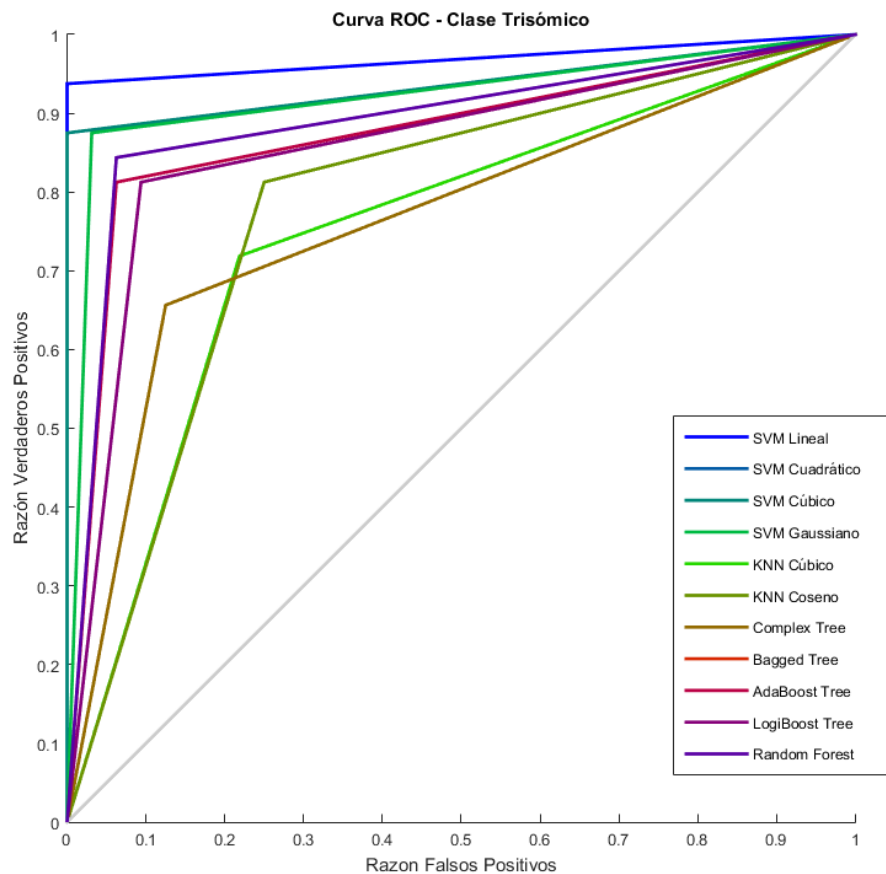


Figura 5.2. Curva ROC clase TR.

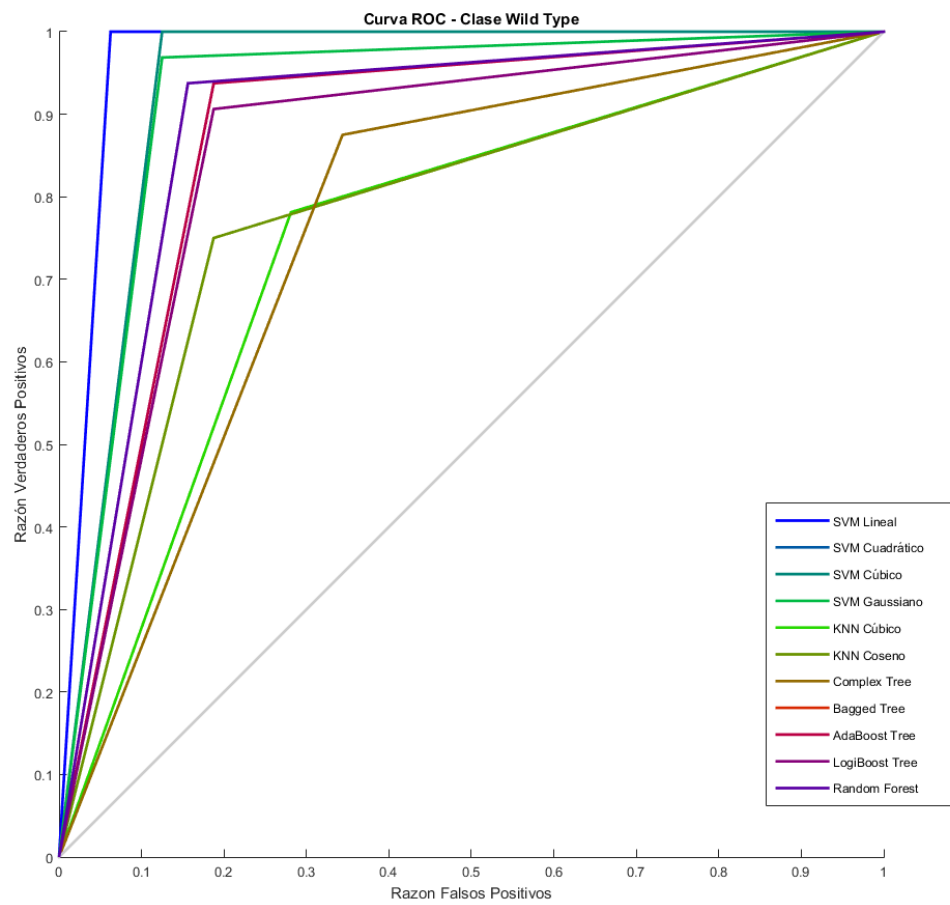


Figura 5.3. Curva ROC clase WT.

5.2. Grupo entrenamiento de 8 muestras

Continuaremos mostrando las tablas de confusión obtenidas a partir de 8 muestras (128 bloques) en el entrenamiento y 2 muestras (32 bloques) en la clasificación:

Observamos que para todos los clasificadores SVM los resultados son óptimos ya que todas las muestras se han clasificado correctamente:

- Tabla de confusión SVM (Lineal Kernel): Probabilidad de acierto del 100%.

Realidad	TR	16 100%	0 0%	100% 0%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión SVM (Quadratic Kernel): Probabilidad de acierto del 100%.

Realidad	TR	16 100%	0 0%	100% 0%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión SVM (Cubic Kernel): Probabilidad de acierto del 100%.

Realidad	TR	16 100%	0 0%	100% 0%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión SVM (Gaussian Kernel): Probabilidad de acierto del 100%.

Realidad	TR	16 100%	0 0%	100% 0%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión KNN (Minkowski Dimension Metric): presenta dos falsos positivos y tres falsos negativos. La probabilidad de acierto es del 84.38%

Realidad	TR	14 87.5%	2 12.5%	87.5%	12.5%
	WT	3 18.75%	13 81.25%	81.25%	18.75%
		TR	WT	VPR/FPR	
		Predicción			

- Tabla de confusión KNN (Cosine Dimension Metric): aumentan los falsos negativos respecto al clasificador anterior, obteniendo así el peor porcentaje de todos los clasificadores: 78.13 %

Realidad	TR	14 87.5%	2 12.5%	87.5%	12.5%
	WT	5 31.25%	11 68.75%	68.75%	31.25%
		TR	WT	VPR/FPR	
		Predicción			

Excepto Complex Tree, que tiene una alta probabilidad de acierto, el resto de clasificadores basados en árboles de decisión tienen una porcentaje de acierto del 100%.

- Tabla de confusión Complex Tree: con un solo falso positivo obtenemos un acierto del 96.88% de las muestras.

Realidad	TR	15 93.75%	1 6.25%	93.75% 6.25%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión Bagged Tree: Probabilidad de acierto del 100%.

Realidad	TR	16 100%	0 0%	100% 0%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión AdaBoost Tree: Probabilidad de acierto del 100%.

Realidad	TR	16 100%	0 0%	100% 0%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión logiBoost Tree: Probabilidad de acierto del 100%.

Realidad	TR	16 100%	0 0%	100% 0%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

- Tabla de confusión Random Forests: Probabilidad de acierto del 100%.

Realidad	TR	16 100%	0 0%	100% 0%
	WT	0 0%	16 100%	100% 0%
		TR	WT	VPR/FPR
		Predicción		

Según la curva ROC para ambas clases, ocho de los once clasificadores son óptimos y el árbol de decisión simple obtiene una probabilidad de acierto muy alta (96.88%). Otra vez más, los clasificadores que peor rendimiento ofrece son los basados en k-NN:

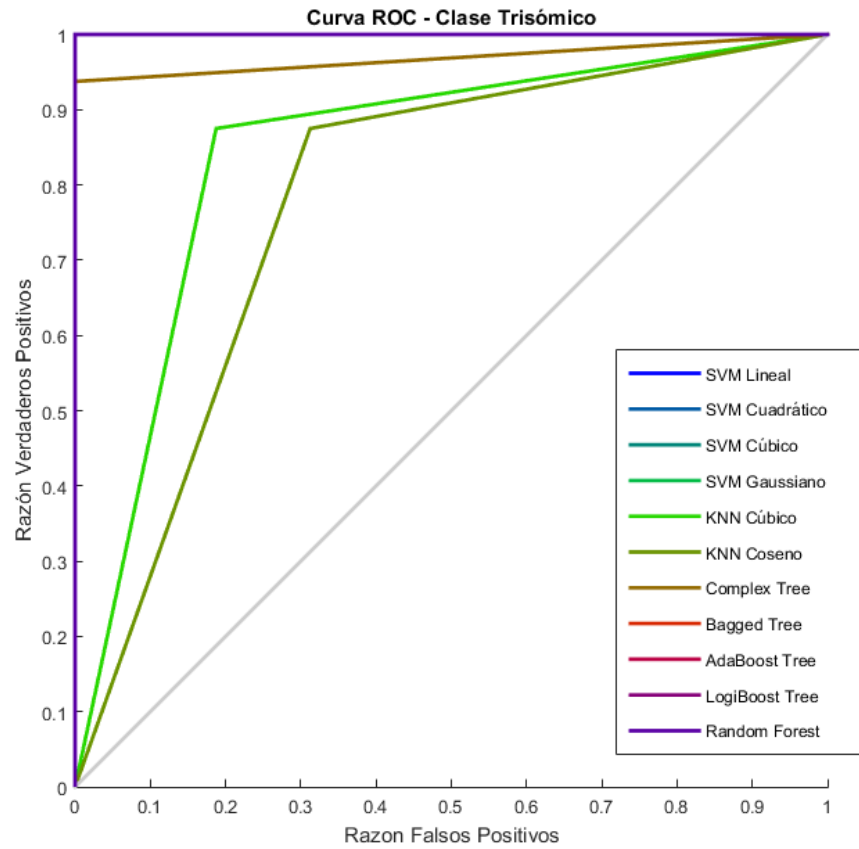


Figura 5.4. Curva ROC clase TR.

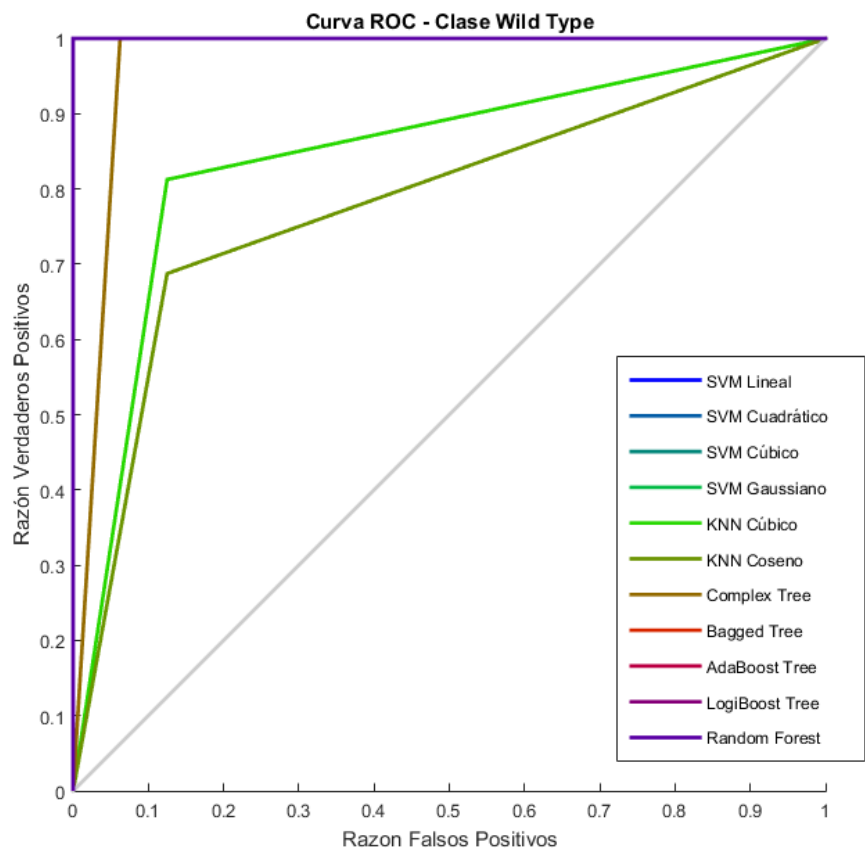


Figura 5.5. Curva ROC clase WT.

6. Conclusiones y líneas futuras

Podemos considerar que el trabajo desarrollado ha tenido unos resultados más que positivos teniendo en cuenta que el planteamiento del problema era una nueva línea de investigación totalmente desconocida hasta ahora.

Los clasificadores que mejores resultados han obtenido, y que por tanto, ha destacado más en la clasificación de patrones de neuronas murino mediante descriptores de texturas son los basados en Vector Support Machine. Destacar también, la fiabilidad de los clasificadores basados en árboles de decisión como Random Forest y AdaBoost Tree.

Como líneas futuras se plantea:

- Modificar el algoritmo para desarrollar técnicas más novedosas como LBP [10], ya que los descriptores de texturas fueron definidos hace más de cuatro décadas.
- Implementar un algoritmo de segmentación automática de la región de interés.
- Mejorar el pre-procesado eliminando las marcas producidas por el vibratomo.
- Estudiar patrones dendríticos en la morfología dendrítica de pacientes con Esclerosis Lateral Amiotrófica (ELA).
- Aumentar la base de datos de imágenes con el objetivo de realizar más pruebas y tener una mayor robustez del sistema.

7. Referencias

[1] Originally Neuron.jpg taken from the US Federal (public domain), redrawn by User:Dhp1080 in Illustrator. Traducida al castellano por User:Acracia. Wikipedia.

[2] Neil R. Carlson: *Fisiología de la conducta*. Addison Wesley, 2009.

[3] A. Alves-Sampaio, J.A. Troca-Marín, M. L.Montesinos: *NMDA-Mediated Regulation of DSCAM Dendritic Local Translation Is Lost in a Mouse Model of Down's Syndrome*. The Journal of Neuroscience, October 6, 2010.

[4] H. V. Carter, H. Gray: *Anatomy of the Human Body*, 1918. Gray739-emphasizing-hippocampus.png Disponible bajo la licencia Dominio público vía Wikimedia Commons.

[5] MathWorks Documentation: Gray-Level Co-Occurrence Matrix (GLCM). Enlace web disponible en: <http://es.mathworks.com/help/images/gray-level-co-occurrence-matrix-glcm.html>

[6] R. M. Haralick, K. Shanmugam, I. Dinstein: *Textural Features for Image Clasification*. IEEE Transactions on systems, man and cybernetics, 1973.

[7] Khaled Hammouda: Texture Segmentation Using Gabor Filters. Department of Systems Design Engineering. University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

[8] Fractal Dimension. En Wikipedia. Última consulta el 12 de Julio de 2015. Enlace disponible en: https://en.wikipedia.org/wiki/Fractal_dimension

[9] A. F. Costa, G. Humpire-Mamani, A. J. M. Traina: *An Efficient Algorithm for Fractal Analysis of Textures*. Brazil, 2012.

[10] T. Ojala, M. Pietikainen, and D. Harwood: Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR 1994), vol. 1, pp. 582 - 585.

[11] *Computer vision using LBP*. Web personal de Abdenour Hadid, Senior Research Scientist. University of Oulu, Finland. Última consulta el 12 de Julio de 2015. Enlace disponible en: <http://www.ee.oulu.fi/~hadid/image439.png>

[12] Wei Xia, Shouyi Yin, Peng Ouyang: *A High Precision Feature Based on LBP and Gabor Theory for Face Recognition*. Sensors 13, 4499-4513, 2013.

[13] *Local Binary Patterns*. En Scholarpedia. Última consulta el 14 de Julio de 2015. Enlace disponible en: http://www.scholarpedia.org/article/Local_Binary_Patterns

[14] Ville Ojansivu and Janne Heikkilä: *Blur Insensitive Texture Classification Using Local Phase Quantization*. Machine Vision Group, Department of Electrical and Information Engineering, University of Oulu, Finland.

[15] Timo Ahonen, Jiri Matas, Chu He and Matti Pietikainen: *Rotation Invariant Image Description with Local Binary Pattern Histogram Fourier Features*. Springer: Image Analysis. Lecture Notes in Computer Science Volume 5575, pp 61-70, 2009.

[16] D. Svoboda, M. Kozubek and S. Stejskal: *Generation of digital phantoms of cell nuclei and simulation of image formation in 3D image cytometry*, *Cytometry A*, vol. 75, n° 6, pp. 494–509, 2009.

[17] O. S. Al-Kadi: *Texture measures combination for improved meningioma classification of histopathological images*, *Pattern Recognition*, vol. 43, pp. 2043–2053, May 2010.

[18] H. Xia, L. Tong, X. Zhou, J. Zhang, Z. Zhou and W. Liu: *Texture Analysis and Volumetry of Hippocampus and Medial Temporal Lobe in Patients with Alzheimer's Disease*, in *Proc. of 2012 International Conference on Biomedical Engineering and Biotechnology*, pp. 905-908, 2012.

[19] G. Jiang, X. Wang, L. Tong: *Study of correlation between hippocampus's texture and morphological features in alzheimer's disease patients based on MR image*. *IFMBE Proceedings*, v 25, n 4, p 1908-1911, 2009.

[20] S. S. Kaushik, J. Sivaswamy: *Unsupervised 3D segmentation of hippocampus in brain MR images*. *International Conference on Bio-Inspired Systems and Signal Processing*, p 182-187, 2011, BIOSIGNALS 2011.

[21] M. Hajiesmaeili, J. Dehmeshki, B. Bagheri Nakhjavanlo, T. Ellis: *Initialisation of 3D level set for hippocampus segmentation from volumetric brain MR images*. *The International Society for Optical Engineering*, v 9159, 2014.

[22] G. Thibault, J. Angulo and F. Meyer: *Advanced Statistical Matrices for Texture Characterization: Application to Cell Classification*, *IEEE Trans. On Biomedical Engineering*, vol. 61 n° 3, 2014

[23] K. S. Angel Viji, J. Jayakumari: *Modified LBP Based Region Growing Segmentation of MR Brain Images*. *Lecture Notes in Electrical Engineering Volume 326*, 2015, pp 1453-1465, 2014.

[24] Isaac Castro-Mateos, Jose M Pozo, Peter E Eltes, Luis Del Rio, Aron Lazary and Alejandro F Frangi: *3D segmentation of annulus fibrosus and nucleus pulposus from T2-weighted magnetic resonance images*. Physics in Medicine and Biology Volume 59 Number 24.

[25] Ketil Oppedal, Trygve Eftestøl, Kjersti Engan, Mona K. Beyer, and Dag Aarsland: *Classifying Dementia Using Local Binary Patterns from Different Regions in Magnetic Resonance Images*. *International Journal of Biomedical Imaging*, 2015.

[26] Liu, Jun; Jing, Xiaojun; Sun, Songlin; Lian, Zifeng: *Local Gabor dominant direction pattern for face recognition*. Source: Chinese Journal of Electronics, v 24, n 2, p 245-250, April 10, 2015.

[27] Hu, Guang-Hua: *Automated defect detection in textured surfaces using optimal elliptical Gabor filters*. Source: Optik, v 126, n 14, p 1331-1340, July 1, 2015

[28] Chakraborty, Jayasree; Midya, Abhishek; Mukhopadhyay, Sudipta; Rangayyan, Rangaraj M.; Sadhu, Anup; Singla, Veenu; Khandelwal, Niranjana; Bhattacharyya, Pinakpani; Azevedo-Marques, Paulo M: *Detection of the nipple in mammograms with Gabor filters and the Radon transform*. Source: Biomedical Signal Processing and Control, v 15, p 80-89, January 2015.

[29] Matthew D. Budde, Joseph A. Frank: *Examining brain microstructure using structure tensor analysis of histological sections*. NeuroImage Volume 63, Issue 1, 15 Pages 1-10, October 2012.

[30] N. S. Altman: *An introduction to kernel and nearest-neighbor nonparametric regression*. The American Statistician 46 (3): 175-185, 1992.

[31] Classification Using Nearest Neighbors. En MathWorks. Última consulta 14 de Julio de 2015. Enlace web disponible: http://es.mathworks.com/help/releases/R2015a/examples/stats/ClassifyingQueryDataUsingKnnsearchExample_05.png

[31] *Artificial neural network*. En Wikipedia. Última consulta el 13 de Julio de 2015. Enlace web disponible en: https://upload.wikimedia.org/wikipedia/commons/thumb/4/46/Colored_neural_network.svg/450px-Colored_neural_network.svg.png

[33] N. Casillas: sistema basado en redes neuronales para el reconocimiento de dígitos manuscritos. Proyecto Final de Carrera. Escuela Politécnica Superior, Universidad Carlos III de Madrid, 2012.

[34] C. Cortes, V. Vapnik: *Support-Vector Networks*. AT&T Bell Labs (USA), 1995.

[35] Robert C. Berwick: An Idiot's guide to Support vector machines. Diapositivas. Última consulta el 14 de Julio de 2015. Enlace web disponible en: web.mit.edu/6.034/www/bob/svm-notes-long-08.pdf

[36] StatSoft (Dell Software): Support Vector Machines (SVM) Introductory Overview. Última consulta el 14 de Julio de 2015. Enlace web disponible en: <http://www.statsoft.com/textbook/graphics/SVMIntro3.gif>

[37] Tad Slaff: Trading the RSI using a Support Vector Machine. Última consulta el 14 de Julio de 2015. Enlace web disponible en: <https://media.licdn.com/mpr/mpr/p/1/005/097/185/267ca63.jpg>

[38] L. Breiman: *Bagging predictors*. Technical Report No. 421. Department of Statistics, University of California, 1994.

[39] Y. Freund, R. E. Schapire: *A Short Introduction to Boosting*. AT&T Labs - Research Shannon Laboratory, 1999.

[40] J. H. Friedman, T. J. Hastie, R. Tibshirani: *Additive logistic regression: a statistical view of boosting*. The Annals of Statistics, 28(2):337– 407, 2000.

[41] L. Breiman: *Random Forests*. University of California, 2001.

[42] Folkmar Bornemann, Tom März: *Fast Image Inpainting Based on Coherence Transport*. Springer Science+Business Media, 2007.

[43] *Histogram Equalization*. Última consulta 14 de Julio de 2015. Enlace disponible en: http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT5/node3.html

[44] Contrast-limited adaptive histogram equalization (CLAHE). En MathWorks. Última consulta 14 de Julio de 2015. Enlace disponible en: <http://es.mathworks.com/help/images/ref/adapthisteq.html>

[45] *Contrast-limited adaptive histogram equalization (CLAHE)*. En Wikipedia. Última consulta 14 Julio de 2015. Enlace disponible en: <https://upload.wikimedia.org/wikipedia/commons/thumb/5/5f/Clahe-redist.svg/450px-Clahe-redist.svg.png>

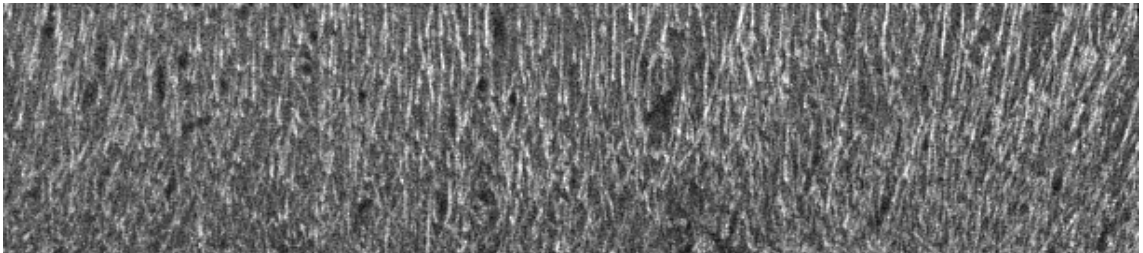
[46] M^aM. Velasco: *Algoritmo automático para la detección de melanoma basado en características de color y textura*. Universidad de Sevilla, 2014.

[47] C. Scharfenberger: *Statistical Textural Distinctiveness for Salient Region Detection in Natural Images*. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

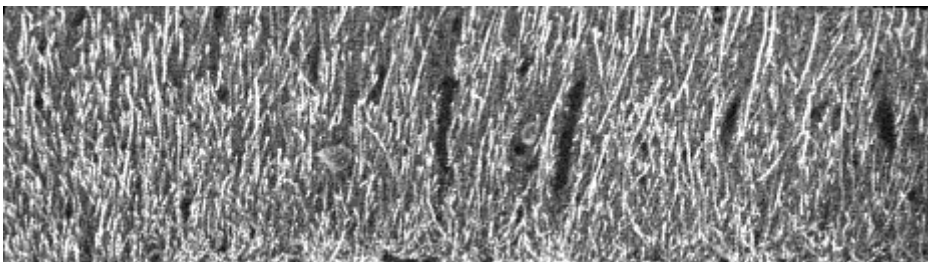
[48] Smith, Lindsay I. Department of Computer Science. University of Otago, 2014. Enlace web disponible en: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

8. Anexo I: ROI de imágenes originales

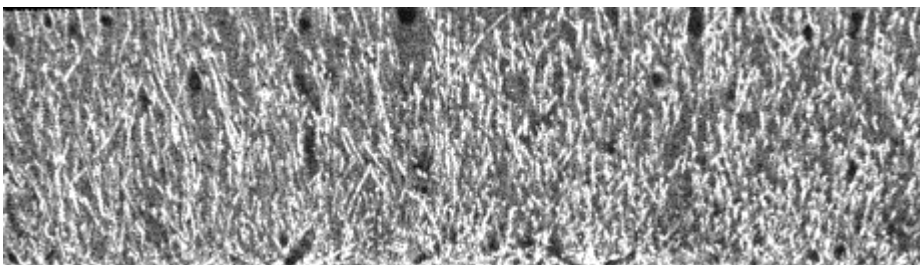
tr0_im1_z04:



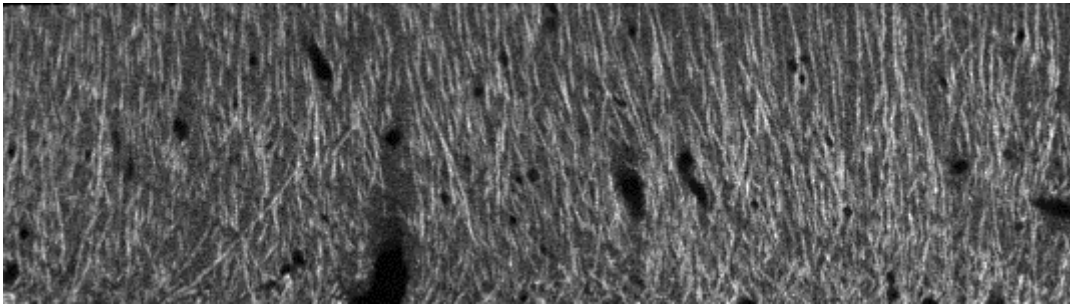
tr1_im2_z04 hipocampo izquierdo:



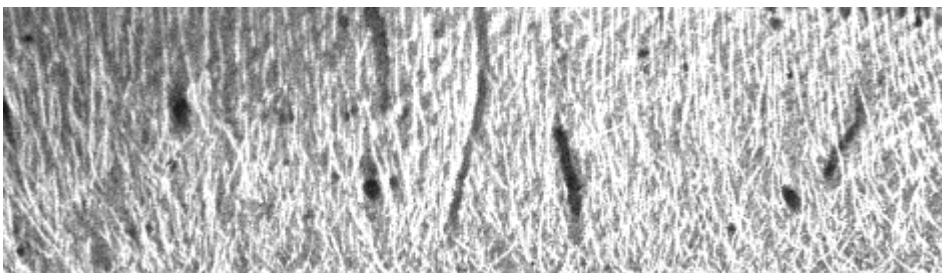
tr1_im2_z04 hipocampo derecho:



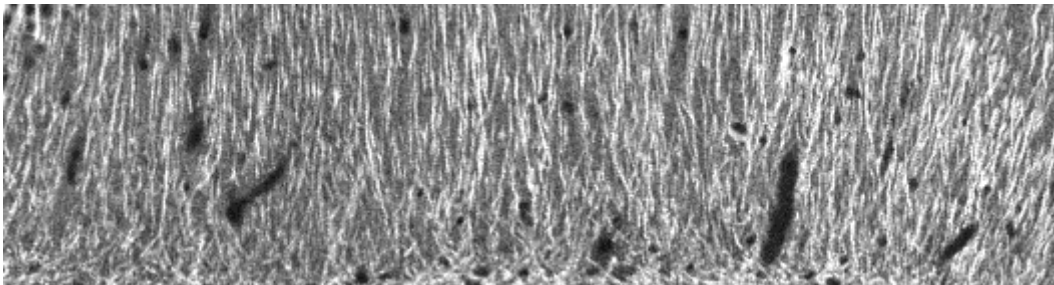
tr2_im1_z04:



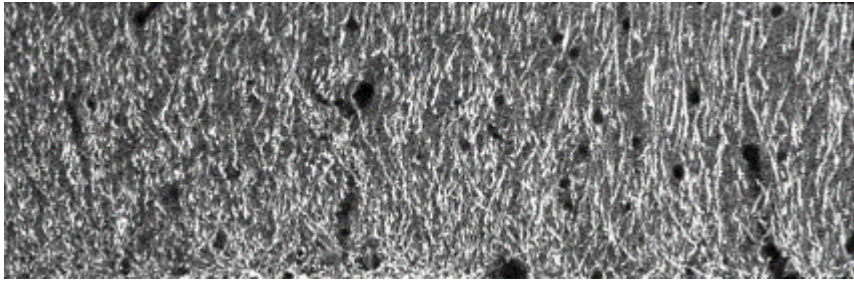
tr3_im2_z04 hipocampo izquierdo:



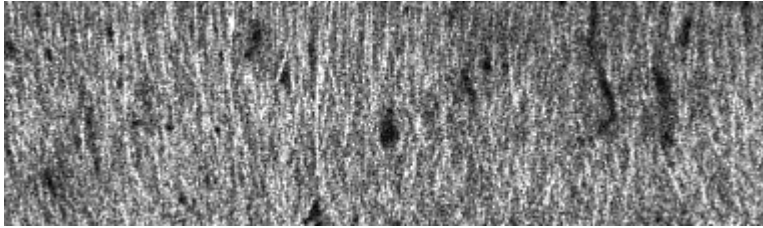
tr3_im2_z04 hipocampo derecho:



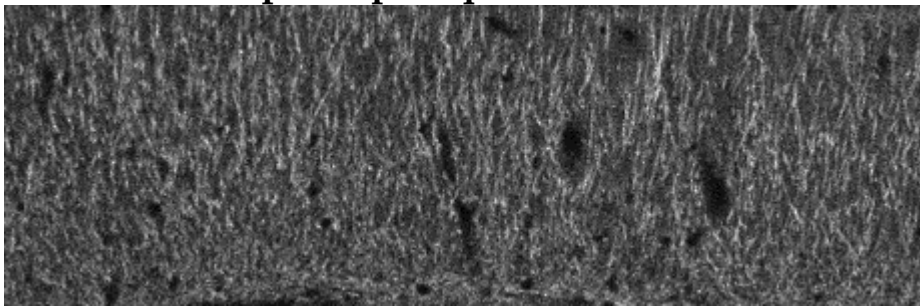
wt0_im1_z04:



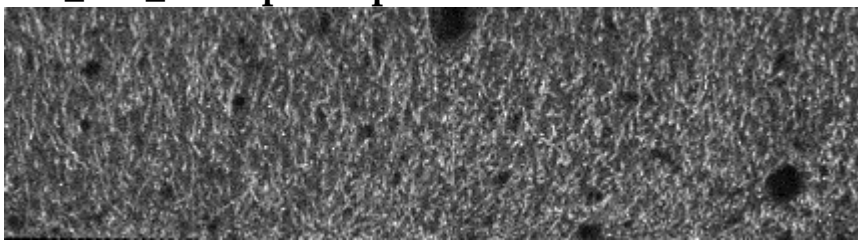
wt1_im1_z04:



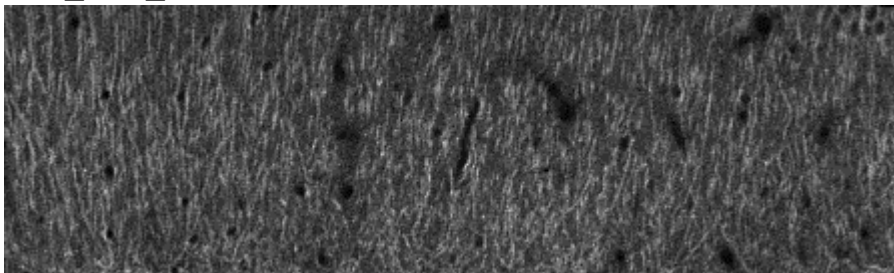
wt1_im2_z04 hipocampo izquierdo:



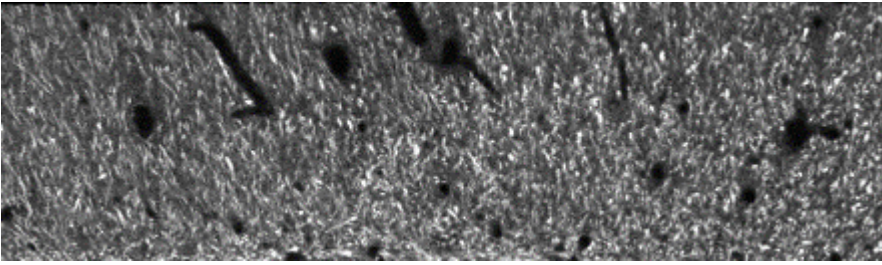
wt1_im2_z04 hipocampo derecho:



wt2_im2_z04:

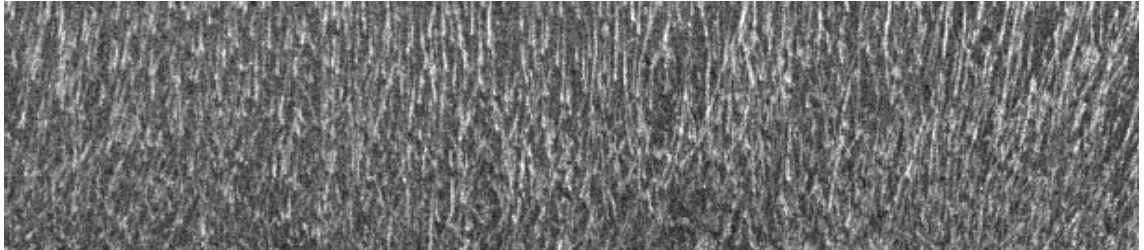


wt3_im1_z04:

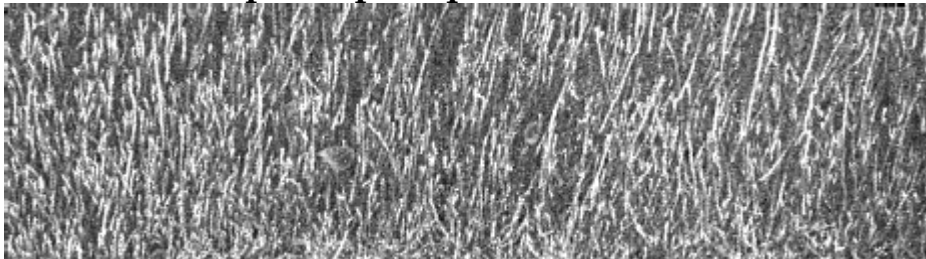


9. Anexo II: ROI con FI Inpainting

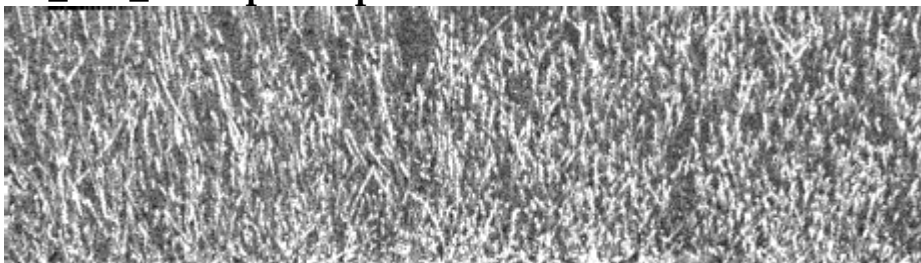
tr0_im1_z04:



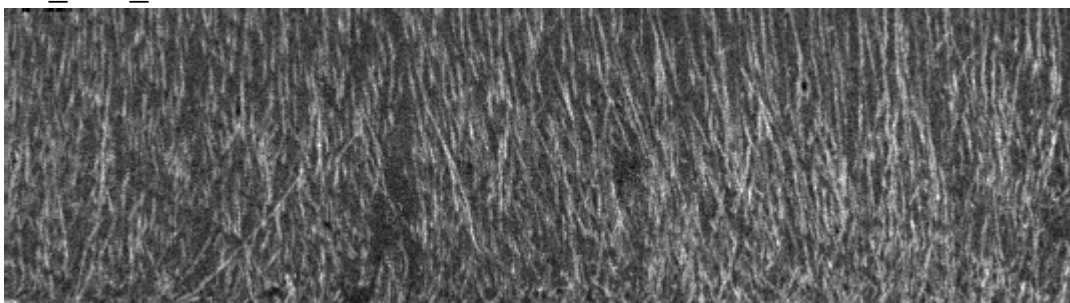
tr1_im2_z04 hipocampo izquierdo:



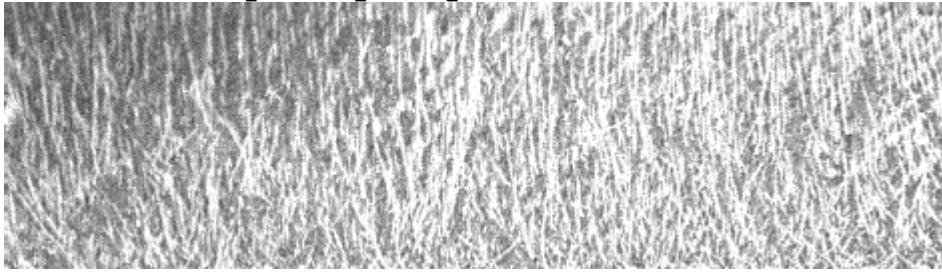
tr1_im2_z04 hipocampo derecho:



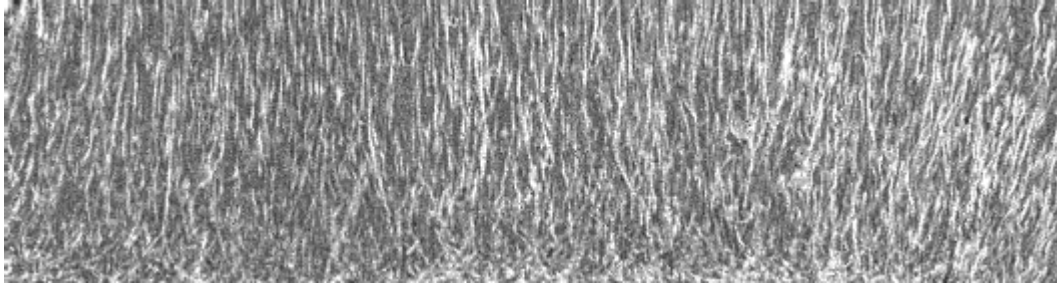
tr2_im1_z04:



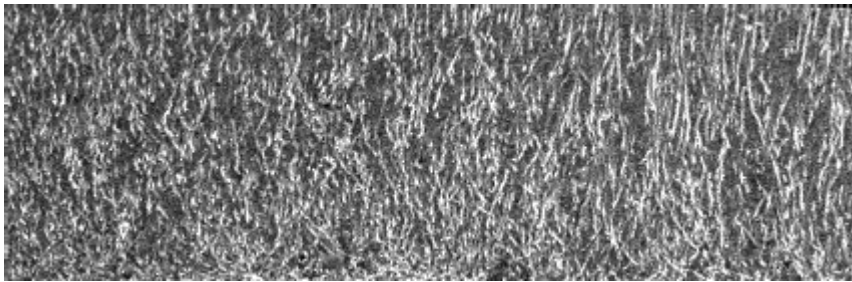
tr3_im2_z04 hipocampo izquierdo:



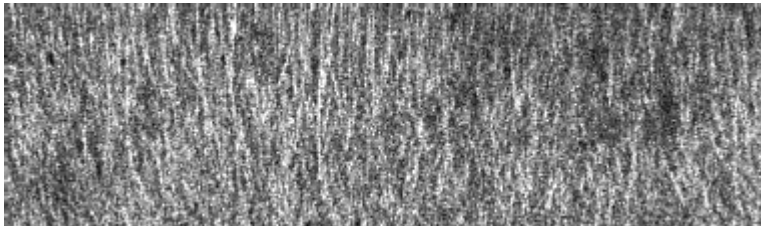
tr3_im2_z04 hipocampo derecho:



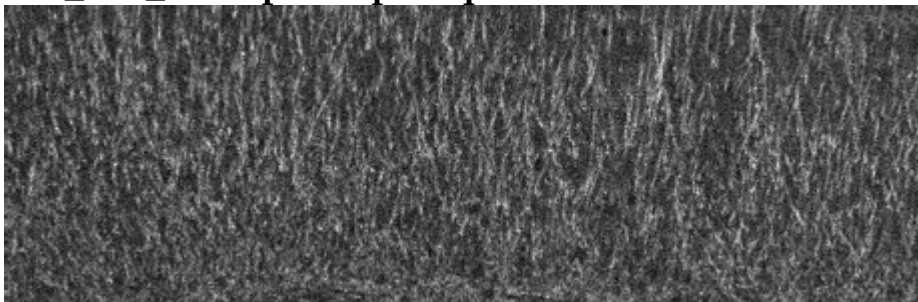
wt0_im1_z04:



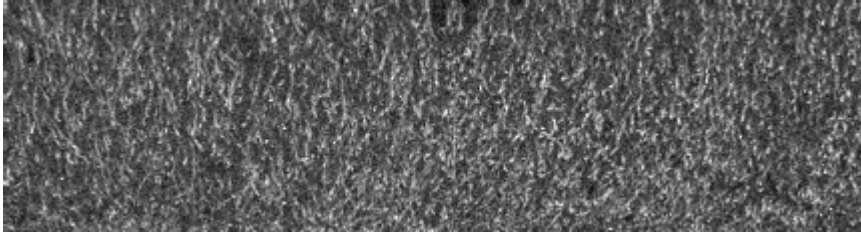
wt1_im1_z04:



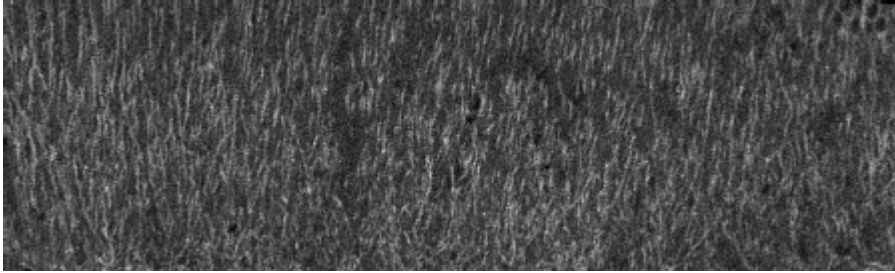
wt1_im2_z04 hipocampo izquierdo:



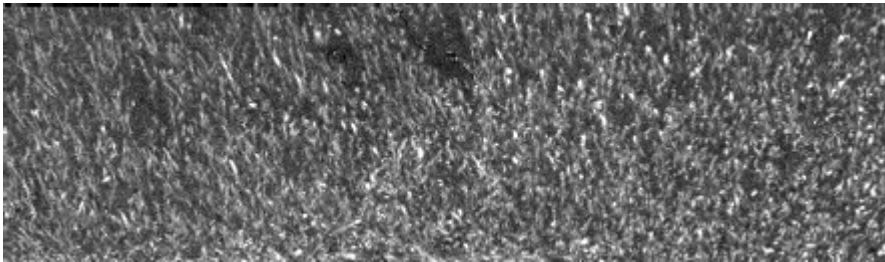
wt1_im2_z04 hipocampo derecho:



wt2_im2_z04:

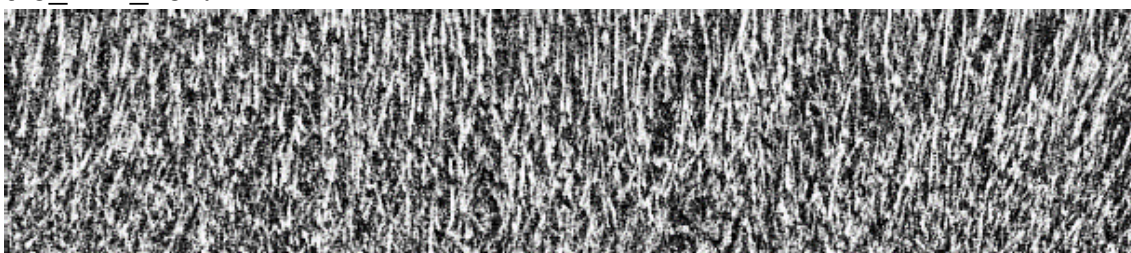


wt3_im1_z04:

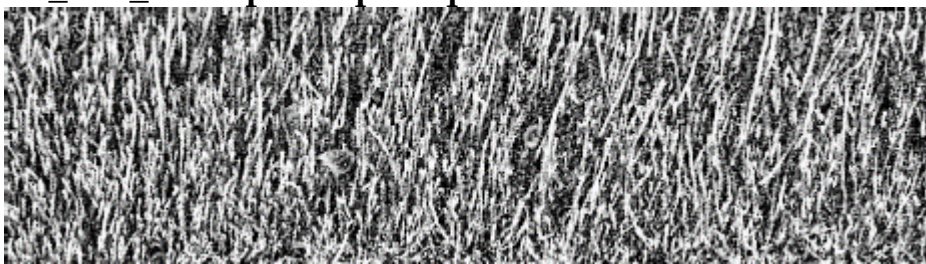


10. Anexo III: Imágenes finales

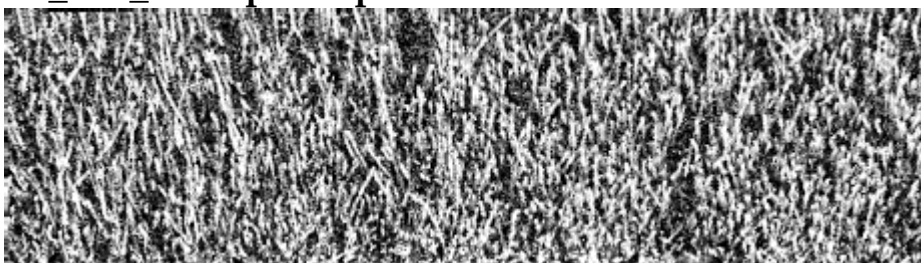
tr0_im1_z04:



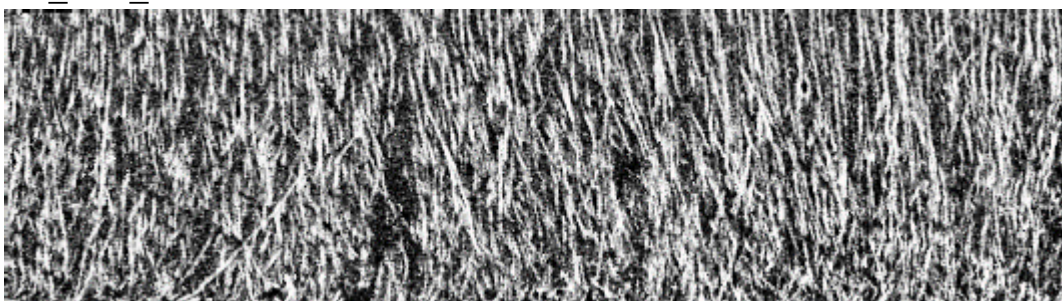
tr1_im2_z04 hipocampo izquierdo:



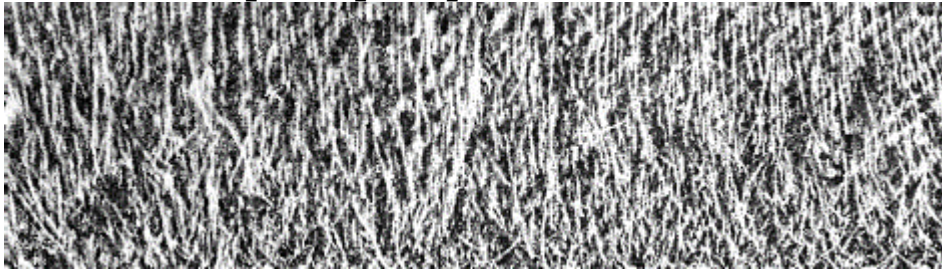
tr1_im2_z04 hipocampo derecho:



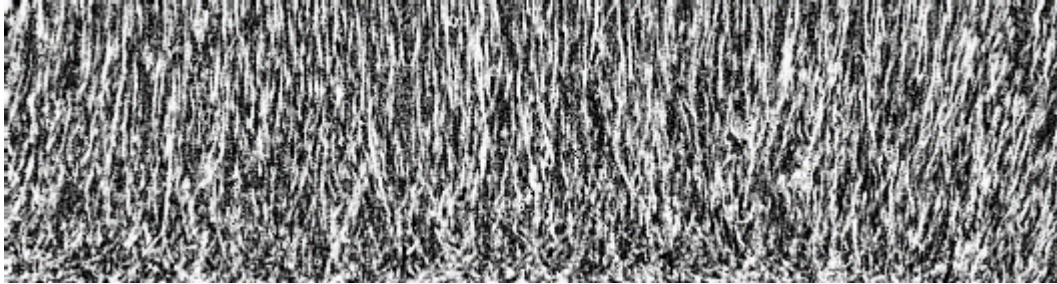
tr2_im1_z04:



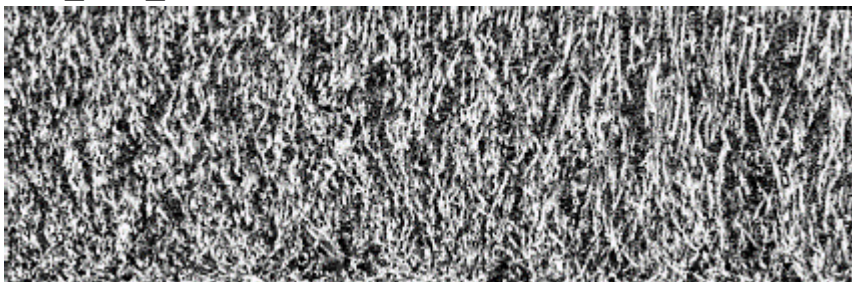
tr3_im2_z04 hipocampo izquierdo:



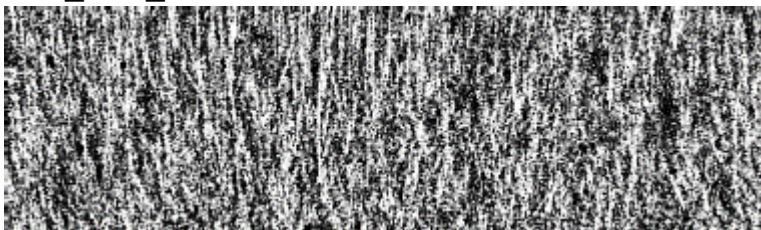
tr3_im2_z04 hipocampo derecho:



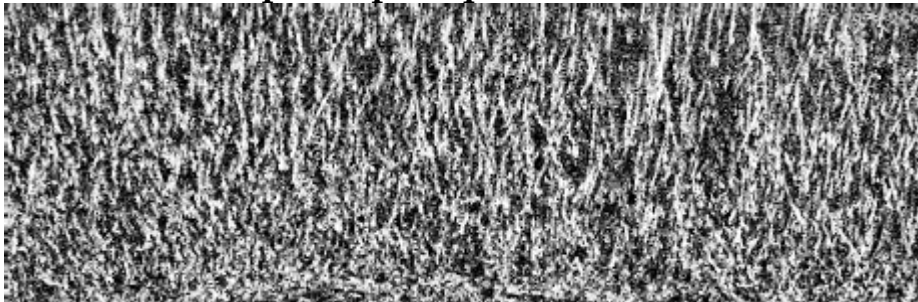
wt0_im1_z04:



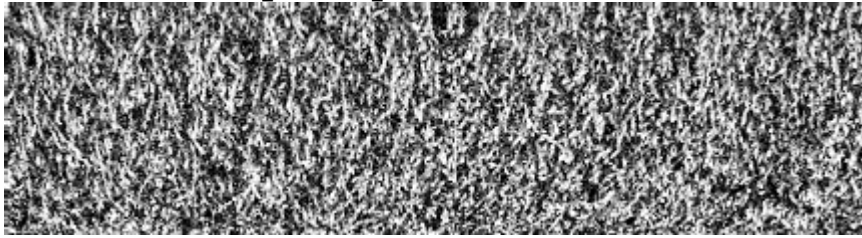
wt1_im1_z04:



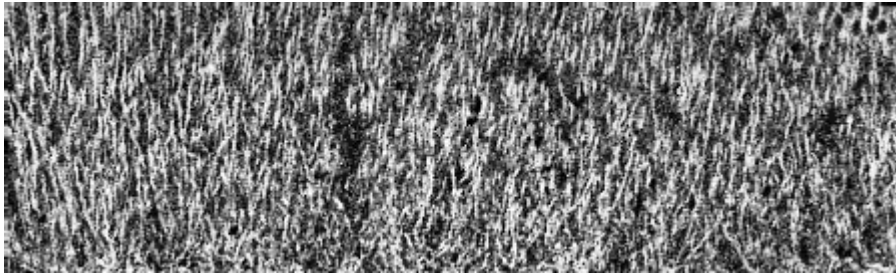
wt1_im2_z04 hipocampo izquierdo:



wt1_im2_z04 hipocampo derecho:



wt2_im2_z04:



wt3_im1_z04:

