

# Uso de redes neuronales en el procesamiento de imágenes odontológicas

Juan Ignacio Porta

Directores: Francisco Tamarit, Jorge Sanchez

Universidad Nacional de Córdoba

30 de septiembre de 2019

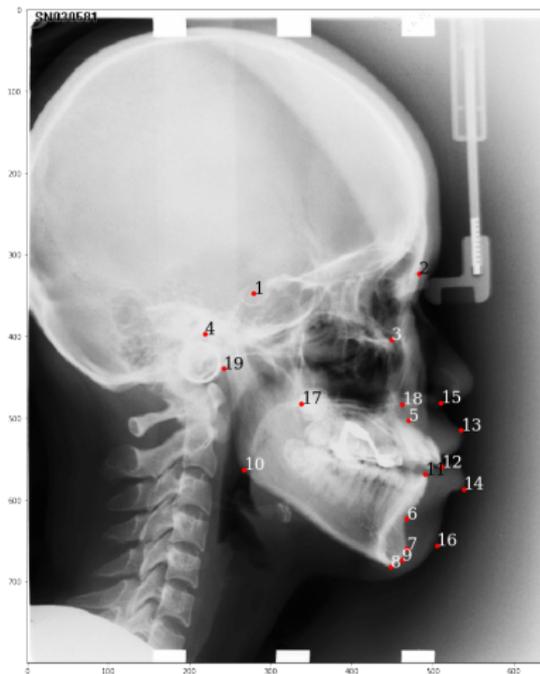
- 1 Objetivos
- 2 Descripción del problema
- 3 Aprendizaje automático
  - Redes Neuronales Artificiales
- 4 Estructuras utilizadas
- 5 Metodología
  - Descripción de modelos
- 6 Resultados
  - Entrenamiento
  - Evaluación de modelos
  - Resultados
- 7 Conclusiones

- Estudiar el problema de la detección automática de puntos característicos en imágenes
- Desarrollar modelos capaces de realizar esta tarea
- Medir el desempeño de los modelos y compararlos con el estado del arte en la materia
- Proporcionar una herramienta de detección de puntos característicos aplicable en muchos problemas de diversa índole.

En este trabajo se aborda la problemática de detectar automáticamente puntos característicos en imágenes.

En particular el ejemplo de interés será la detección de puntos cefalométricos en imágenes de perfil de cráneo en rayos X.

# Problema



**Figura:** Imagen en rayos X de perfil de cráneo con puntos cefalométricos marcados.

Los puntos cefalométricos son una parte primordial de los cefalogramas, los cuales son estudios ortodónticos de estructuras presentes en el cráneo humano, y cuya determinación y posterior análisis es una herramienta fundamental para los ortodoncistas a la hora de hacer evaluaciones del estado y/o evolución de un paciente, a la hora de planear un tratamiento odontológico, de controlarlo y de medir sus resultados.

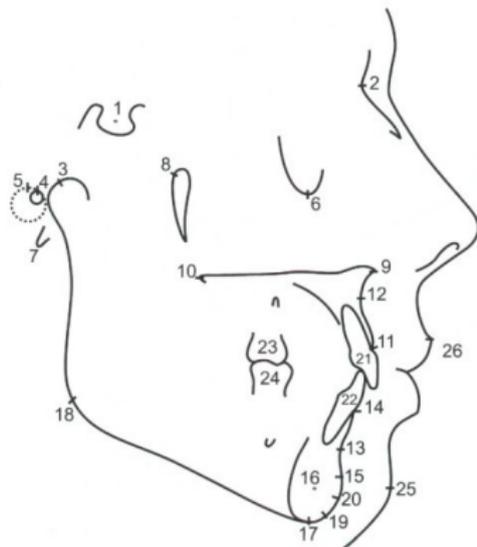


Figura: Ejemplo de cefalograma.

“Se dice que un programa de computadora aprende a través de la experiencia  $E$ , con respecto a una clase de tarea  $T$  y una medida de desempeño  $P$ , si su desempeño en tareas en  $T$ , medido por  $P$ , mejora con la experiencia  $E$ .”

Mitchell (1997).

# Métodos de aprendizaje

- Aprendizaje supervisado
- Aprendizaje por refuerzo
- Aprendizaje no supervisado

# Métodos de aprendizaje

- Aprendizaje supervisado
- Aprendizaje por refuerzo
- Aprendizaje no supervisado

- Regresión: Salida a  $\mathbb{R}^n$
- Clasificación: Salida a  $D \subset \mathbb{Z}$ ,  $D$  finito, cada elemento en  $D$  representa una clase.

# Redes Neuronales Artificiales

Una red neuronal artificial es un conglomerado de Neuronas Artificiales, unidas a través de una arquitectura de conexiones sinápticas, que tiene la capacidad de aprender a desarrollar una tarea, es decir, es una herramienta de aprendizaje automático.

# Tipos de arquitecturas

Según el flujo de información, podemos agrupar a las arquitecturas de la siguiente manera:

- Redes prealimentadas (Feed-Forward)
- Redes recurrentes

# Tipos de arquitecturas

Según el flujo de información, podemos agrupar a las arquitecturas de la siguiente manera:

- Redes prealimentadas (Feed-Forward)
- Redes recurrentes

- Neuronas de entrada, ocultas y de salida
- Estructura en capas
- Flujo direccionado de información



Estructuras y arquitecturas de Redes Neuronales utilizadas.

Definimos a  $f$  como:

$$f(\vec{\zeta}) = \left( \sum_i \mathbf{K}_i \star \zeta_i \right) + b \quad (1)$$

donde

- $\star$ : operación de convolución discreta.
- $b$ : Término independiente o *bias*.
- $\mathbf{K}_i$ : Núcleo de convolución discreta, con parámetros ajustables.

# Neurona Convolutiva

La operación de convolución discreta, en su versión de dos dimensiones (opción por defecto en datos tipo imagen) tiene la siguiente forma:

$$C[m, n] = \sum_{u=-l}^l \sum_{v=-l}^l K[u, v] \zeta[m - u, n - v] \quad (2)$$

Donde:

- $K[u, v]$  es el elemento del núcleo  $\mathbf{K}$  en la fila  $u$  y columna  $v$
- $\mathbf{K}$  tiene  $(2l + 1) \times (2l + 1)$  elementos
- $\zeta$  es 2-dimensional, de  $d \times d$  elementos
- $\mathbf{C}$  es la salida de la convolución, también 2-dimensional, en principio de  $(d - 2l) \times (d - 2l)$  elementos

# Neurona Convencional

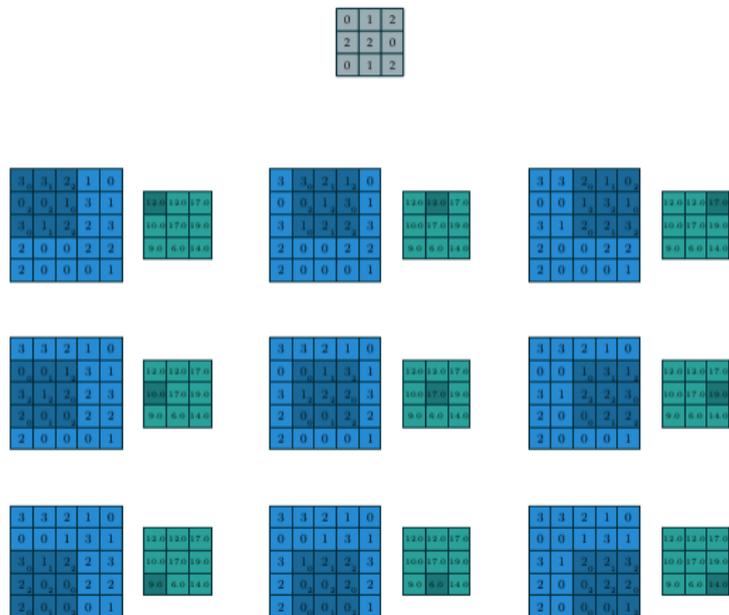


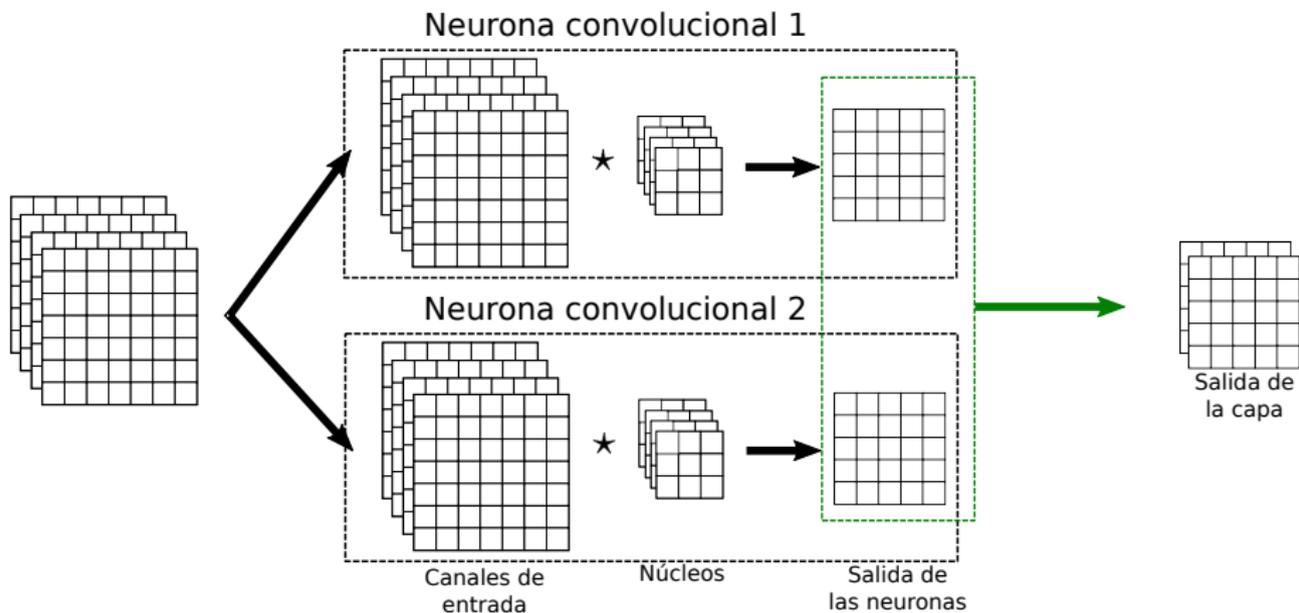
Figura: Ejemplo de convolución discreta, en su forma más simple. El núcleo se presenta en gris, la entrada en celeste y la salida en verde. Imagen extraída de Dumoulin y Visin (2016).

Volviendo a la neurona convolutiva

$$f(\vec{\zeta}) = \left( \sum_i \mathbf{K}_i \star \zeta_i \right) + b$$

# Capa Convolutiva

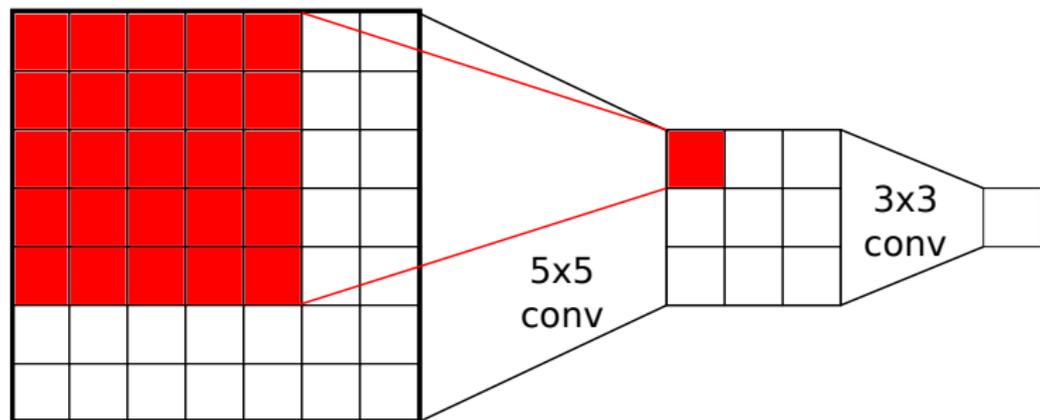
Volviendo a la neurona convolutiva



**Figura:** Esquema de capa convolutiva con 2 neuronas de núcleo  $3 \times 3$  (en realidad  $3 \times 3 \times 4$ , donde la cantidad de canales del núcleo queda implícitamente definida por la cantidad de canales de entrada).

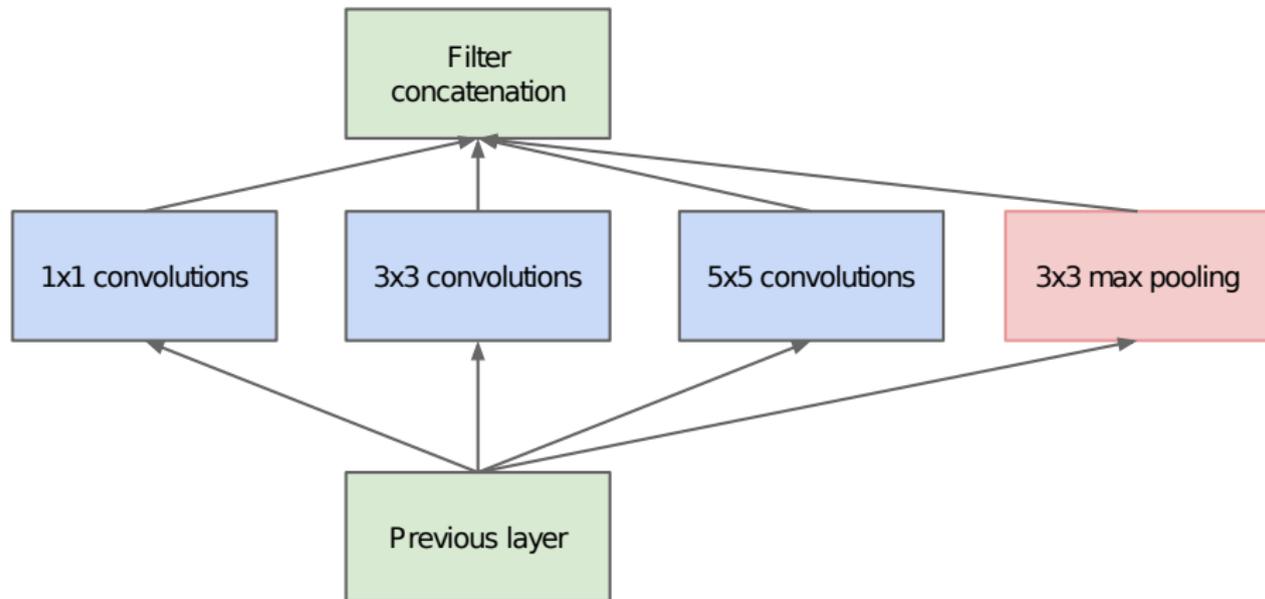
# Campo receptivo

Campo receptivo: región de la imagen de entrada o de alguna entrada anterior que aporta información para el cálculo del valor de un píxel de salida.



**Figura:** Esquema de representación del campo receptivo teórico a través de operaciones convolucionales. La región roja en la primera capa representa el campo receptivo del píxel rojo en la segunda capa, mientras que la primera capa completa es el campo receptivo del píxel de la tercera capa.

# Capa Inception



**Figura:** Esquema de la versión naïve del módulo inception. Como se puede ver, se aplican distintas capas convolucionales o de pooling en paralelo, y luego se las concatenan en la dimensión de los canales. Figura extraída de Szegedy y col. (2015).

# Autoencoder

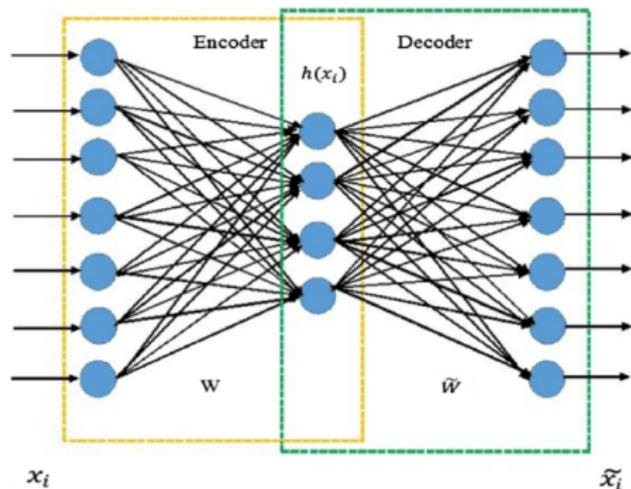


Figura: Ejemplo de la arquitectura básica de un autoencoder. Figura extraída de Ahmed et al. (2018).

## Conjunto de datos:

- Dataset público utilizado en IEEE ISBI 2015 Challenge
- 400 radiografías cefalométricas obtenidas de 400 pacientes con una edad entre 6 y 60 años.
- resolución de las imágenes de  $1935 \times 2400$  píxeles.
- 19 puntos por imagen marcados por dos doctores expertos.
- Imágenes reducidas en 3 en cada dimensión.

# Autoencoder convolucional restringido

- Autoencoder convolucional simple restringido
- Restricción: se busca reproducir un mapa de probabilidad de que cada pixel de entrada sea un landmark en específico.
- Mapa de entrenamiento construido en base a gaussiana, con  $\sigma$  relacionado con las restricciones impuestas en las métricas del problema.

El algoritmo de descenso de gradiente elegido es Adam, el cual presenta notables mejoras en el tiempo de convergencia con respecto a los algoritmos de descenso por gradiente tradicionales (Kingma y Ba, 2014).

# Autoencoder Inception

- Similar al anterior, pero con capas inception en el encoder y mayor profundidad.
- Se busca tener distintos niveles de localidad, ya que la experticia ortodóntica nos marca que hay puntos alejados en el espacio pero relacionados en su localización.
- Dos variantes del modelo, diferenciadas por la cantidad de canales en capas convolucionales del decoder. Se les llama **autoencoder inception 32** y **autoencoder inception 64**, haciendo referencia al número de canales.
- El optimizador elegido es el mismo que en el modelo anterior.

# Autoencoder Inception

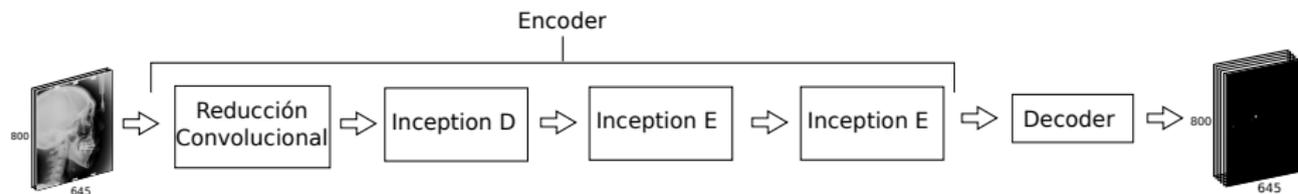


Figura: Esquema general del Autoencoder Inception

# Función de costo

Usualmente en los autoencoders se utiliza como función de costo el error cuadrático medio. Nosotros utilizaremos Entropía cruzada binaria, de la siguiente forma:

$$\frac{1}{N} \sum_{i=1}^N -w_i [p_i y_i \cdot \log \sigma(x_i) + (1 - y_i) \cdot \log (1 - \sigma(x_i))]. \quad (3)$$

donde:

- $N$  es el número de elementos en el batch
- $w_i$  es un factor de peso (opcional) entre los elementos del batch
- $y_i$  es la clase a comparar (0 o 1).
- $x_i$  es la salida del modelo,  $x_i \in [0, 1] \subseteq \mathbb{R}$
- $p_i$  es el peso otorgado a la clase positiva.
- $\sigma$  es la función sigmoide.

El peso  $p_i$  será considerado un hiperparámetro a ajustar, que nos permitirá cambiar el modelo variando el hiperparámetro.

## Coeficiente de detecciones exitosas

El coeficiente de detecciones exitosas o *sucess detection rate*  $p_z$  con precisión menor a  $z$  mm se formula como (Wang et al. (2016)):

$$p_z = \frac{\#\{j : \|L_d(j) - L_r(j)\| < z\}}{\#\Omega} \times 100\%. \quad (4)$$

donde  $L_d, L_r$  representan la localización del landmark detectado y del landmark etiquetado, respectivamente,  $z$  denota la precisión en la medición, en nuestro caso cuatro valores:  $z = 2mm, 2,5mm, 3mm$  y  $4mm$ ;  $j \in \Omega$ , y  $\#\Omega$  representa el número de detecciones realizadas.



# Transformaciones en el Dataset

Antes de entrenar los modelos, existe una serie de transformaciones realizadas sobre el conjunto de datos.

- Se realizan reescalados aleatorios de la imagen (entre el 98 % y 102 %)
- Se realizan traslaciones aleatorias horizontales y verticales a la imagen (de hasta 2 % de la imagen en cada dirección)
- Se realizan rotaciones aleatorias a la imagen (hasta 5°)
- Se normaliza el conjunto de imágenes para tener media 0 y varianza 1.

Los valores determinados previamente son:

- Tamaño del Batch: 5
- Número de épocas: 250

Búsqueda en grilla:

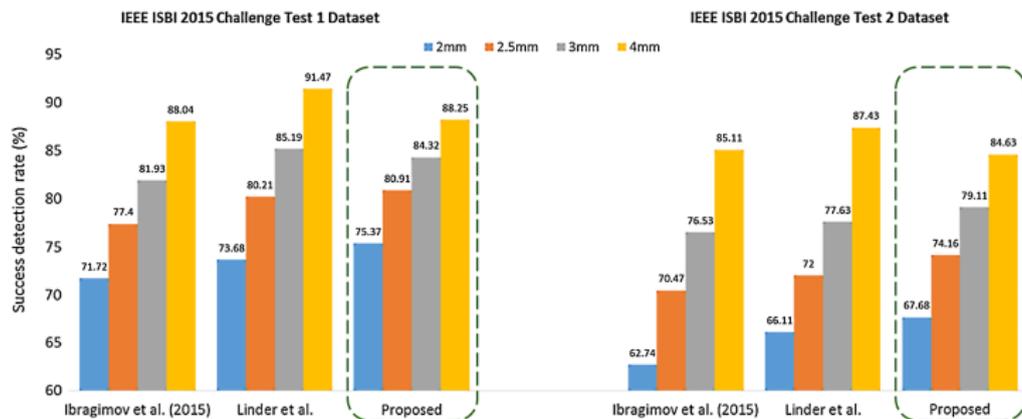
Se realiza una búsqueda en grilla del resto de los hiperparámetros sobre una partición aleatoria del conjunto de entrenamiento. Se entrena sobre el 80 % del mismo y se mide la métrica sobre el 20 % restante. La métrica utilizada es el Coeficiente de detecciones exitosas a  $2mm$ .

Como los valores encontrados para el modelo Autoencoder simple no son comparables a los de la literatura, se descarta este modelo y se continúa con las dos variantes del autoencoder inception, ya que asumimos que el problema de este modelo es la falta de complejidad y la localidad del mismo.



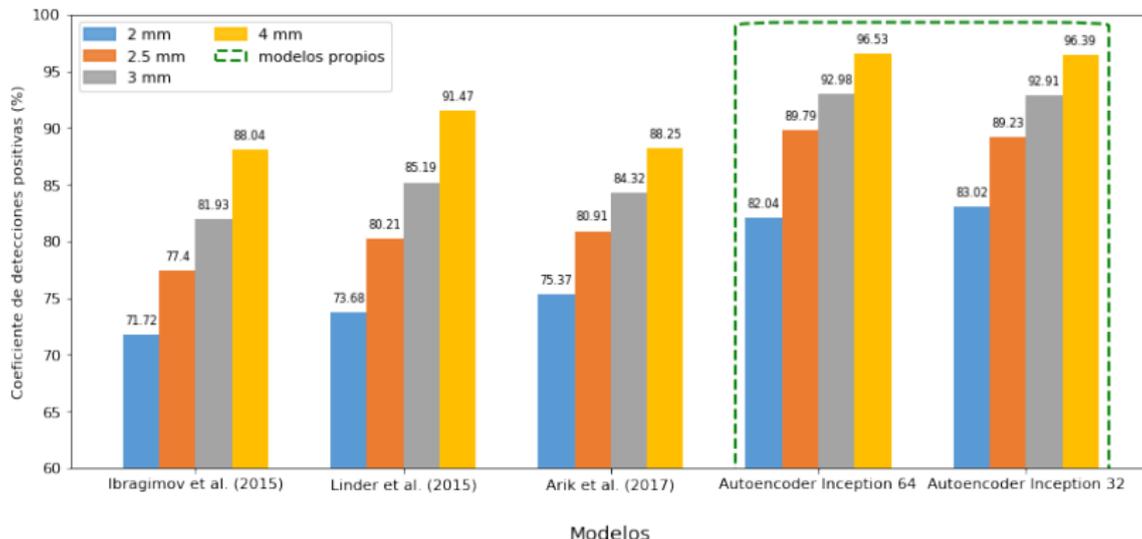
# Conjuntos de evaluación

Existen dos conjuntos de evaluación, con 150 y 100 imágenes respectivamente. Podemos ver que los modelos de la literatura se desempeñan de peor manera en las detecciones del segundo conjunto de evaluación.



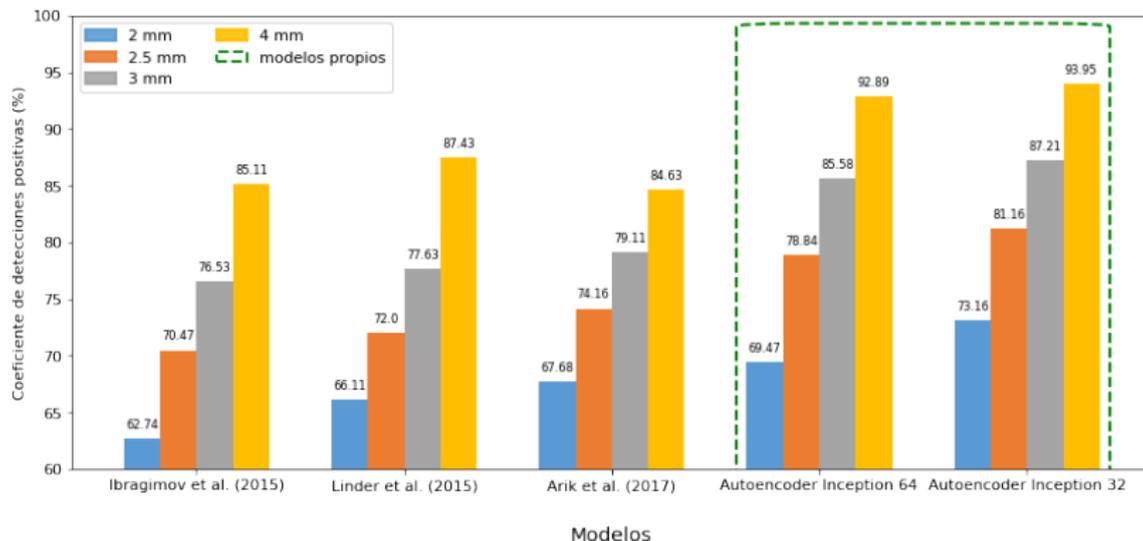
**Figura:** Comparación de los modelos de Wang et al. (2016) y Arik et al. (2017) en las métricas de coeficiente de detecciones exitosas para 2, 2.5, 3 y 4 mm. El modelo etiquetado como proposed es el perteneciente a este último. Imagen extraída de Arik et al. (2017).

# Resultados



**Figura:** Resultados de los modelos propios comparados con los modelos de Ibragimov y Col. y Lindner y Cootes presentados en Wang et al. (2016) y el modelo de Arik et al. (2017) sobre el conjunto de test 1. Las métricas evaluadas son los coeficientes de detecciones positivas para 2, 2.5, 3 y 4 mm. En recuadro verde los modelos desarrollados en este trabajo.

# Resultados

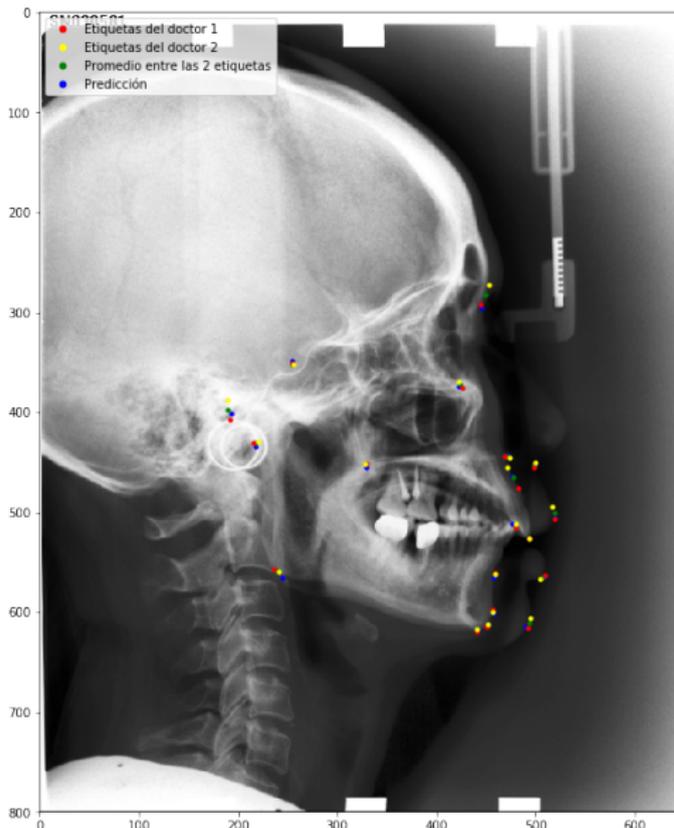


**Figura:** Resultados de los modelos propios comparados con los modelos de Ibragimov y Col. y Lindner y Cootes presentados en Wang et al. (2016) y el modelo de Arik et al. (2017) sobre el conjunto de test 2. Las métricas evaluadas son los coeficientes de detecciones positivas para 2, 2.5, 3 y 4 mm. En recuadro verde los modelos desarrollados en este trabajo.

Como se puede ver en las figuras, los modelos originales presentados en este trabajo superan en todas las métricas de coeficiente de detecciones exitosas a los modelos previos. Las mejoras obtenidas con respecto a la literatura son considerables, sobre todo teniendo en cuenta las diferencias que los modelos previos presentan entre ellos.

# Resultados

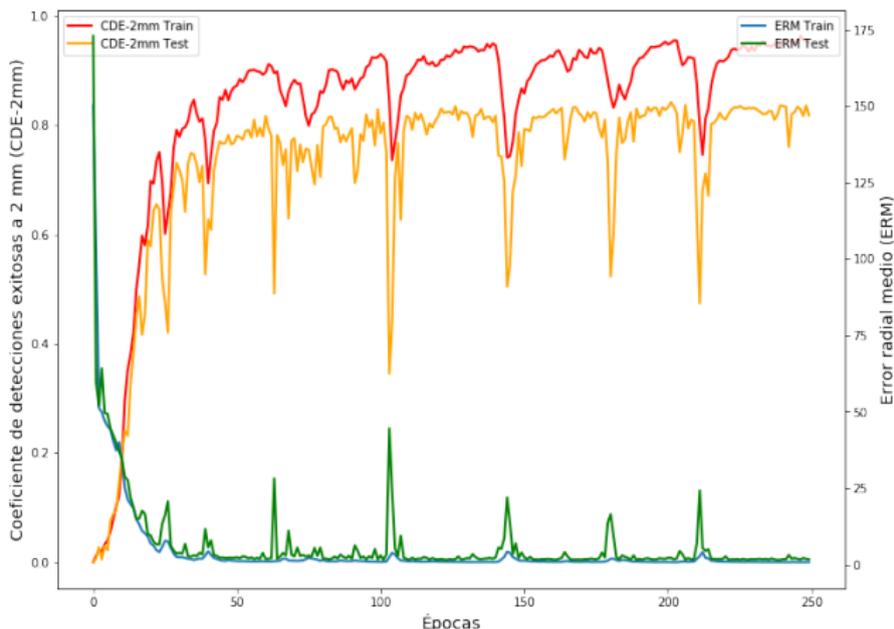
Ejemplo de detecciones para nuestro mejor modelo:



**Figura:** Detecciones y etiquetas sobre una imagen del conjunto de test 1. Recordar que los puntos considerados como verdaderos son el promedio entre las etiquetas de los dos doctores, marcados en verde.

# Resultados

Desempeño a través de las épocas de aprendizaje:



**Figura:** Coeficiente de detecciones exitosas a 2 mm y error radial medio en los conjuntos de entrenamiento y test 1 en función de las épocas de aprendizaje.

# Conclusiones

- Hemos estudiado el problema de la detección automática de puntos característicos, o landmarks, en imágenes odontológicas obtenidas a partir de técnicas de rayos X desde una perspectiva neuronal, utilizando redes neuronales convolucionales profundas.
- Hemos abordado un problema de particular interés desde el punto de vista de la salud.
- Nos dedicamos a un problema desafiante desde la perspectiva de la inteligencia artificial, que ha generado incluso competencias internacionales en el tema.
- Por primera vez para este problema hemos aplicado modelos exclusivamente neuronales, e introducido ideas originales como la idea de autoencoder restringido, enriqueciéndolo a su vez con la incorporación de capas inception.

# Conclusiones

- Hemos comparado los modelos autoencoder simple y autoencoder inception 32 y 64, y mostramos que la introducción de capas inception mejora notablemente el desempeño, aunque aumentando también la profundidad del modelo y el número de parámetros.
- Se realizaron exploraciones de hiperparámetros para encontrar los mejores modelos, y se descartó el modelo de autoencoder simple restringido, por no alcanzar valores comparables al estado del arte en la métrica coeficiente de detecciones exitosas.
- Se evaluaron los modelos restantes y se los comparó con los mejores modelos del estado del arte. Ambos modelos mostraron una gran precisión en detección de posiciones de landmarks, superando hasta en más de 7 puntos porcentuales los desempeños de los mejores modelos sobre la literatura.

# Bibliografía I



Sercan Ö Arik, Bulat Ibragimov, and Lei Xing, *Fully automated quantitative cephalometry using convolutional neural networks*, Journal of Medical Imaging **4** (2017), no. 1, 014501.



Hosameldin Ahmed, M L. Dennis Wong, and Asoke Nandi, *Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features*, Mechanical Systems and Signal Processing **99** (2018), 459–477.



María José Arcieri, Gabriela López, Mariana Romeo, Cecilia Soto, Ana Inés Villar, and Daniela Dominguez, *¿ es aplicable el cefalograma de ricketts en diferentes poblaciones?*, Actas odontológicas **10** (2016), no. 2, 12–18.



John Cardillo and Maher A Sid-Ahmed, *An image processing system for locating craniofacial landmarks*, IEEE transactions on medical imaging **13** (1994), no. 2, 275–289.



Vincent Dumoulin and Francesco Visin, *A guide to convolution arithmetic for deep learning*, arXiv e-prints (2016), arXiv:1603.07285.



Idris El-Feghi, Maher A Sid-Ahmed, and Majid Ahmadi, *Automatic localization of craniofacial landmarks for assisted cephalometry*, Pattern Recognition **37** (2004), no. 3, 609–621.



Ricardo A Guarnieri, Enio B Pereira, and Sin Chan Chou, *Solar radiation forecast using artificial neural networks in south brazil*, Proceedings of the 8th ICSHMO (2006), 24–28.



J Hertz, John , Krough , Anders Flisberg, Palmer , and Richard G, *Introduction to the theory of neural computation*, vol. 44, ch. 1,5, 01 1991.



J J Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, Proceedings of the National Academy of Sciences **79** (1982), no. 8, 2554–2558.

# Bibliografía II



Bulat Ibragimov, Boštjan Likar, F Pernus, and Tomaž Vrtovec, *Automatic cephalometric x-ray landmark detection by applying game theory and random forests*, Proc. ISBI Int. Symp. on Biomedical Imaging, 2014.



Diederik P. Kingma and Jimmy Ba, *Adam: A method for stochastic optimization*, CoRR **abs/1412.6980** (2014).



Claudia Lindner, Paul A Bromiley, Mircea C Ionita, and Tim F Cootes, *Robust and accurate shape model matching using random forest regression-voting*, IEEE transactions on pattern analysis and machine intelligence **37** (2015), no. 9, 1862–1874.



Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel, *Understanding the effective receptive field in deep convolutional neural networks*, Advances in Neural Information Processing Systems 29 (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), Curran Associates, Inc., 2016, pp. 4898–4906.



Tom Mitchell, *Machine learning*, McGraw-Hill, New York, 1997.



Robert M Ricketts, *Orthodontic diagnosis and planning:—their roles in preventive and rehabilitative dentistry*, vol. 2, Rocky Mountain/Orthodontics, 1982.



Frank Rosenblatt, *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*, Tech. report, CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.



C. Szegedy, , P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, 1–9.



Ana Torre, F Garcia, I Moromi, P Espinoza, and L Acuña, *Prediction of compression strength of high performance concrete using artificial neural networks*, Journal of Physics: Conference Series **582** (2015).



Flavio Vellini-Ferreira, *Ortodoncia: Diagnóstico y planificación clínica*, Editorial Artes Médicas, (2002), 313–314.

# Bibliografía III



Ching-Wei Wang, Cheng-Ta Huang, Meng-Che Hsieh, Chung-Hsing Li, Sheng-Wei Chang, Wei-Cheng Li, Rémy Vandaele, Raphaël Marée, Sébastien Jodogne, Pierre Geurts, et al., *Evaluation and comparison of anatomical landmark detection methods for cephalometric x-ray images: a grand challenge*, IEEE transactions on medical imaging **34** (2015), no. 9, 1890–1900.



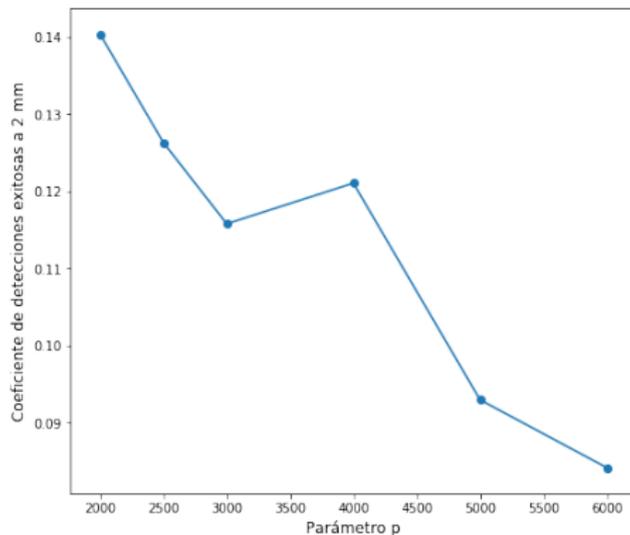
Ching-Wei Wang, Cheng-Ta Huang, Jia-Hong Lee, Chung-Hsing Li, Sheng-Wei Chang, Ming-Jhih Siao, Tat-Ming Lai, Bulat Ibragimov, Tomaz Vrtovec, Olaf Ronneberger, et al., *A benchmark for comparison of dental radiography analysis algorithms*, Medical image analysis **31** (2016), 63–76.



Weining Yue, Dali Yin, Chengjun Li, Guoping Wang, and Tianmin Xu, *Automated 2-d cephalometric analysis on x-ray images by a model-based approach*, IEEE transactions on biomedical engineering **53** (2006), no. 8, 1615–1623.

# Anexo: Resultados de la búsqueda en grilla.

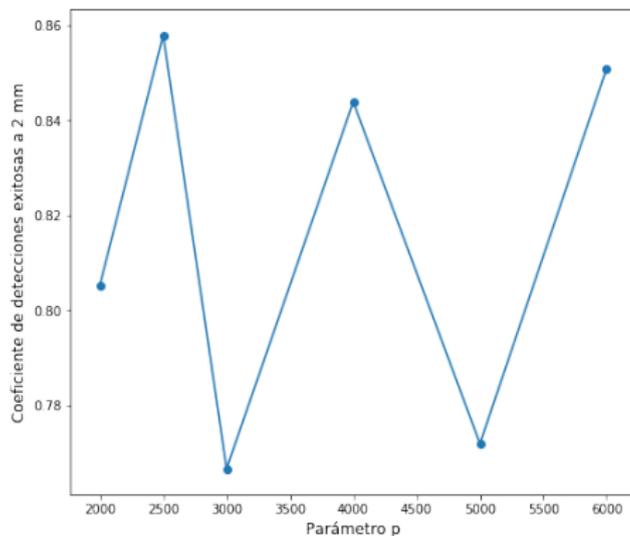
Resultados en  $p$ :



**Figura:** Resultados de la búsqueda en grilla sobre el hiperparámetro  $p$  para el modelo Autoencoder simple. El modelo que presenta el mejor resultado es  $p = 2000$ .

## Anexo: Resultados de la búsqueda en grilla.

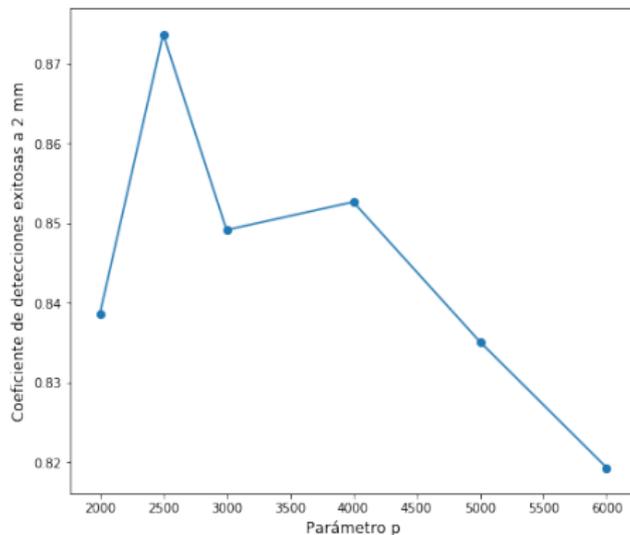
Resultados en  $p$ :



**Figura:** Resultados de la búsqueda en grilla sobre el hiperparámetro  $p$  para el modelo Autoencoder Inception 32. El modelo que presenta el mejor resultado es  $p = 2500$ .

# Anexo: Resultados de la búsqueda en grilla.

Resultados en  $p$ :



**Figura:** Resultados de la búsqueda en grilla sobre el hiperparámetro  $p$  para el modelo Autoencoder Inception 64. El modelo que presenta el mejor resultado es  $p = 2500$ .

## Anexo: Resultados de la búsqueda en grilla.

Resultados en  $\epsilon$  y  $\lambda$ :

		tasa de aprendizaje $\epsilon$ :		
		$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$
peso de regularización $\lambda$ :	$\lambda = 10^{-3}$	4,2 %	0,4 %	0 %
	$\lambda = 10^{-4}$	10,9 %	0 %	0 %
	$\lambda = 10^{-5}$	14,0 %	0,4 %	0 %

**Cuadro:** Resultados de la búsqueda en grilla sobre los hiperparámetros  $\epsilon$  y  $\lambda$  para el modelo Autoencoder simple.

## Anexo: Resultados de la búsqueda en grilla.

Resultados en  $\epsilon$  y  $\lambda$ :

		tasa de aprendizaje $\epsilon$ :		
		$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$
peso de regularización $\lambda$ :	$\lambda = 10^{-3}$	5,6 %	5,6 %	4,4 %
	$\lambda = 10^{-4}$	85,1 %	51,2 %	4,4 %
	$\lambda = 10^{-5}$	85,8 %	74,9 %	4,2 %

**Cuadro:** Resultados de la búsqueda en grilla sobre los hiperparámetros  $\epsilon$  y  $\lambda$  para el modelo Autoencoder inception 32.

## Anexo: Resultados de la búsqueda en grilla.

Resultados en  $\epsilon$  y  $\lambda$ :

		tasa de aprendizaje $\epsilon$ :		
		$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-5}$
peso de regularización $\lambda$ :	$\lambda = 10^{-3}$	5,6 %	7,7 %	4,4 %
	$\lambda = 10^{-4}$	81,6 %	77,0 %	7,5 %
	$\lambda = 10^{-5}$	87,4 %	81,1 %	7,5 %

**Cuadro:** Resultados de la búsqueda en grilla sobre los hiperparámetros  $\epsilon$  y  $\lambda$  para el modelo Autoencoder inception 64.