UNIVERSIDAD DE SANTIAGO DE CHILE

FACULTAD DE CIENCIA

Departamento de Física

Formato de Tesis, en Latex, para el Departamento de Física, USACH

Student Studentsson

Profesor Guía:
Professor Professorson

Proyecto de Título para optar al Título Profesional de Diploma Diplomasson

Santiago - Chile

2021

Si su trabajo de titulación forma parte de un proyecto financiado con fondos públicos, indique aquí:

Tipo Proyecto — Número — Título del Proyecto.

Formato de Tesis, en Latex, para el Departamento de Física, USACH

Student Studentsson

Este trabajo de graduación fue elaborado bajo la supervisión del profesor patrocinante Professor Professorson y ha sido aprobado por los miembros de la comisión calificadora.

Professor Professorson 2 _____

Professor Professorson 3 _____

_____

Professor Professorson

DEDICATORIA

geschrieben werden

AGRADECIMIENTOS

geschrieben werden

RESUMEN

En este trabajo de tesis se han presentado 3 problemas de procesamiento de imágenes utilizando técnicas de aprendizaje profundo para resolverlos. Gracias al poder computacional actual y la evolución en los algoritmos de inteligencia artificial, el campo del procesamiento de imágenes ha sido un gran beneficiado.

Se ha creado un clasificador de imágenes de radiografías de tórax para identificar Covid-19, Pneumonia y pacientes sanos, la particularidad de esta implementación es que puede ser facilmente modificable para implementarla en cualquier otro tipo de problema de clasificación de imágenes. La implementación está escrita completamente en Python y la red puede ser entrenada tanto en un compilador como en un *jupyter notebook*.

Palabras Claves: Aprendizaje Automático, CNN, Rayos X, Detector Facial, YOLO, Detector de mascarillas

ABSTRACT

In this thesis, 3 image processing problems have been presented using deep learning techniques to solve them. Thanks to the current computational power and the evolution of artificial intelligence algorithms, the field of image processing has been a great beneficiary.

An X-ray chest image classifier has been created to identify Covid-19, Pneumonia and healthy patients, the strong point of this implementation is that it can be easily modified to implement it for any other type of image classification problem. The implementation is written entirely in Python and the network can be trained in both a compiler or in a *jupyter notebook*.

Keywords: Machine Learning, CNN, X-ray, Face Detector, YOLO, Mask Detector

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO 1. INTRODUCCIÓN

1.1 ANTECEDENTES Y MOTIVACIÓN

geschrieben werden

1.2 OBJETIVOS Y ALCANCE

1.2.1 Objetivo general

1.2.2 Objetivos específicos

1.2.3 Alcances

1.3 METODOLOGÍA Y HERRAMIENTAS UTILIZADAS

1.3.1 Metodología

1.3.2 Herramientas de desarrollo

1.4 CASOS DE ESTUDIO

1.4.1 Dataset: Covid-19

1.4.2 Reconocimiento facial

1.4.3 Detector de Objetos: YOLO

CAPÍTULO 2. MARCO TEÓRICO

2.1 REDES NEURONALES ARTIFICIALES

Con n entradas en el nodo desde x_1 hasta x_n y φ la función de activación. La salida del nodo número j es:

$$y_j = \varphi(\sum_{j=i}^{n} w_{ji}x_i + b_j) \tag{2.1}$$

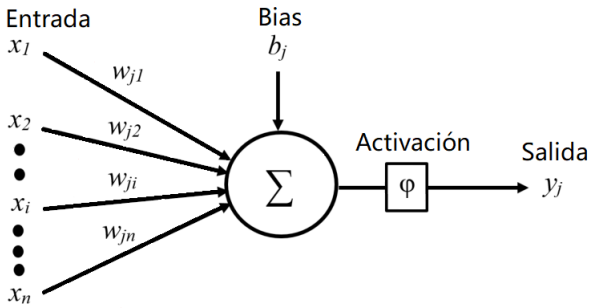

Figura 2.1: Nodo artificial unitario que imita la estructura cerebral. Su función es aprender y es un elemento básico de aprendizaje.

2.2 FUNCIONES DE ACTIVACIÓN

2.2.1 Sigmoide

También se conoce como función de activación logística. Toma un número de valor real y lo acota en un rango entre 0 y 1. También se usa en la capa de salida donde el objetivo final es predecir una probabilidad. Convierte grandes números negativos en 0 y grandes números positivos en 1. Matemáticamente se representa como:

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}} \tag{2.2}$$

y su derivada:

$$\frac{\partial}{\partial x}\sigma(x) = \sigma(x)(1 - \sigma(x)) \tag{2.3}$$

2.2.2 LeakyReLU

$$\frac{\partial}{\partial x}z(x) = \begin{cases} 1 & x > 0 \\ 0.1 & x < 0 \end{cases} \tag{2.4}$$

2.3 MÉTODOS DE OPTIMIZACIÓN

2.3.1 Descenso del Gradiente Estocástico

$$W_{t+1} = W_t - \alpha \nabla f(W_t) \tag{2.5}$$

donde α es la tasa de aprendizaje y $\nabla f(W_t)$ es el gradiente (o derivada) de la función de pérdida con respecto de W.

2.4 PROPAGACIÓN HACIA ATRÁS (*BACKPROPAGATION*)

Es una técnica utilizada para realizar eficientemente el descenso del gradiente en una red neuronal (Rumelhart et al., 1988).

2.5 MÁQUINAS DE SOPORTE VECTORIAL (*SUPPORT-VECTOR MACHINE*)

Las Máquinas de Soporte Vectorial (SVM) son de los algoritmos más populares en Machine Learning. Estos suelen utilizarse en problemas de clasificación, regresión, e incluso detección de valores atípicos (*outliers*). El método de soporte vectorial fue presentado por Boser, Guyon y Vapnik (Boser et al., 1992) en la Conferencia ACM de Teoría del Aprendizaje Computacional (COLT92).

La idea de construir un hiperplano separador óptimo en un contexto no-paramétrico, desarrollado por Vapnik y Chervonenkis (Vapnik & Chervonenkis, 1964) y por Cover (Cover, 1965).

2.6 REDES NEURONALES CONVOLUCIONALES (*CNN*)

2.6.1 Capas Convolucionales

2.7 ARQUITECTURAS COMUNES

2.7.1 YOLO: You Only Look Once

YOLO fue presentado en el 2016 como un nuevo enfoque para la detección de objetos. En vez de re-utilizar clasificadores para realizar la detección, este enmarca la detección de objectos como un problema de regresión de cajas delimitadores (*bounding box*) espacialmente separadas con una probabilidad asociada. Solo una red neuronal predice cajas delimitadores y probabilidades de las clases directamente desde la imagen completa en una sola evaluación. De esta manera, esta arquitectura se deshace de complejos diagramas de flujo (*pipelines*) y la hace extremadamente rápida (Redmon et al., 2016).

Columna	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19	74.1	91.8	7.29	1246	**171**
Resnet-101	77.1	93.7	19.7	1039	53
Resnet-152	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

Tabla 2.1: Comparación de Darknet-53 con otras columnas vertebrales (*backbones*) de otras arquitecturas.

	Tipo	Filtros	Tamaño	Salida
	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3 / 2	128 x 128
	Convolutional	32	1 x 1	
1 x	Convolutional	64	3 x 3	
	Residual			128 x 128
	Convolutional	128	3 x 3 / 2	64 x 64
	Convolutional	64	1 x 1	
2 x	Convolutional	128	3 x 3	
	Residual			64 x 64
	Convolutional	256	3 x 3 / 2	32 x 32
	Convolutional	128	1 x 1	
8 x	Convolutional	256	3 x 3	
	Residual			32 x 32
	Convolutional	512	3 x 3 / 2	16 x 16
	Convolutional	256	1 x 1	
8 x	Convolutional	512	3 x 3	
	Residual			16 x 16
	Convolutional	1024	3 x 3 / 2	8 x 8
	Convolutional	512	1 x 1	
4 x	Convolutional	1024	3 x 3	
	Residual			8 x 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Tabla 2.2: Arquitectura Darknet-53

CAPÍTULO 3. IMPLEMENTACIÓN Y RESULTADOS

3.1 DATASET: COVID-19

3.1.1 Implementación

```
total_train = sum(len(files) for _, _, files in os.walk(traindir))
pbar = tqdm(total=total_train, desc='Loading training images')
for d in os.listdir(traindir):
    categories.append(d)
    train_imgs = os.listdir(os.path.join(traindir, d))
    valid_imgs = os.listdir(os.path.join(validdir, d))
    test_imgs = os.listdir(os.path.join(testdir, d))
    n_train.append(len(train_imgs))
    n_valid.append(len(valid_imgs))
    n_test.append(len(test_imgs))

    for i in train_imgs:
        img_categories.append(d)
        img = cv2.imread(os.path.join(traindir, d, i))
        img_array = np.array(img)
        pbar.update(1)
        hs.append(img_array.shape[0])
        ws.append(img_array.shape[1])
pbar.close()

cat_df = pd.DataFrame({'category': categories,
                       'n_train': n_train,
                       'n_valid': n_valid,
                       'n_test': n_test}).\
                        sort_values('category')

cat_df.sort_values('n_train', ascending=False, inplace=True)
```

3.2 RECONOCIMIENTO FACIAL

3.2.1 Implementación

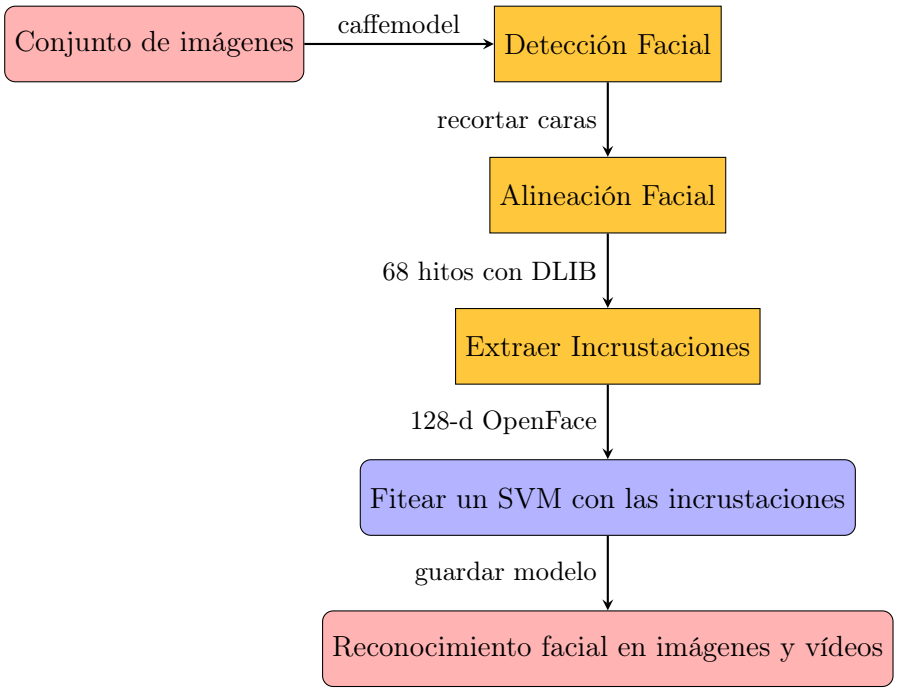

Figura 3.1: Diagrama de Flujo en la implementación para el reconocimiento y clasificación facial

3.3 YOLO: DETECTOR DE MASCARILLAS

3.3.1 Implementación

3.3.2 Resultados

(a) Hay una sobredetección de un rostro con mascarilla y una baja detección de un rostro sin mascarilla.

(b) Hay una sobredetección, No-Mask, sobre una detección correcta, esto se podría corregir con un umbral mayor de Non-maximum Suppression.

(c) Todas las personas usando mascarillas se han detectado correctamente.

(d) Se he detectado correctamente las personas con y sin mascarillas.

Figura 3.2: Imágenes reservadas para pruebas. Aunque las métricas del modelo son bastante favorables, en producción, el modelo podría tener un error mucho más alto. La única manera de corregirlo es aumentar la cantidad de datos de entrenamiento recopilando más imágenes y anotaciones.

CONCLUSIONES

geschrieben werden

REFERENCIAS BIBLIOGRÁFICAS

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In D. Haussler (Ed.) *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92)*, (pp. 144–152). Pittsburgh, PA, USA: ACM Press.

URL http://doi.acm.org/10.1145/130385.130401

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, *EC-14*(3), 326–334.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection.

URL https://arxiv.org/abs/1506.02640

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). *Learning Representations by Back-Propagating Errors*, (p. 696–699). Cambridge, MA, USA: MIT Press.

URL https://dl.acm.org/doi/book/10.5555/65669

Vapnik, V., & Chervonenkis, A. (1964). A note on a class of perceptrons. In *Automation and Remote Control,*, no. 25 in Automation and Remote Control„ (pp. 103–109).

APÉNDICE A. APÉNDICE

geschrieben werden