

# Detección de Rostros basado en *Deep Learning*

Diego Irarrázaval

Universidad de Chile, FCFM

*diego.irarrazaval@ug.uchile.cl*

22 de Enero, 2020

- 1 Introducción
  - Problema a resolver
- 2 Metodología
  - *Backbone* y Optimizador
- 3 Resultados
- 4 Conclusiones

# Problema a resolver

Problema a resolver por la red:



Figure: Detección de rostros basado en *Deep Learning*[1].

## Código base:

Se utiliza como modelo base la red RetinaFace. Repositorio con modificaciones propias en [https://github.com/Diego-II/Pytorch\\_Retinaface](https://github.com/Diego-II/Pytorch_Retinaface)

### Backbone utilizados:

- 1 ResNet-50
- 2 Res2Net-101 [2]
- 3 EfficientNet-b5 [3]

### Optimizadores:

- 1 SGD
- 2 AdaBoundW [4]
- 3 SDGP [5]

Modelo	Optimizador	Hiperparametros	Resultados Easy	Resultados Medium	Resultados Hard
ResNet-50	SDG	150 épocas, bs 32	91.90%	88.80%	77.01%
ResNet-50	AdaBoundW	150 épocas, bs 32	94.90%	93.58%	85.89%
Res2Net-101	AdaBoundW	150 épocas, bs 16	95.60%	93.62%	86.01%
EfficientNet-b5	SDGP	80 épocas, bs 12	00.12%	00.19%	00.15%

**Table:** Modelos utilizados junto con los hiperparámetros del entrenamiento y los resultados obtenidos.

- Uso de **DGX**, ambientes como *Docker* y otros.
- Importante efecto del optimizador en entrenamiento.
- Más capas/parámetros no implica mejor desempeño.
- Dificultades.

# Referencias



[Pytorch Retinaface](#)

[https://github.com/biubug6/Pytorch\\_Retinaface](https://github.com/biubug6/Pytorch_Retinaface).



[Res2Net Pretrained Models.](#)

<https://github.com/Res2Net/Res2Net-PretrainedModels>.



[Timm](#)

<https://rwightman.github.io/pytorch-image-models/>



[AdaBound](#)

<https://github.com/Luolc/AdaBound>



[AdamP](#)

<https://github.com/clovaai/adamp>