



INTRODUCCIÓN A LOS MODELOS COMPUTACIONALES:

CUARTO CURSO DEL GRADO DE ING. INFORMÁTICA EN COMPUTACION

REDES NEURONALES CONVOLUCIONALES: EJEMPLOS

César Hervás-Martínez
Grupo de Investigación AYRNA

**Departamento de Informática y Análisis
Numérico**
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es

2020-2021



Ejemplo 1



Age and Gender classification using Convolutional Neural Networks



**Gil Lei and Tal Hassner. Age and Gender clasification using
Convolutional Neural Networks. IEEE Workshop on Analysis and
Modelling of Faces and Gestures (AMFG) (CVPR) Boston, June 2015**



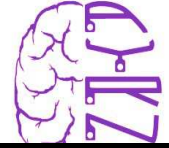
Visión de conjunto

Los autores utilizan CNN profundas para la tarea de clasificación automática de edad y género.

A pesar de la gran dificultad de las imágenes de la **base de datos Adience** y la simplicidad del diseño de red utilizado, el método mejora significativamente a otros métodos del estado del arte.



La base de datos de prueba Adience

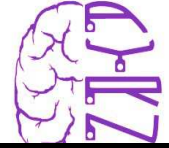


Ejemplo de imágenes de dos colecciones relevantes existentes en la colección Adience.

Izquierda: imágenes de referencia de PubFig. A pesar de estar consideradas "en la naturaleza", estas imágenes a menudo están limpias en términos de condiciones de visualización y muestran la participación de los sujetos fotografiados.



La base de datos de prueba Adience



Ejemplo de imágenes de dos colecciones relevantes existentes en la colección Adience..

Medio: La colección Gallagher, proporciona imágenes con un sesgo intencionado hacia grupos de personas, típicamente frente a la cámara y posando para sus tomas.



La base de datos de prueba Adience



Ejemplo de imágenes de dos colecciones relevantes existentes en la colección Adience. **Derecha:** Imágenes de la colección, cargadas automáticamente a Flickr, sin prefiltrado manual por parte de sus propietarios. En consecuencia, incluyen sujetos que miran hacia los lados, con desenfoque de movimiento, iluminación deficiente y más ruido, todo lo cual presenta desafíos adicionales para los sistemas de análisis de rostro automatizados.



La base de datos de prueba Adience



Consiste en imágenes cargadas automáticamente a Flickr desde teléfonos inteligentes.

Las condiciones de visualización de estas imágenes son muy restrictivas, por lo que se capturan variaciones extremas en la postura de la cabeza, en las condiciones de luminosidad, en el desenfoque, en la oclusión, en las expresiones, etc.

Incluye aproximadamente 26K imágenes de 2,284 sujetos.

Para las pruebas, se usa una versión de las caras alineadas en el plano.



Arquitectura de la red

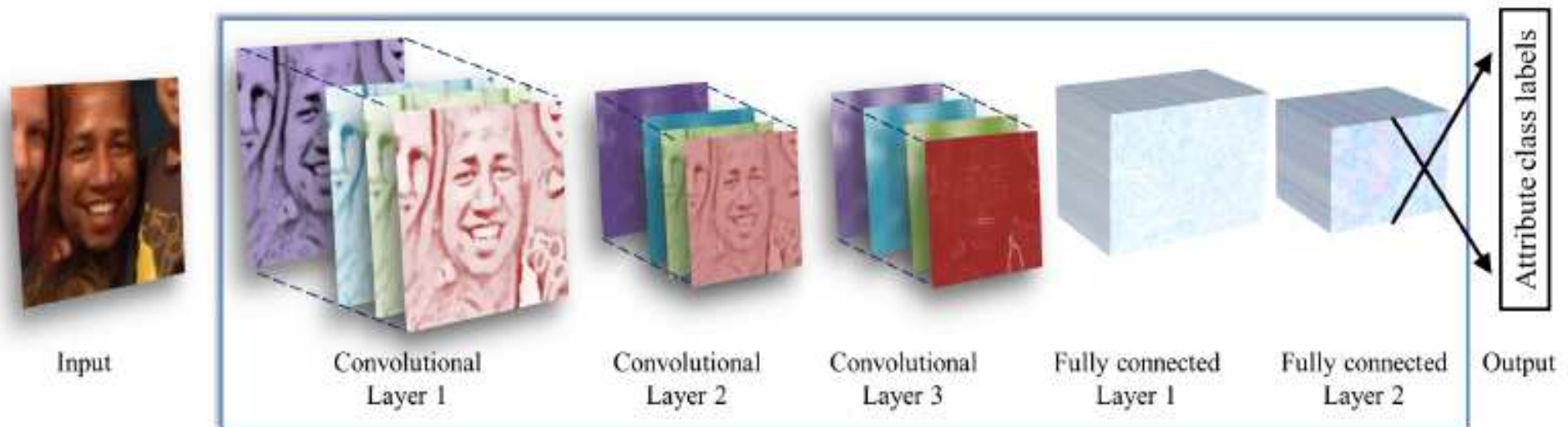
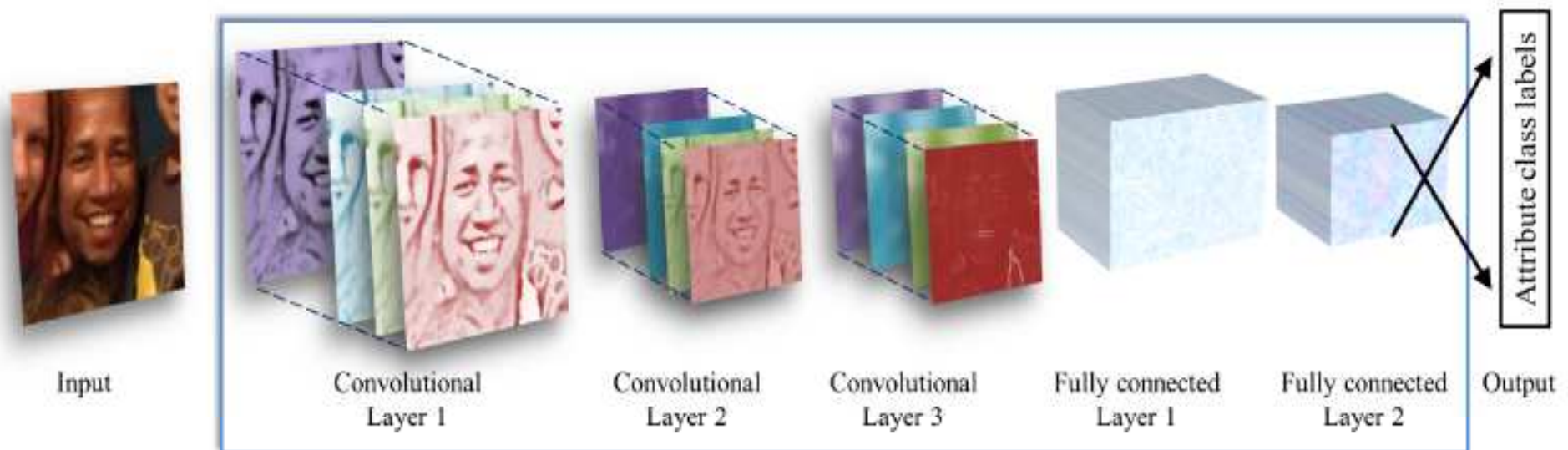


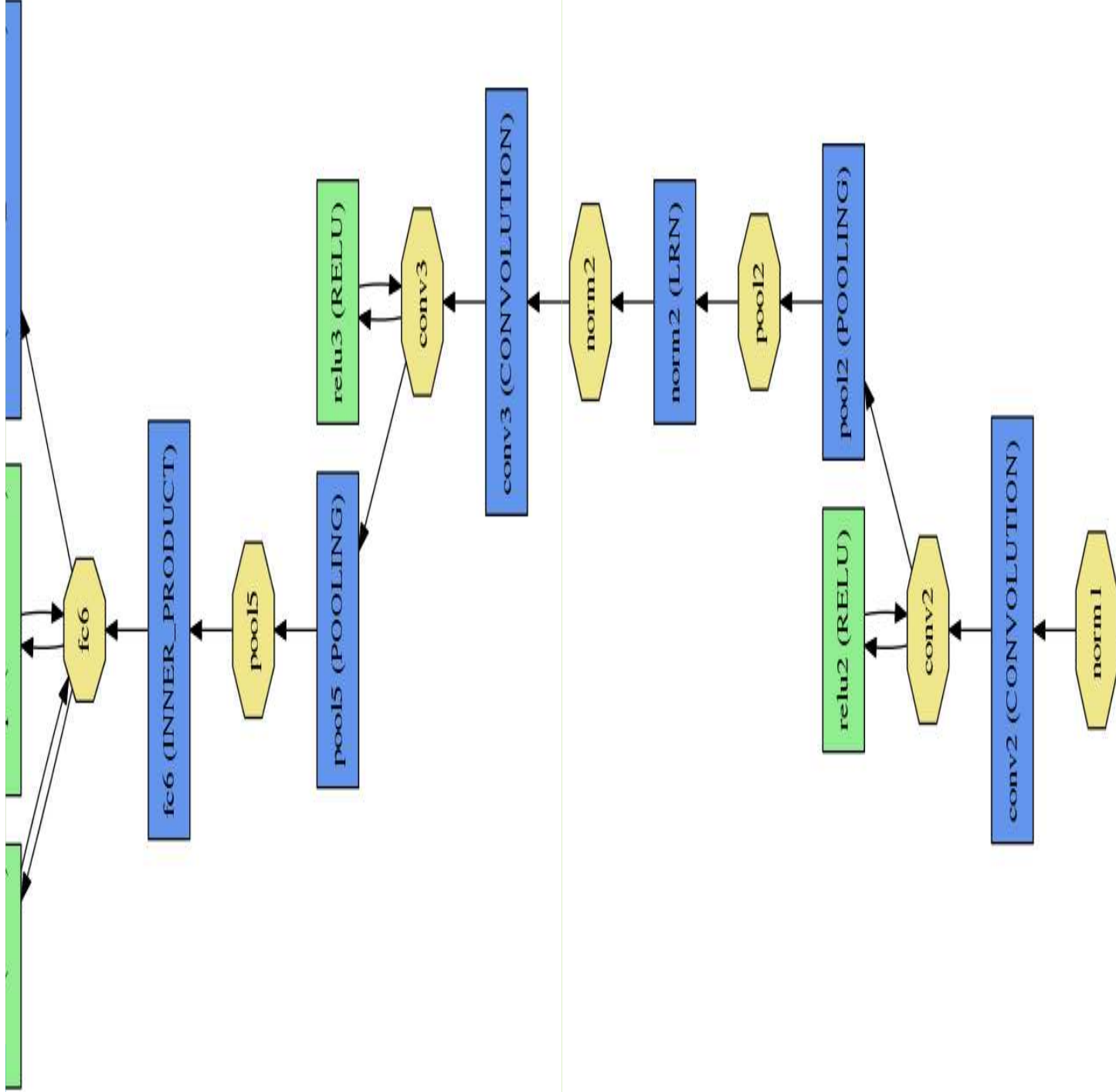
Ilustración de la arquitectura CNN. La red contiene tres capas convolucionales, cada una seguida de una capa de rectificado lineal ReLU y una capa de *pooling*. A las dos primeras capas le sigue también una normalización usando la normalización de respuesta local [28].



Arquitectura de la red



La primera capa convolucional contiene 96 filtros de 7 x 7 píxeles, la segunda capa convolucional contiene 256 filtros de 5 x 5 píxeles, la tercera y la última capa convolucional contienen 384 filtros de 3 x 3 píxeles. Finalmente, se agregan dos capas completamente conectadas, cada una con 512 neuronas. Ver la figura para un vista esquemática detallada.





Medidas para evitar el sobreentrenamiento



Una arquitectura de red simple con **tres capas convolucionales** y **dos capas completamente conectadas** considerando el tamaño de la base de datos y el número de etiquetas del problema a considerar (8 clases de edades y dos clases de género)

Aprendizaje por eliminación: Seleccionar aleatoriamente el valor de salida de las **neuronas de la capa completamente conectada o dense** de la red a 0 con una proporción de eliminación de 0,5 (50% de posibilidad) **Dropout**.

Disminución de los pesos. Utilizar un parámetro de disminución para llevar la magnitud de los pesos cerca de 0.

Aumento de datos: Tomar una muestra aleatoria de 227x227 a partir de una imagen de 256x256 y obtenerla de forma aleatoria en cada pasada de entrenamiento hacia adelante y hacia atrás.



Inicialización



Los pesos en todas las capas se inicializan con valores aleatorios con distribución Gaussiana de media cero y desviación estándar 0.01.

Para enfatizar esto, no se usan modelos pre-entrenados para inicializar la red; la red está entrenada, desde el inicio, sin usar datos fuera de las imágenes y las etiquetas disponibles en la base de datos de referencia.

Los valores de las etiquetas para el entrenamiento se representan como vectores binarios correspondientes a las clases originales.

Para cada imagen de entrenamiento, el objetivo es obtener la etiqueta del vector de salida de la red cuya longitud es el número de clases (dos para el género, ocho para las ocho clases de edad para la clasificación de la edad), con una codificación 1 de J



Predicción



Experimentamos con dos métodos de utilización de la red para producir predicciones de edad y género para caras nuevas:

Recorte central:

Alimentaremos la red con la imagen de la cara, recortada a 227 x 227 alrededor del centro de la cara.

Sobremuestreo:

Extraemos cinco muestras de regiones de 227 x 227 píxeles, cuatro de las esquinas de la cara de una imagen de 256 x 256 y una región de muestre adicional desde el centro de la cara.

La red se presenta con las cinco imágenes, junto con sus reflejos horizontales. La predicción final se considera el valor de predicción promedio a través de todas estas variaciones.



Experimentos



El método se implementa utilizando el marco de código abierto Caffe [26]. El entrenamiento se realizó en una máquina GPU de Amazon con 1,536 núcleos CUDA y 4GB de memoria de video. Toda la colección Adience incluye aproximadamente 26,000 imágenes de 2,284 sujetos.

La Tabla 1 enumera el desglose de la colección en las diferentes categorías de edad. Las pruebas para la clasificación por edad o sexo se realizan **utilizando un protocolo estándar de validación cruzada de cinco *folds*, exclusivo para cada sujeto**, definido en [10]. Utilizamos la versión alineada en el plano de las caras, originalmente utilizada en [10].

	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-	Total
Hombre	745	928	934	734	2308	1294	392	442	8192
Mujer	682	1234	1360	919	2589	1056	433	427	9411
Ambos	1427	2162	2294	1653	4897	2350	825	869	19487

Tabla 1. Desglose de la base de datos AdienceFaces en las diferentes clases de edad y género.



Experimentos



Scrip

5-fold cross-validation
basado en la distribución de
los folds exclusivos del
sujeto pre-especificado

Lo que usaron otros autores

Entrenaron una máquina
GPU Amazon con 1,536
CUDA cores y 4 GB GDDR5
RAM

Lo que usaron estos autores

Entrenamiento en un Nvidia
Quadro k2200 con 640 CUDA
cores y 4 GB GDDR5 RAM

net: "trainer.prototxt"

test_iter: 1000

test_interval: 1000

base_lr: 0.001

lr_policy: "step"

gamma: 0.1

stepsize: 10000

display: 20

max_iter: 50000

momentum: 0.9

weight_decay: 0.0005

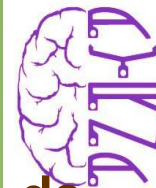
snapshot: 1000

snapshot_prefix: "snaps/age_train"

solver_mode: GPU



Resultados



Las Tablas 2 y 3 presentan los resultados para la clasificación de género y edad, respectivamente

Exacto: Debe de pertenecer el patrón en test a la clase exacta de edad.

1-off: Debe de pertenecer el patrón a la clase exacta o a una contigua.

Método	Accuracy	Método	Exacto	1-off
Lo mejor de [10]	77.8 \pm 1.3	Lo mejor de [10]	45.1 \pm 2.6	79.5 \pm 1.4
Lo mejor de [23]	79.3 \pm 0.0	Propuesto usando un solo cultivo	49.5 \pm 4.4	84.6 \pm 1.7
Propuesto usando un solo cultivo	85.9 \pm 1.4	Propuesto utilizando sobremuestreo	50.7 \pm 5.1	84.7\pm2.2
Propuesto utilizando sobremuestreo	86.8\pm1.4			

Tabla 2. Resultados de la **estimación de género** en la base de datos Adience. Se enumeran la media de precisión \pm las desviaciones típicas en todas las categorías de edad.

Tabla 3. Resultados de la **estimación de edad** en la base de datos Adience. Se enumeran la media de precisión \pm las desviaciones típicas en todas las categorías de edad..

[illegible][illegible]



Resultados



Figura 4. Clasificaciones erróneas de género.

Fila superior: sujetos femeninos clasificados erróneamente como hombres.

Fila inferior: sujetos masculinos clasificados erróneamente como mujeres

Ejemplos de clasificaciones erróneas de género y edad se muestran en las Figuras 4 y 5, respectivamente. Estos muestran que muchos de los errores cometidos por el sistema se deben a las condiciones de visualización extremadamente difíciles de algunas de las imágenes de referencia de Adience.

Los errores de estimación de género también ocurren con frecuencia para imágenes de bebés o niños muy pequeños donde los atributos de género obvios aún no son visibles.



Resultados



Figura 5. Clasificaciones erróneas de edad.

Fila superior: sujetos mayores clasificados erróneamente como más jóvenes.

Fila inferior: sujetos más jóvenes clasificados erróneamente como mayores.

Lo más notable, son los errores causados por el desenfoque o la baja resolución y las oclusiones (particularmente por un maquillaje excesivo).



Referencias

- [10] E. Eiding, R. Enbar, and T. Hassner. Age and gender estimation of unfiltered faces. *Trans. on Inform. Forensics and Security*, 9(12), 2014

- [23] T. Hassner, S. Harel, E. Paz, and R. Enbar. Effective face frontalization in unconstrained images. *Proc. Conf. Comput. Vision Pattern Recognition*, 2015.

- [28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Inform. Process. Syst.*, pages 1097–1105, 2012.

.



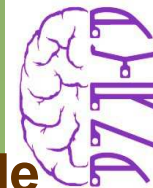
Ejemplo 2

Fast Convolutional Neural Network Training Using Selective Data Sampling: Application to Hemorrhage Detection in Color Fundus Images

M. J. J. P. van Grinsven, B. van Ginneken, C. B. Hoyng, T. Theelen, and C. I. Sánchez. Fast Convolutional Neural Network Training Using Selective Data Sampling: Application to Hemorrhage Detection in Color Fundus Images. IEEE Transactions on Medical Imaging, vol. 35, no. 5, may 2016, 1273-1284



Base de datos de Kaggle:



El conjunto de datos de Kaggle consta de 35,126 imágenes de entrenamiento clasificadas en cinco etapas de Retinopatía Diabética, DRP, y 53,576 imágenes de prueba con etapa DRP no revelada.

Se seleccionó un subconjunto formado por 6.679 imágenes del conjunto de entrenamiento Kaggle. Este subconjunto consta de:
4.450 imágenes seleccionadas al azar de DRP etapa 0 (**normal**),
488 imágenes seleccionadas al azar de DRP etapa 1 (**leve**),
1.058 imágenes seleccionadas al azar de DRP etapa 2 (**moderada**)
593 imágenes seleccionadas al azar de DRP etapa 3 (**severa**).

Las 6.679 imágenes seleccionadas se dividieron en un conjunto de entrenamiento, validación y test de acuerdo con una división 60-20-20. Las imágenes del mismo paciente se mantuvieron en el mismo subconjunto.



Base de datos Messidor



La base de datos Messidor consta de 1200 imágenes adquiridas en tres sitios diferentes.

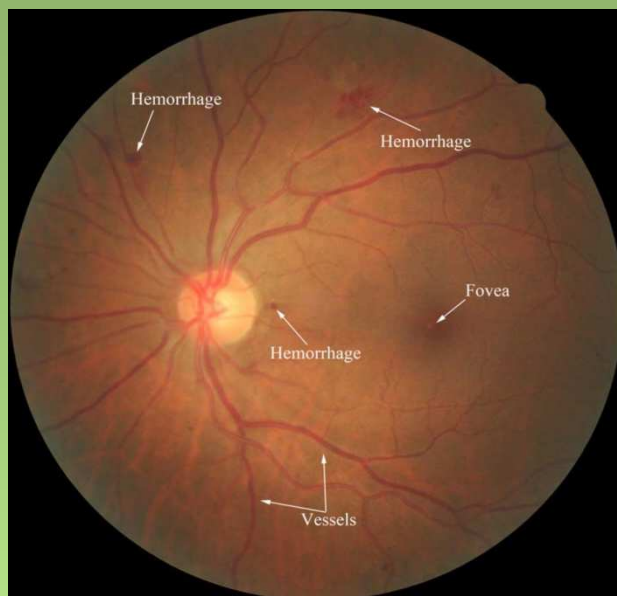
Las imágenes se obtuvieron utilizando una cámara de video en color 3CCD en un retinograma no midriático Topcon TRC NW6 con un campo de visión de 45 grados.

Las imágenes tienen resoluciones de 1440 x 960, 2240 x 1488 o 2304 x 1536 píxeles.

El conjunto Messidor se utilizará exclusivamente como un conjunto independiente para test.

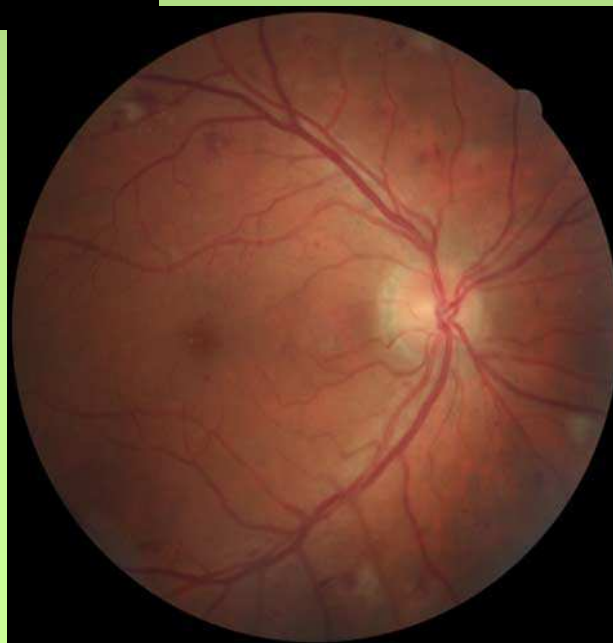


Fast Convolutional Neural Network Training Using Selective Data Sampling:

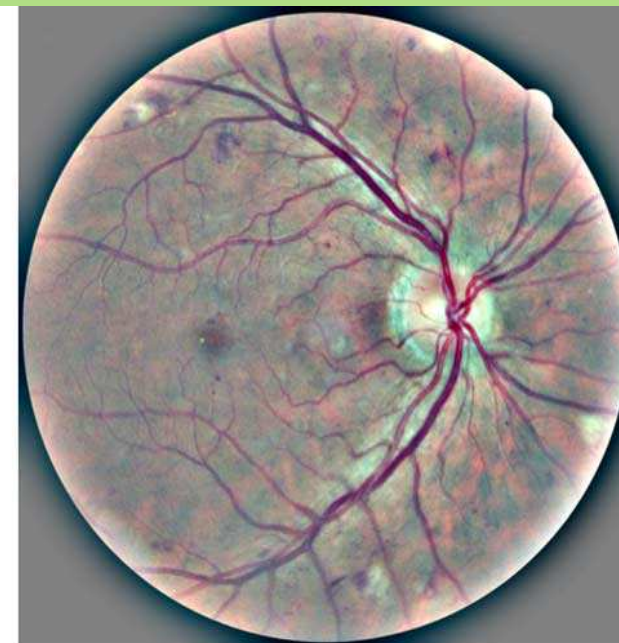


Ejemplo de una imagen de fondo de color que muestra la presencia de hemorragias

Etapas de preprocesamiento de mejora de contraste.
(a) Imagen del fondo de color original. (b) Imagen mejorada de contraste.



(a)



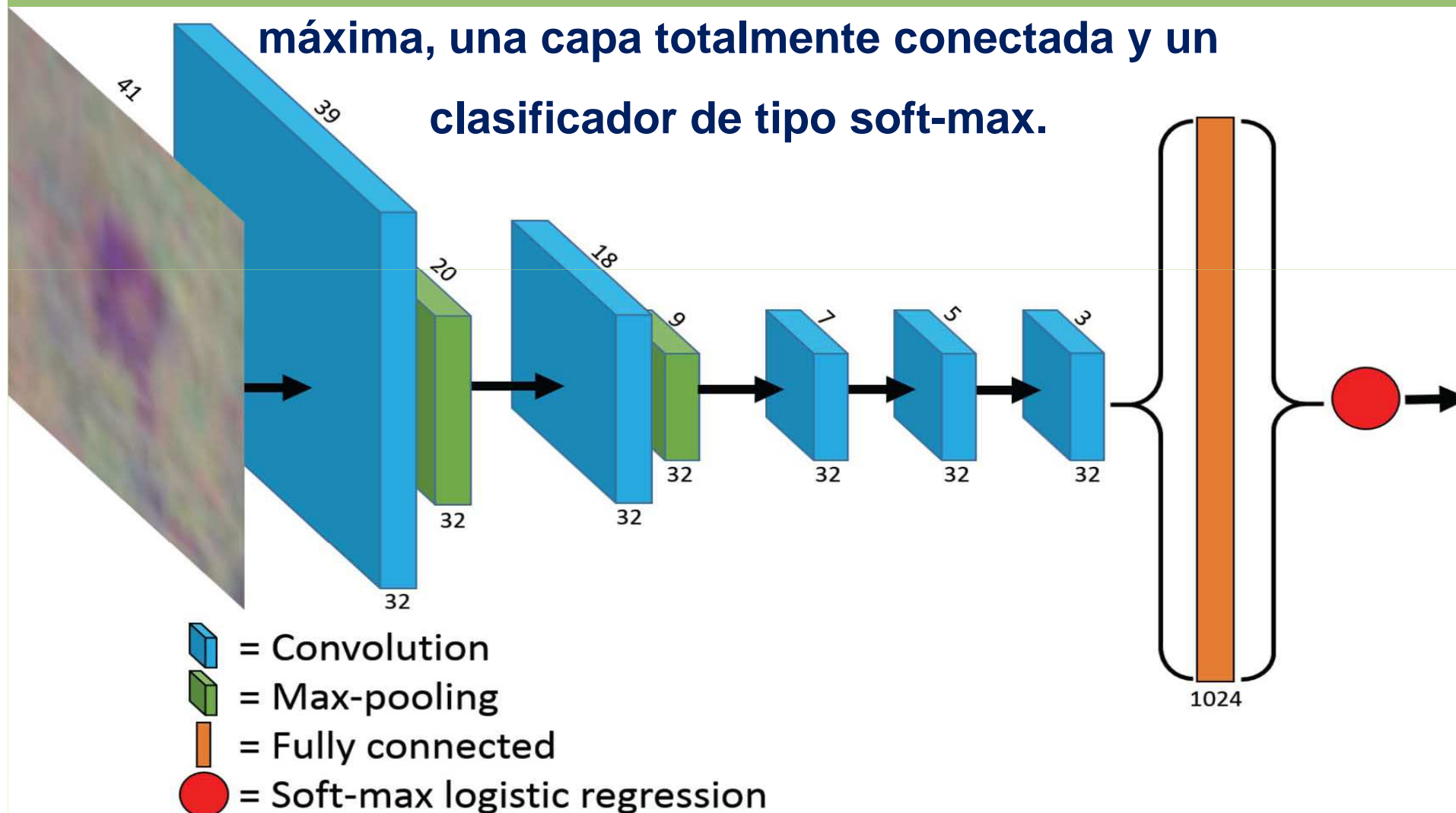
(b)



Fast Convolutional Neural Network Training Using Selective Data Sampling:

Descripción esquemática de la arquitectura CNN que
contiene capas convolucionales, capas de agrupación

máxima, una capa totalmente conectada y un
clasificador de tipo soft-max.





Fast Convolutional Neural Network Training Using Selective Data Sampling:



La arquitectura CNN utilizada en este estudio consta de cinco capas convolucionales seguidas de **Unidades Rectificadas Lineales (ReLU)** [1] y capas de pooling máximo.

Las capas finales de la red consisten en una capa completamente conectada y una capa final de clasificación softmax.

Este diseño está inspirado por el **OxfordNet** [2] que mostró un buen rendimiento para la clasificación de imágenes de escenas naturales, los autores **utilizan 32 filtros de tamaño pequeño de 3x3 píxeles en cada capa convolucional.**

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv:14091556*, 2014.



Fast Convolutional Neural Network Training Using Selective Data Sampling:



El *poolong* máximo es de tamaño 2×2 y se aplica un *stride* de 2 después de las dos primeras capas convolucionales, reduciendo a la mitad el tamaño del mapa de características después de estas operaciones.

El *poolong* máximo reduce el número de parámetros libres e introduce una pequeña invarianza espacial en la red [3].

La capa totalmente conectada consta de 1024 nodos seguidos de una regresión logística softmax que genera un valor que oscila entre 0 y 1, lo que indica la probabilidad de que el píxel pertenezca a la clase positiva.



Preprocesamiento



En un paso de *preprocesamiento*, el campo de visión de las imágenes del fondo de color se segmenta para limitar el análisis de la CNN a la región de interés.

La coincidencia con la plantilla circular se utiliza para extraer el campo de visión y las imágenes se recortan en el cuadrado delimitador de este campo de visión circular [26].

Las imágenes se redimensionan a **512 x 512** para reducir los costos computacionales y se aplica el *preprocesamiento* para mejorar el contraste de la imagen [27], [28].



Preprocesamiento



Se obtiene una imagen mejorada de contraste

$I_{ce}(x, y; \sigma)$ de la siguiente manera [29]:

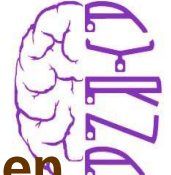
$$I_{ce}(x, y; \sigma) = \alpha I(x, y) + \beta G(x, y; \sigma) * I(x, y) + \gamma$$

donde * representa el operador de convolución y $G(x, y; \sigma)$ un filtro gaussiano con escala σ . Los valores de los parámetros fueron elegidos empíricamente como:

$$\alpha = 4, \beta = -4, \sigma = 512 / 30, \gamma = 128$$



Preprocesamiento



Para construir los datos de entrenamiento de la CNN, se extraen píxeles negativos solo de imágenes negativas y píxeles positivos solo de imágenes positivas en ubicaciones de hemorragia correspondiente a parches de entrenamiento, centrados en los píxeles extraídos, se crean parches de tamaño 41 x 41 y 3 canales de profundidad durante la rutina de entrenamiento de las CNN.

La etiqueta del parche está determinada por la etiqueta del píxel central.

El aumento de datos se hace por traslación espacial de un píxel en dirección horizontal y vertical y el volteo vertical y horizontal se aplica a los parches positivos (glaucoma) para aumentar artificialmente el número de positivos.

Los parches negativos (no glaucoma) también se voltean al azar vertical y horizontalmente para contrarrestar un posible sobreajuste.



Fast Convolutional Neural Network Training Using Selective Data Sampling:

La regularización o degradación de los pesos se agrega a cada capa para penalizar los parámetros de gran peso durante la retro-propagación del gradiente en la rutina de optimización.

La Tabla II y la Fig. 3 muestran una visión general de la arquitectura de red con la omisión de las funciones ReLU.



Fast Convolutional Neural Network Training Using Selective Data Sampling:



Todos los parámetros de red se **inician aleatoriamente** según una distribución normal con una varianza igual a 0.05.

La CNN está entrenada usando un descenso de gradiente estocástico con una tasa de aprendizaje de 5×10^{-5} , minimizando una función de costo **C** definida como la entropía de una distribución binaria:

$$C(l, s) = - \sum_{i=0}^B l_i \log(s_i) + (1 - l_i) \log(1 - s_i)$$

donde **s** es la probabilidad asignada al píxel, **l** es la etiqueta del píxel de referencia y **B** el número total de muestras en un mini-lote.



Fast Convolutional Neural Network Training Using Selective Data Sampling:



Se utiliza un tamaño de mini-lote de 256 parches y una época se define como 4000 mini-lotes.

Esto significa que contamos con 1.024.000 muestras, de las cuales la mitad son positivas y la otra mitad negativas, que se utilizan en cada época para entrenar a la red CNN.



Fast Convolutional Neural Network Training Using Selective Data Sampling:



TABLA II.- Arquitectura de la red CNN. Para cada capa convolucional, se muestra el *ancho x alto x profundo* del kernel junto con el número de kernels K. En cada capa max-pooling, se aplica un 2 x 2 max-pooling con paso de *a pixeles*

Layer	Operation	Input size	Details
Layer 1	convolution	41 x 41	3x3x3, K=32
Layer 2	max-pooling	39 x 39	2x2, a=2
Layer 3	convolution	20 x 20	3x3x3, K=32
Layer 4	max-pooling	18 x 18	2x2, a=2
Layer 5	convolution	9 x 9	3x3x3,
Layer 6	convolution	7 x 7	3x3x3,
Layer 7	convolution	5 x 5	3x3x3,
Layer 8	Fully connected	3 x 3	1024 nodes
Layer 9	Soft-max	1024 x 1	2 classes



Muestreo selectivo



En cada época de entrenamiento de la red CNN, **se asigna un peso a cada muestra negativa**, proporcional a su probabilidad de salir en el muestreo: mayor peso significa una mayor probabilidad de ser seleccionado para la próxima época.

Para reducir el número de muestras redundantes (**pixeles negativos**) en el conjunto de entrenamiento, **se asignan pesos más altos a las muestras mas representativas**.

Las muestras representativas se consideran aquellas muestras negativas con un error de clasificación mayor en el estado actual de entrenamiento de la CNN.



Algoritmo iterativo



El algoritmo iterativo propuesto para seleccionar dinámicamente los píxeles de entrenamiento para entrenar una red c CNN siguen estos pasos:

1) Inicializar los conjuntos de píxeles positivos $X_+^t \subset X_+$ y negativos $X_-^t \subset X_-$ seleccionando aleatoriamente **M** muestras con reemplazamiento para cada clase X_+ y de X_- , respectivamente.

2) Entrenar la red c con $X^t = X_+^t \cup X_-^t$ usando del algoritmo del gradiente descendente.

3) Clasificar cada píxel x_i con la red entrenada c^t .

Se obtiene un valor de probabilidad de cada pixel x_i en X_- después de la clasificación

4) Se asigna a cada x_i en X_- un peso $w_i^t = |s_i^t - l_i|$

Se asigna un peso mayor a aquellos píxeles de los cuales la predicción preliminar de red difiere más de la etiqueta de referencia inicial.



Algoritmo iterativo



5) Actualizar X_+^t y X_-^t seleccionando M muestras para cada clase. Se selecciona aleatoriamente x_i en X_+^t mientras que x_i en X_-^t se selecciona con probabilidad p_i^t [32] - [34]:

$$p_i^t = \frac{w_i^t}{\sum_{x_j \in X} w_j^t} \quad w_i^t = |s_i^t - l_i|$$

6) Entrenar la red c con $X^t = X_+^t \cup X_-^t$ utilizando el algoritmo estocástico de gradiente descendente.

7) Repetir los pasos (3) a (6) hasta alcanzar el criterio de parada..



Algoritmo iterativo



En este algoritmo de muestreo selectivo iterativo (SeS) propuesto, el conjunto de píxeles de entrenamiento negativos y positivos se cambia dinámicamente en cada época de entrenamiento, evitando que el proceso de entrenamiento se centre en muestras negativas redundantes de forma tal que entrenemos eficientemente a la red CNN.

El parámetro M no se puede ajustar por sí mismo, sino que depende del tamaño del mini lote y del número de mini lotes en una época. Se puede cambiar el valor de M modificando cualquiera de los dos.

Para obtener un esquema más eficiente, los autores consideran aplicar los pasos (3) y (4) una vez cada cinco épocas.



Medida de rendimiento

Los valores de AUC basados en píxeles son engañosos debido a la distribución desigual de píxeles positivos y negativos. Por lo tanto, medimos el valor de AUC según el rendimiento de la clasificación de las imágenes.

Se obtiene una puntuación para cada imagen clasificando todos los píxeles de la imagen y considerando la máxima probabilidad de los píxeles como la puntuación de la imagen.

Cuando el valor de AUC en el conjunto de validación alcanza un máximo estable determinado después de la inspección visual, la fase de entrenamiento CNN se considera terminada.



Identificación de la lesión hemorrágica



El mapa de probabilidad de píxeles obtenido se convoluciona con un filtro gaussiano **con escala** para suavizar los valores.

Las regiones candidatas de hemorragia se identifican detectando máximos locales en el mapa suavizado de probabilidad de píxeles.

Las ubicaciones máximas locales se utilizan como puntos de partida de la programación dinámica para segmentar a los **candidatos individuales** objeto de tener una hemorragia.



Identificación de imágenes con hemorragias



Para determinar si una imagen contiene hemorragias, se calcula una puntuación de imagen a partir del mapa de probabilidad de píxeles obtenido.

Después de aplicar el paso de suavizado gaussiano al mapa de probabilidad de píxeles, la probabilidad máxima de píxeles se asigna como puntuación de la imagen.

Se utilizó el análisis ROC de respuesta libre (FROC) para comparar el rendimiento de las CNN para la detección de hemorragias individuales [35].

Aquí, solo se tuvieron en cuenta los falsos positivos encontrados en las imágenes negativas para evitar las ambigüedades en las anotaciones de referencia las cuales pueden influir en el resultado [37], [38].



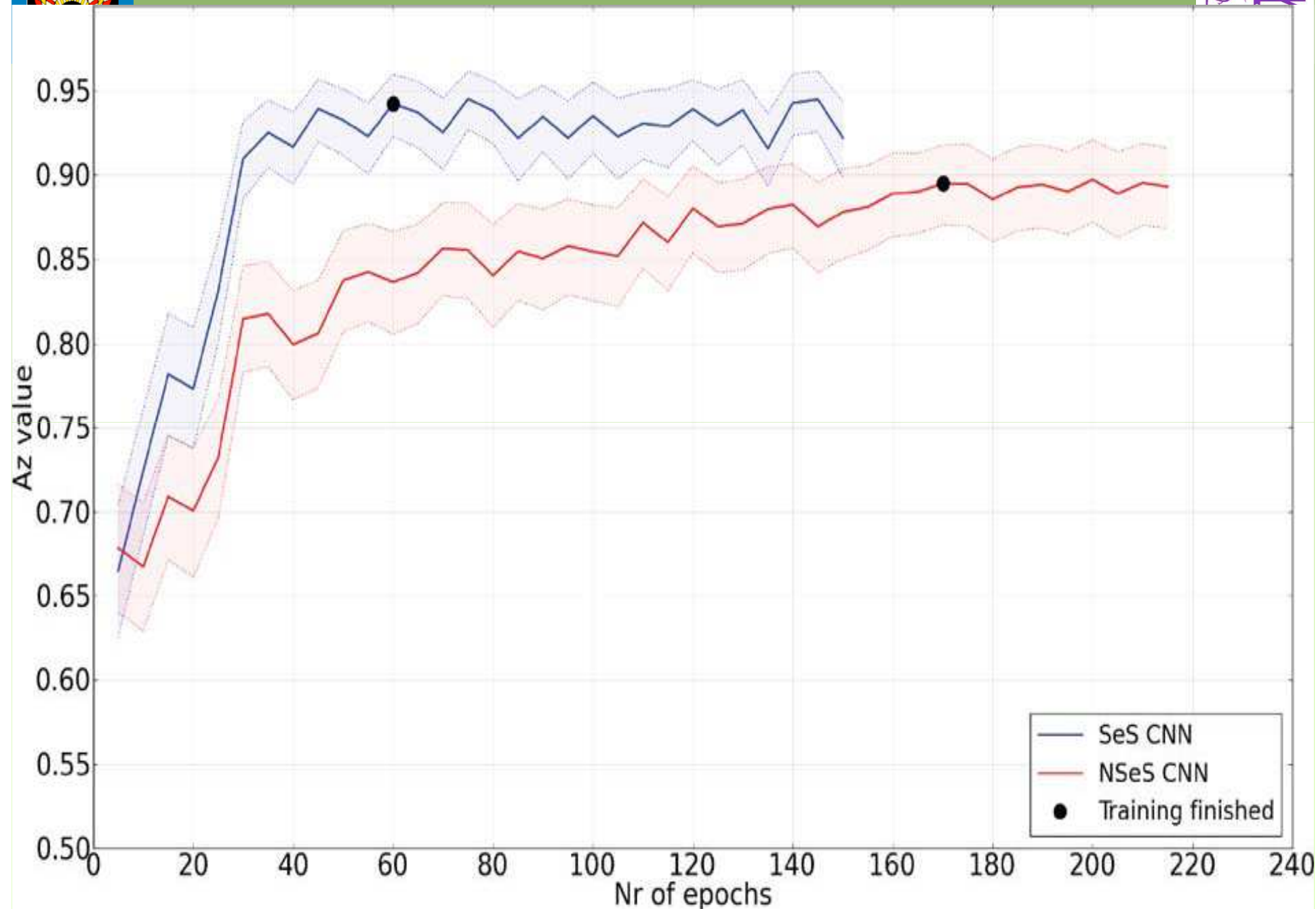
Hemorrhage Detection in Color Fundus Images



Valores de AUC basados en imágenes en el conjunto de validación para modelos SeS CNN y NSeS CNN durante un número de épocas de entrenamiento. Las regiones sombreadas indican los intervalos de confianza del 95% de los valores de AUC. Después de 60 y 170 épocas, las fases de entrenamiento de SeS CNN y NSeS CNN se consideraron terminadas.



Hemorrhage Detection in Color Fundus Images

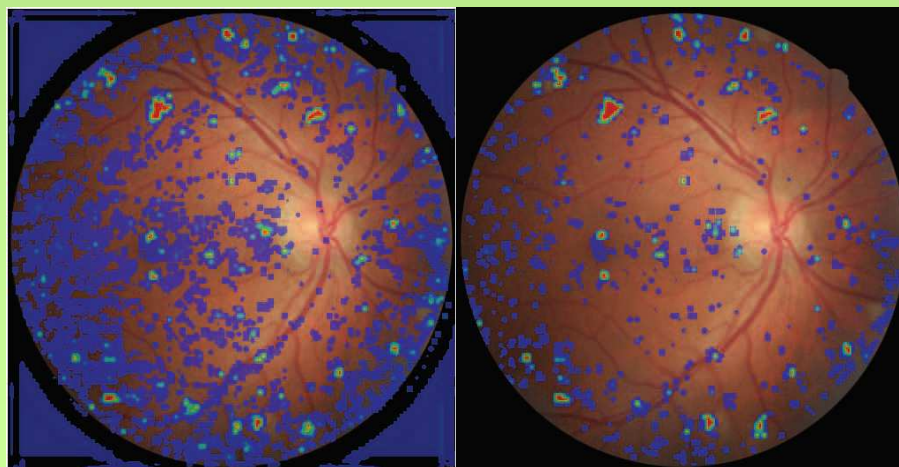




Resultados

Mapas de probabilidad de píxeles obtenidos aplicando los modelos CNN SeS y NSeS a una imagen de muestra del conjunto de entrenamiento después de entrenar la red para un número diferente de épocas.

Las superposiciones se muestran utilizando un código de mapa de calor, donde los colores rojos denotan altas probabilidades y azules bajas probabilidades

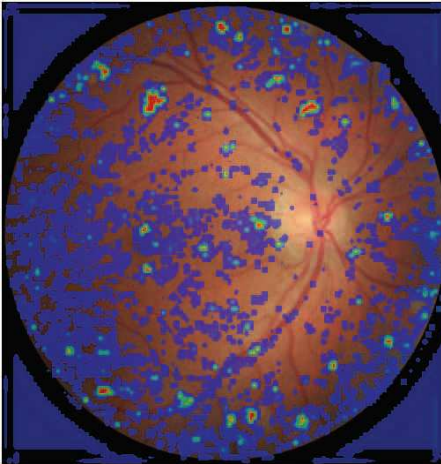




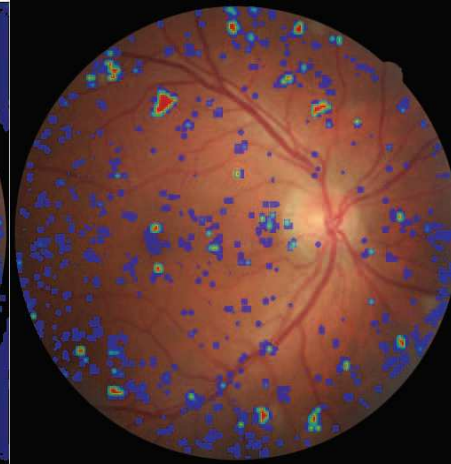
Resultados



SeS CNN



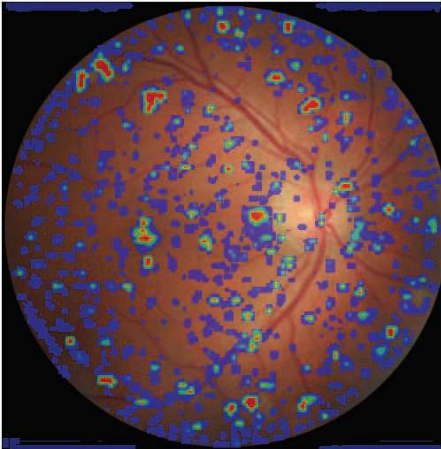
40 épocas



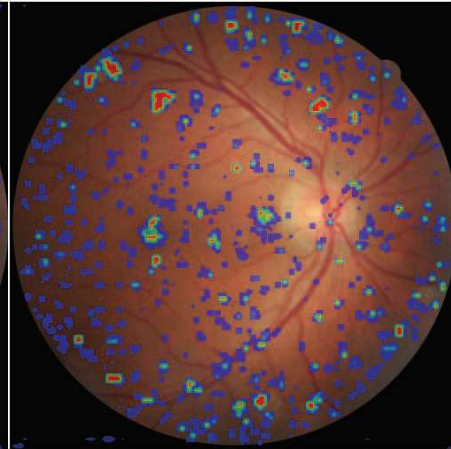
60 épocas

El modelo CNN SeS **necesitó 60 épocas** para alcanzar el rendimiento final, mientras que el modelo CNN NSSS **necesitó 170 épocas** para alcanzar el rendimiento final.

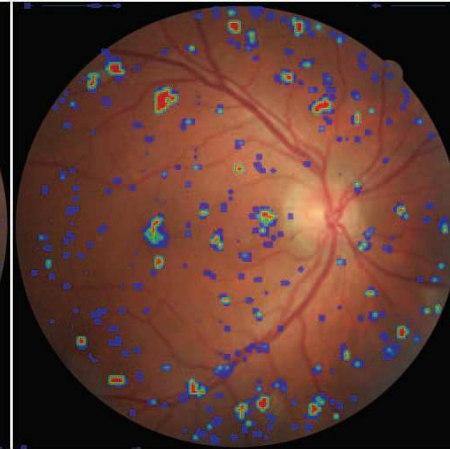
NSeS CNN



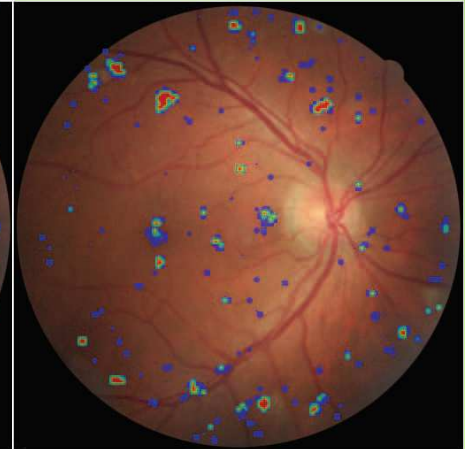
40 épocas



60 épocas



150 épocas



170 épocas



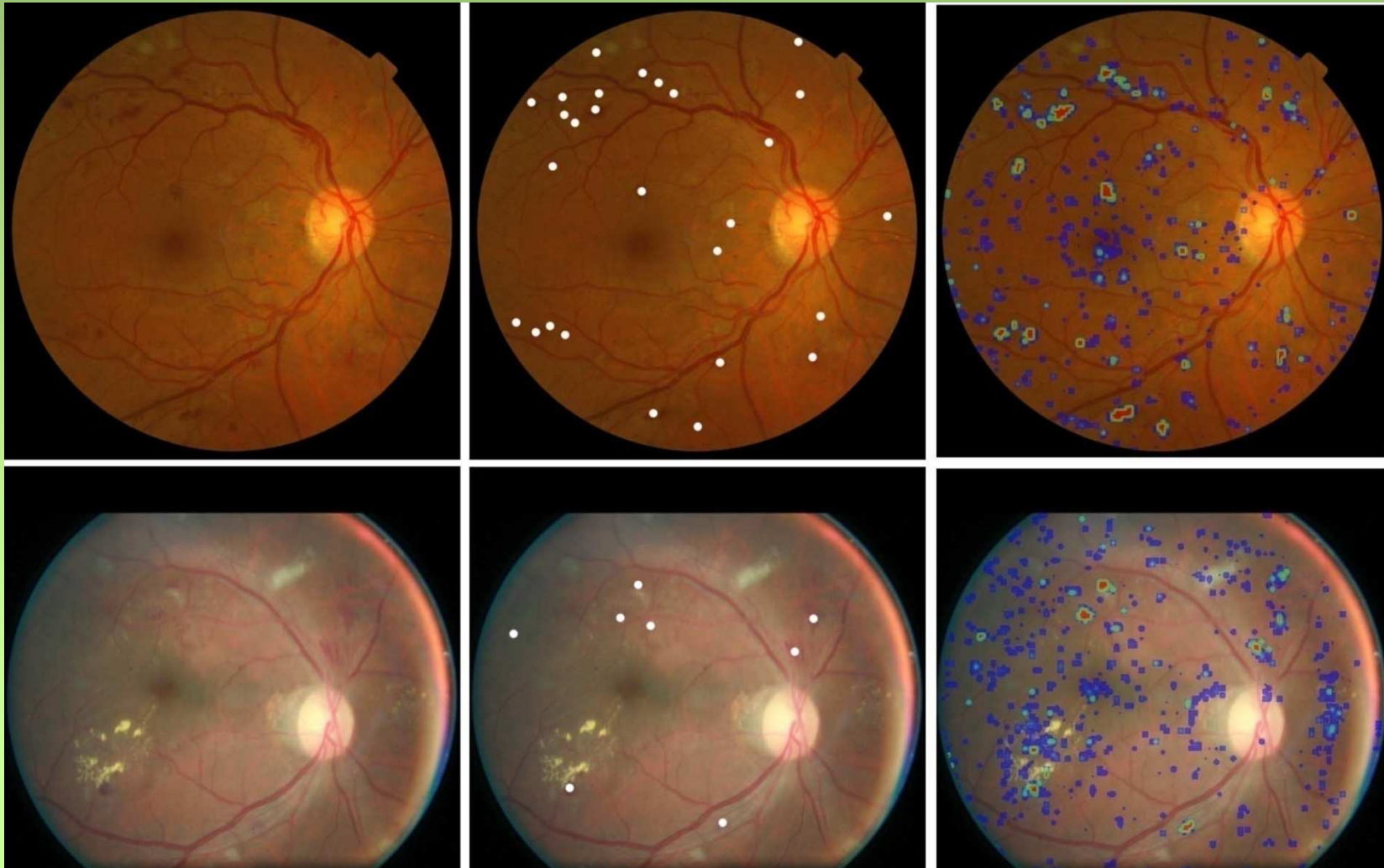
Hemorrhage Detection in Color Fundus Images



Columna izquierda: ejemplo de imágenes de fondos de color del conjunto de pruebas Kaggle.

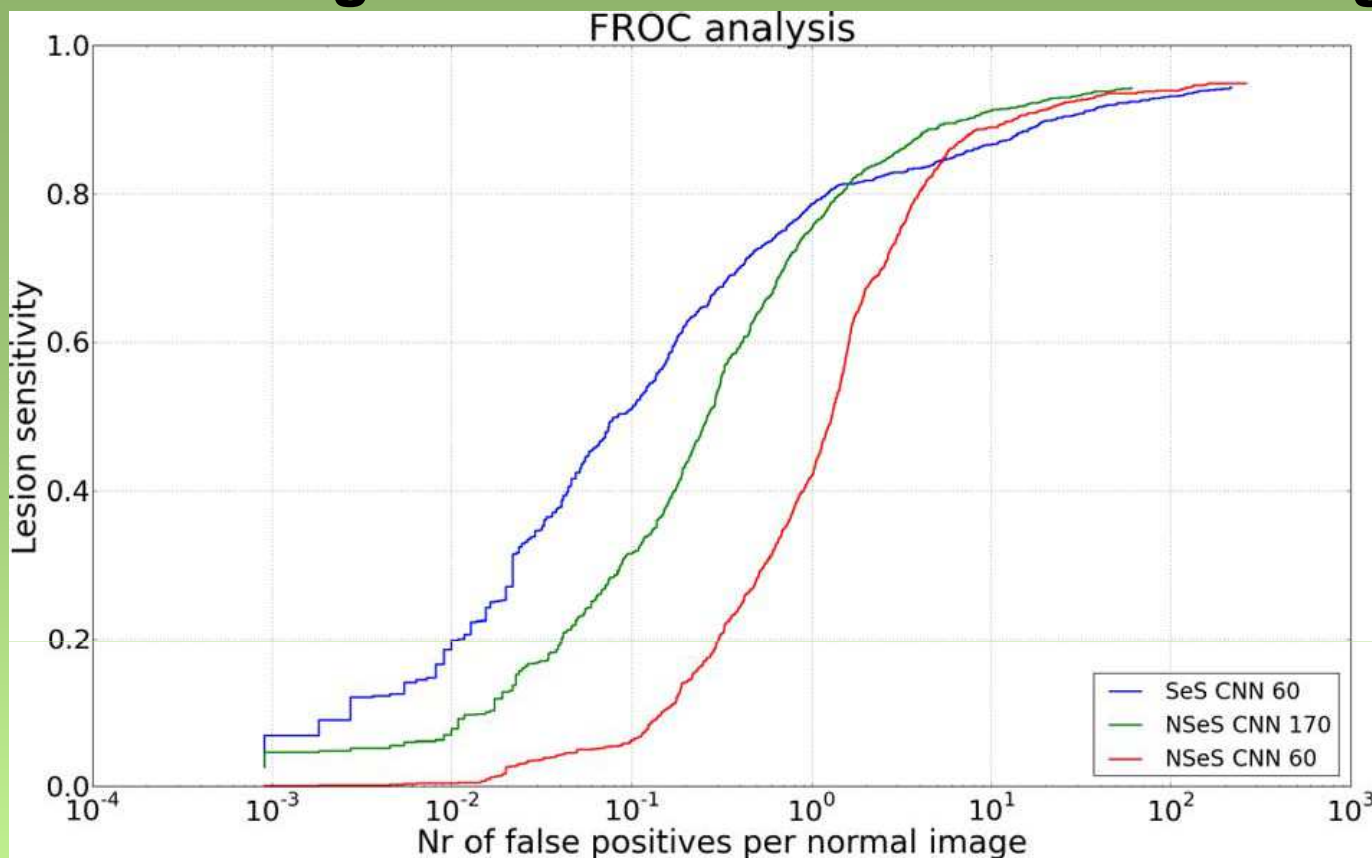
Columna media: ubicaciones del centro de hemorragia de referencia.

Columna derecha: salida del modelo SeS CNN 60 épocas.





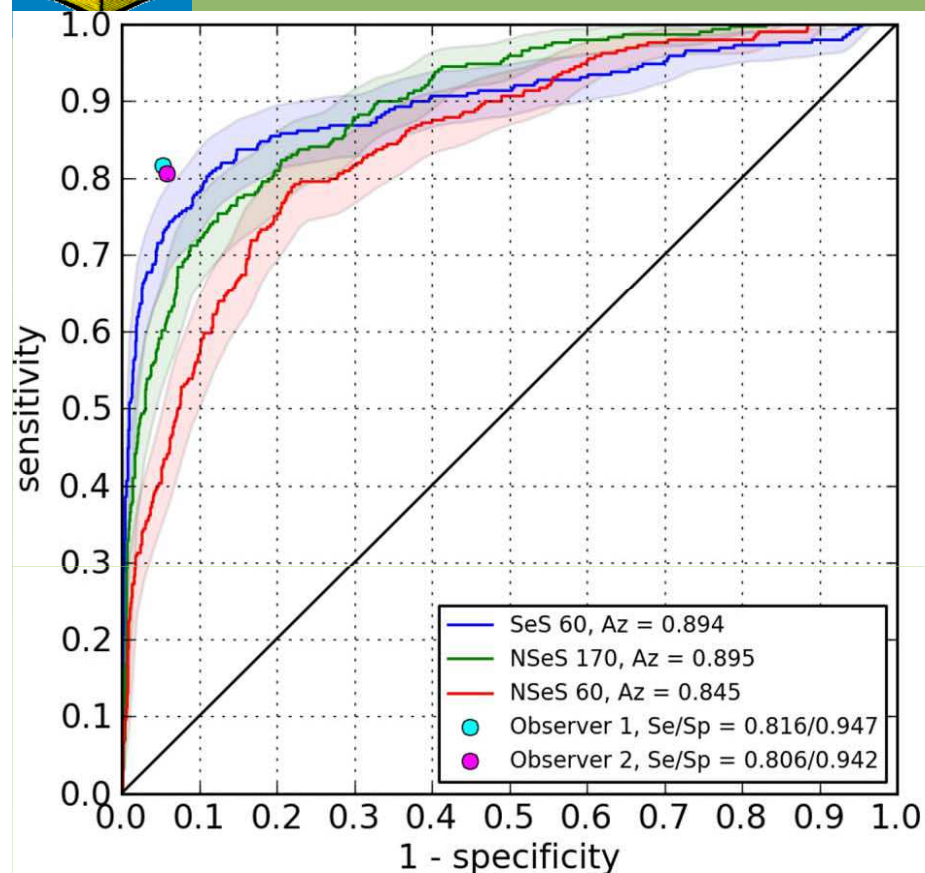
Hemorrhage Detection in Color Fundus Images



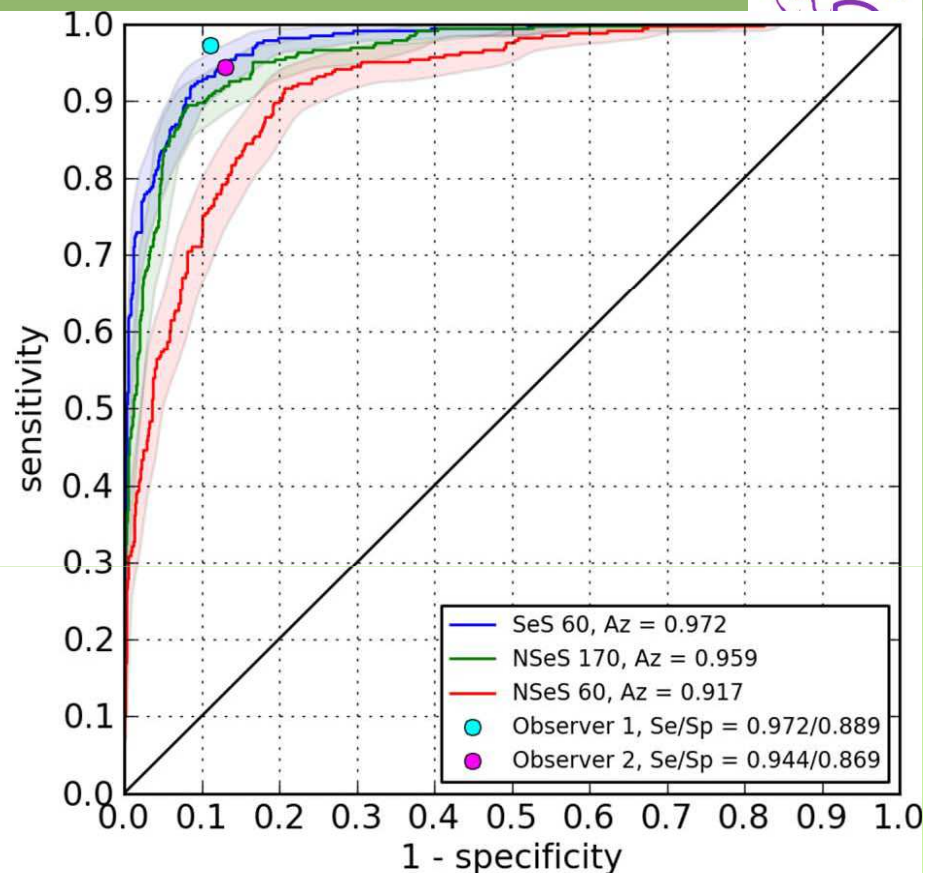
Curvas FROC de los modelos SeS CNN 60 épocas y NSeS CNN 170 épocas en el conjunto de prueba Kaggle. La curva FROC de NSeS CNN después de 60 épocas de entrenamiento se agrega para la comparación directa con SeS CNN después de 60 épocas de entrenamiento.



Resultados



(a)



(b)

Fig. . Curvas ROC basadas en imágenes en los conjuntos de prueba Kaggle (a) y Messidor (b). Los puntos asociados a los observadores humanos se agregan en los gráficos. (a) Kaggle (b) Messidor.



Resultados



TABLE III

CONTINGENCY HUMAN OBSERVERS, SeS CNN 60 AND NSeS CNN 170 ON THE KAGGLE AND MESSIDOR TEST SETS. κ = KAPPA AGREEMENT WITH 95% CONFIDENCE INTERVALS, Se/Sp = SENSITIVITY AND SPECIFICITY, Az = AREA UNDER THE ROC WITH 95% CONFIDENCE INTERVALS

		Observer 1		Observer 2		SeS CNN 60		NSeS CNN 170	
Reference	Kaggle	-	1046	-	1040	-	940	-	939
		+	53	+	56	+	47	+	65
		$\kappa = 0.759$ [0.716-0.802]		$\kappa = 0.740$ [0.696-0.785]		$\kappa = 0.598$ [0.549-0.648]		$\kappa = 0.554$ [0.501-0.607]	
		Se/Sp = 0.816/0.947		Se/Sp = 0.806/0.942		Se/Sp = 0.837/0.851		Se/Sp = 0.774/0.851	
						Az = 0.894 [0.867-0.919]		Az = 0.895 [0.874-0.914]	
	Messidor	-	781	-	764	-	803	-	807
		+	9	+	18	+	26	+	34
		$\kappa = 0.791$ [0.753-0.829]		$\kappa = 0.742$ [0.701-0.783]		$\kappa = 0.793$ [0.755-0.832]		$\kappa = 0.783$ [0.743-0.822]	
		Se/Sp = 0.972/0.889		Se/Sp = 0.944/0.869		Se/Sp = 0.919/0.914		Se/Sp = 0.894/0.918	
						Az = 0.972 [0.963-0.980]		Az = 0.959 [0.947-0.970]	



Resultados



TABLE IV

CONTINGENCY HUMAN OBSERVERS, SeS CNN 60 AND NSeS CNN 170 ON THE KAGGLE AND MESSIDOR TEST SETS AFTER REMOVAL OF POOR QUALITY IMAGES κ = KAPPA AGREEMENT WITH 95% CONFIDENCE INTERVALS, Se/Sp = SENSITIVITY AND SPECIFICITY, Az = AREA UNDER THE ROC WITH 95% CONFIDENCE INTERVALS

		Observer 1		Observer 2		SeS CNN 60		NSeS CNN 170					
Reference	Kaggle	-	+	-	+	-	+	-	+				
		•	568	25	•	563	30	•	536	57	•	486	107
		+	23	174	+	28	169	+	30	167	+	34	163
		$\kappa = 0.838$ [0.794-0.883]		$\kappa = 0.805$ [0.756-0.853]		$\kappa = 0.719$ [0.663-0.775]		$\kappa = 0.576$ [0.512-0.639]					
		Se/Sp = 0.883/0.958		Se/Sp = 0.858/0.949		Se/Sp = 0.848/0.904 Az = 0.917 [0.888-0.944]		Se/Sp = 0.827/0.820 Az = 0.909 [0.884-0.932]					
	Messidor	-	+	-	+	-	+	-	+				
		•	727	86	•	709	104	•	744	69	•	757	56
		+	7	282	+	12	277	+	20	269	+	28	261
		$\kappa = 0.800$ [0.761-0.839]		$\kappa = 0.753$ [0.711-0.796]		$\kappa = 0.802$ [0.763-0.842]		$\kappa = 0.809$ [0.770-0.848]					
		Se/Sp = 0.976/0.894		Se/Sp = 0.958/0.872		Se/Sp = 0.931/0.915 Az = 0.979 [0.970-0.985]		Se/Sp = 0.903/0.931 Az = 0.966 [0.954-0.976]					



Referencias

- [3] A. Seff et al., “2D view aggregation for lymph node detection using a shallow hierarchy of linear classifiers,” Proc. MICCAI, vol. 8673, LNCS, pp. 544–552, 2014.
- [35] X. He and E. Frey, “ROC, LROC, FROC, AFROC: An alphabet soup,” J. Am. Coll. Radiol., vol. 6, no. 9, pp. 652–655, 2009.
- [37] N. Karssemeijer et al., “Computer-aided detection versus independent double reading of masses on mammograms,” Radiology, vol. 227, no. 1, pp. 192–200, 2003.
- [38] A. Gubern-Mérida et al., “Automated localization of breast cancer in DCE-MRI,” Med. Image Anal., vol. 20, no. 1, pp. 265–274, 2015.



INTRODUCCIÓN A LOS MODELOS COMPUTACIONALES:

CUARTO CURSO DEL GRADO DE ING. INFORMÁTICA EN COMPUTACION

REDES NEURONALES CONVOLUCIONALES: EJEMPLOS

César Hervás-Martínez
Grupo de Investigación AYRNA

**Departamento de Informática y Análisis
Numérico**
Universidad de Córdoba
Campus de Rabanales. Edificio Einstein.
Email: chervas@uco.es

2020-2021