

Universidad de Buenos Aires
Facultad de Ingeniería



Propuesta de Trabajo Profesional

APLICACIÓN DE REDES NEURONALES DE
CONVOLUCIÓN PARA EVALUAR CALIDAD DE
IMÁGENES EN REDES SOCIALES

Martín Ezequiel Buchwald - 93155
Ezequiel David Genender Peña - 93163

Mayo 2015

Índice

1. Introducción	2
2. Descripción del Problema	2
2.1. Recolección de datos	2
2.2. Algoritmos para clasificación de imágenes	3
3. Trabajos Relacionados	4
4. Objetivos	5
4.1. Aprendiendo sobre los <i>features</i> importantes	6
4.2. Integración con herramientas Open Source	6
5. Características del Trabajo	6
5.1. Modulo de recolección de datos	6
5.2. Modulo de calificador automático	7
5.3. Resultados: Comparación de resultados obtenidos	7
6. Tecnologías	7
6.1. Librería principal: Theano	7
6.2. Herramientas	8
7. Alcance	8
8. Plan de Trabajo	8
8.1. Equipo de Trabajo	8
8.2. Metodología	9
8.3. Estimación	9
8.4. Cronograma de entregables	10

1. Introducción

El siguiente documento presenta la propuesta de Trabajo Profesional de Ingeniería en Informática de los estudiantes Martín Buchwald (padrón 93155) y Ezequiel Genender Peña (padrón 93163).

El objetivo del proyecto es aplicar los conocimientos adquiridos en la carrera; el tema elegido es '**Aplicación de Redes Neuronales de Convolución para Evaluar Calidad de Imágenes en Redes Sociales**'.

Sitios como Flickr¹, Deviantart² o 500px.com³ (y otras redes sociales) tienen una abundante información sobre la calidad de una foto, las imágenes con mayor cantidad de *comentarios*, *likes* o *calificaciones positivas* tienden a ser buenas y las fotos que pasan desapercibidas tienden a ser malas. Estos sitios pueden tomarse como un gigantesco set de entrenamiento para un algoritmo de clasificación, que dada una foto decida automáticamente si la misma es buena o no, o incluso le asigne una calificación, por ejemplo, otorgándole un puntaje de 1 a 10.

El objetivo de este Trabajo Profesional es desarrollar dicho algoritmo, apoyado en un sistema de recolección de datos de los sitios mencionados, calificar dichas recolecciones, y poner a prueba el sistema dentro de las redes sociales.

2. Descripción del Problema

Hay un gran avance en el estudio de las distintas ramas de aprendizaje automático, así como del uso de las redes sociales. Específicamente, la rama de aprendizaje automático que será de interés para nuestro Trabajo Profesional es la de Clasificación.

El problema del Calificador automático de calidad de Imágenes puede dividirse en dos partes:

1. Obtención del set de entrenamiento para el clasificador.
2. Algoritmos a utilizar para realizar la clasificación.

2.1. Recolección de datos

Para entender este primer problema, es necesario considerar que buscamos determinar la calidad de una imagen en función de las 'consideraciones humanas', esto es, la calidad según como es percibida por los seres humanos. Necesitamos poder entrenar al algoritmo para que pueda detectar las características que son percibidas como importantes y determinan la calidad de una imagen.

Para ésto, podemos utilizar la información obtenible de las redes sociales: Dada una imagen en una red social, existen diversas formas para identificarlas como buenas (comentarios, valoraciones positivas, altos puntajes, etc..) o malas (ausencia de comentarios, valoraciones negativas, bajos puntajes). Tomando en cuenta estos parámetros, podemos determinar una calificación para dicha imagen a partir de una Heurística. Inclusive, hay redes sociales como **500px.com** que ya cuentan con algo similar a la heurística que planteamos⁴.

¹<https://www.flickr.com>

²<http://www.deviantart.com>

³<http://www.500px.com>

⁴Definido como el *pulse* (o *pulso*, en español) de la publicación correspondiente a la imagen.

La Heurística dependerá de la red social, puesto que cada una tiene distintos parámetros a considerar. Dicha Heurística será la que determine la calificación de la imagen, usando exclusivamente dichos parámetros (i.e. no se basa en lo absoluto en la imagen en sí, sino en la información adicional obtenida de la red social). Es posible incluso, que sea necesario utilizar algoritmos adicionales cuando la información de la red social sea escasa o no concluyente. Por ejemplo, en algunas redes sociales los comentarios suelen ser casi exclusivamente positivos, por lo que en estos casos la cantidad de comentarios debería ser un buen indicador; pero en otras redes sociales puede haber mayor variedad en este aspecto, habiendo tanto comentarios positivos como negativos. En estos casos será importante determinar si lo que buscamos es que la imagen no pase desapercibida (con lo cual, nuevamente la cantidad de comentarios es un indicador adecuado), mientras que en el otro caso, nos importará saber cuántos comentarios son positivos y cuántos negativos, por lo que será necesario utilizar un esquema de análisis de sentimiento.

La variación de calificación será determinada más adelante al realizar distintas pruebas, dependiendo de los resultados experimentales obtenidos, ya sea del 1 al 10, 1 a 5 estrellas o de algún otro tipo. Esta parte del proyecto no supone un desafío algorítmico, puesto que no es un problema difícil de analizar y comprender, pero sí un trabajo arduo siendo que será necesario analizar cómo implementar el programa que realice la descarga de datos y parámetros, así como la experimentación con distintas heurísticas para obtener buenos resultados. Además, es deseable obtener una variedad importante de cada clasificación (y relativamente pareja). Dicho trabajo debe realizarse por cada red social a utilizar. Se utilizarán las APIs que pudiesen brindar las distintas redes sociales (como es el caso de **Flickr**), para facilitar esta tarea.

2.2. Algoritmos para clasificación de imágenes

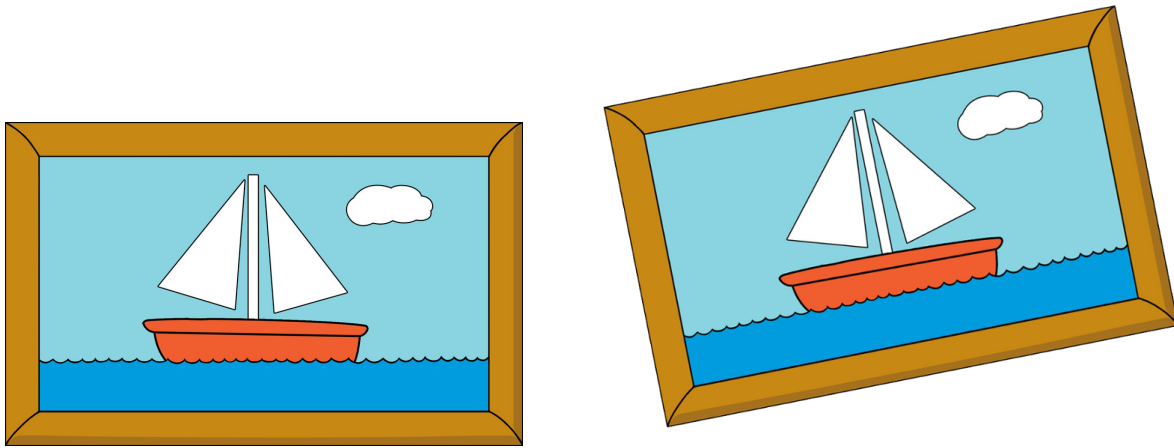
La segunda parte implica mucha investigación sobre los distintos algoritmos a utilizar para poder realizar la clasificación de una nueva imagen a partir del set de entrenamiento. Para esto, utilizaremos algoritmos de Deep Learning, que es el estado del arte hoy en día. Específicamente, para clasificación de imágenes lo más utilizado hoy en día es el uso de **Redes Neuronales de Convolución** (CNN, por sus siglas en inglés)[1][2], en conjunto con otros algoritmos para obtener características (o *features*) de las imágenes, puesto que la sola imagen en sí puede no ser suficiente para proporcionar una clasificación satisfactoria. Por ejemplo:



Se puede decir que las fotos son ciertamente distintas, y corresponden a cosas totalmente

distintas, por lo que nuestro algoritmo no puede basarse sólo en una cuestión de píxeles y colores de píxeles para determinar el parentesco de las imágenes (aunque tampoco se debe descartarlo, si se lo requiriere).

También puede suceder este otro caso:



A simple vista es evidente que ambas imágenes son iguales, con una diferencia en rotación, y podrían calificar casi idénticamente (aunque puede ocurrir que la imagen girada tenga mejor o peor votación por el solo hecho de estar girada), pero algorítmicamente no es sencillo ver esta similitud, puesto que la disposición de la imagen puede ser totalmente distinta, por lo que muchas veces se utilizan propiedades del espectro (en 2 dimensiones) de dichas imágenes.

El problema en sí está sumergido en un alto grado de subjetividad y percepción humana. Tal vez sí es verdad que todas las imágenes de color preponderante (como en el primer caso, el blanco) sean todas calificadas de la misma forma. Todo depende de las calificaciones que hayan realizado las personas, y el algoritmo debe aprender a razonar como ellas y, por ende, aprender qué *features* son importantes en una imagen para poder determinar su calificación (color, bordes, la aparición de rostros, formas, etc...), y ésto a su vez depende exclusivamente de la red social en cuestión, dado que no se espera el mismo comportamiento de los usuarios de Flickr que de los usuarios de Facebook⁵, por dar un ejemplo.

Una forma que tendremos de poder analizar el nivel de acierto del clasificador es utilizar imágenes ya existentes en las redes sociales, y verificar que los resultados del clasificador coincidan con los reales (analizados por la Heurística utilizada, a partir de los datos reales), y otra es predecir los resultados de una imagen inédita, y luego subirla a la red social, a la espera de los resultados reales.

Dicho algoritmo permitirá clasificar una nueva imagen a partir del set de entrenamiento, donde cada clase es un valor de calificación entre las calificaciones posibles.

Utilizaremos una librería de Deep Learning, desarrollada en Python[10], que está siendo desarrollada hasta hoy en día, pero que está teniendo muy buenos resultados: **Theano**[11].

3. Trabajos Relacionados

Las CNN han sido objeto de gran estudio en los últimos años dado los muy buenos resultados que se obtienen a la hora de realizar clasificación de imágenes, como en el clásico

⁵<http://www.facebook.com>

ejemplo del reconocimiento óptico de caracteres (OCR)[4]. Además, es notable como con unas mínimas variaciones en las imágenes, fácilmente implementables, se pueden obtener inclusive mucho mejores resultados (por ejemplo, rectificando y normalizando las imágenes).

Si bien hay trabajo previo sobre algoritmos que tratan de obtener una calificación automática de la calidad de una imagen en función de parámetros propios de la imagen[5][6] (como puede ser la distribución de color, iluminación, las formas, y otros parámetros), nosotros planteamos que la calidad de una imagen es dependiente del contexto en el que se presenta, y cómo es percibida por las personas que la observan en dicho contexto. En otras palabras, planteamos que la calidad no puede ser decidida haciendo uso únicamente de los atributos de la imagen, sino que el contexto en la que se encuentra juega un papel importante en la calidad percibida por los individuos que se mueven en dicho contexto, dando lugar a tantos niveles de calidad como contextos diferentes se presenten.

A su vez, también se han realizado estudios que proponen que seres humanos puntúen las imágenes del set de datos, para luego aplicar un algoritmo de clasificación de imágenes[7][8], lo cual es similar a lo que nosotros proponemos. Nuestra propuesta se diferencia en obtener los puntajes no manualmente, sino automáticamente, utilizando la información obtenible a partir de redes sociales, con la ventaja de poder disponer de un gigantesco set de datos, con información proveniente de seres humanos.

Otros trabajos reflejan importantes resultados utilizando **CNNs**, gracias a su gran capacidad de distinguir por sí mismas cuáles son los *features* más importantes, aunque para ello es necesario un set de datos por cada clase existente (en nuestro caso, las posibles calificaciones), para que la precisión sea lo más cercana al estado del arte[3]. A su vez, dichos trabajos discuten las posibles mejoras para resolver algunas pequeñas deficiencias que tienen las **CNNs** (por ejemplo, que todas las imágenes deben ser del mismo tamaño), utilizando dos **CNN** en paralelo, que permiten generar una muy buena precisión.

4. Objetivos

El Trabajo Profesional consta de 5 objetivos:

- Desarrollar el módulo que permita recolectar automáticamente el set de datos de cada red social a analizar, con las respectivas Heurísticas (en base a los parámetros de dicha red social) que determinen la calificación de cada imagen de dicho set de datos.
- Desarrollar el módulo que realice la calificación automática de una nueva imagen, a partir del set de entrenamiento obtenido.
- Realizar ejemplos que permitan contrastar los resultados (sean positivos o negativos) dentro de distintas redes sociales, y obtener conclusiones de los resultados.
- Aprender cuáles son los *features* que tienen mayor impacto en la calidad de una imagen.
- Integrar el algoritmo con herramientas existentes de código abierto de Aprendizaje Automático (*Machine Learning*).

4.1. Aprendiendo sobre los *features* importantes

Uno de los objetivos del proyecto es poder aprender cuáles son los *features* importantes (o relevantes) a la hora de determinar la calificación de la calidad de una imagen. Una gran ventaja de las **CNNs** sobre otros métodos de clasificación de imágenes es que no necesitan que se determine previamente cuáles son los *features* a utilizar, sino que la red aprende por sí misma cuáles son. Por lo tanto, nosotros podemos aprender del algoritmo cuáles son los *features* importantes, en vez de nosotros enseñarle al algoritmo.

¿Por qué resulta ésto de tanto interés? Podemos pensar al espacio de las imágenes como un enorme espacio, pero finito, dado que si tenemos imágenes de 128×128 píxeles, tenemos a lo sumo $128 \times 128 \times 255 \times 3 = 12533760$ imágenes posibles. Es claro que no todas las imágenes posibles tienen sentido, inclusive podemos decir que las *imágenes exitosas*, de gran calidad, son un subespacio muy pequeño dentro de dicho espacio de imágenes. Resulta de mucho interés aprender, y poder interpretar las características de las imágenes que permiten determinar si una imagen pertenece a dicho subespacio, por lo que podemos hablar directamente de conocer las características de dicho subespacio.

A su vez, es necesario tener en cuenta que el algoritmo propuesto estará utilizando los parámetros de las redes sociales, que son determinados por seres humanos. Como el público de cada red social suele ser distinto, o al menos los roles que cumplen dentro de ella difieren, es factible que una imagen tenga una determinada calidad en una red social, y otra totalmente distinta en otra red social. Básicamente, la calidad de la imagen no sólo dependerá de la imagen en sí, sino del contexto en el que se presenta, por lo que también resulta de mucho interés conocer cómo dicho contexto hace variar la importancia de los *features* de las imágenes.

4.2. Integración con herramientas Open Source

Otro objetivo del proyecto es el de realizar un aporte a la comunidad de código abierto, desde el área que abarcamos.

Como ya fue mencionado, se utilizará Theano como herramienta base de desarrollo, la cuál ya cuenta con un módulo de redes neuronales de convolución, pero aún se encuentra en desarrollo. Nuestra propuesta es integrar nuestro algoritmo de calificación de la calidad de imágenes, con las modificaciones y mejoras que se planteen durante el desarrollo del proyecto, como es el caso de un algoritmo planteado en RAPID[3] (utilizando dos columnas de **CNNs**).

Aún no se definió si dicha contribución se hará directamente sobre el proyecto **Theano**, algún *wrapper* especializado en redes neuronales, como **Lasagne**[14] (basado en Theano), o mediante la implementación de alguna solución independiente, similar a la anterior mencionada.

5. Características del Trabajo

5.1. Modulo de recolección de datos

El módulo de recolección de datos, o más correctamente **los** módulos, serán implementados cada uno exclusivamente para responder a una red social. Por cada red social utilizada, será necesario implementar un nuevo módulo que permita:

1. Obtener una cantidad determinada de links a imágenes.

2. Obtener los parámetros de cada una de las imágenes antes obtenidas.
3. Descargar el archivo de la imagen, propiamente dicho.
4. Obtener la calificación correspondiente a cada una de las imágenes.

Se puede ver que la obtención de parámetros de cada red social (y cuáles son los parámetros en sí) variará de red social a red social. En general, se tratará de utilizar las herramientas que nos brinde la red social para poder obtener todo lo antes mencionado, por ejemplo: una API que brinde la red social para realizar consultas.

Dado que además se deberán realizar una gran cantidad de consultas y descargas de dichos sitios, nos contactaremos con cada uno de ellos para obtener la habilitación correspondiente, para que no haya ningún conflicto relacionado con el proceso de obtención de datos.

5.2. Modulo de calificador automático

El módulo de calificación automática será basado en **CNN**, que son el estado del arte en lo que refiere a clasificación de imágenes, basado en técnicas conocidas de redes neuronales, así como de extracción de características (o *features*) de imágenes, así como la utilización del espectro de dichas imágenes, aplicándole filtros para obtener distintas características.

Para realizar dicho módulo, se utilizará principalmente Theano, una librería actualmente en desarrollo para realizar cálculos de manera eficiente, con módulos específicos para el desarrollo de aplicaciones que involucren técnicas de Deep Learning, como es el caso de las **CNNs**.

Este módulo será único para todas las redes sociales analizadas, por lo que cada uno de los módulos que recolecten información de cada red social debe tener un resultado común a todas (las imágenes y la calificación de cada una de ellas).

5.3. Resultados: Comparación de resultados obtenidos

Una vez terminado el módulo de calificación automática, y el módulo de recolección de datos de una red social, se pasará a comprobar sus resultados con nuevas imágenes que se acaben de subir a dicha red social, tanto como imágenes propias que vayamos a subir, y observar los resultados. Luego de transcurrido un tiempo, volveremos a utilizar el módulo de recolección de datos, pero esta vez para obtener la calificación real (basada en la heurística determinada), para comparar resultados y obtener conclusiones, así como realizar correcciones a los módulos en caso de ser necesario.

6. Tecnologías

6.1. Librería principal: Theano

Theano es una librería que comenzó a desarrollarse en 2007, en la Universidad de Montreal, y que aún se encuentra en sus primeros pasos de desarrollo (la versión 0.7 ha sido lanzada a fines de marzo del corriente año). Permite realizar cálculos numéricos, evaluación de expresiones matemáticas que puedan incluir arreglos multidimensionales, de una manera eficiente, tal que permite la utilización de GPUs⁶ para agilizar los cálculos.

⁶Unidades de Procesadores Gráficos.

Theano está pensada para realizar aplicaciones que involucren técnicas de aprendizaje automático, como así lo demuestran sus varios ejemplos en el sitio oficial.

El principal problema de Theano es su gran diversidad: está pensada para resolver problemas tan diversos, que es difícil saber cómo utilizarlo, y la curva de aprendizaje sigue siendo bastante alta, a pesar de tratarse de una librería realizada para un lenguaje como Python. Dado que nuestro objetivo es utilizar un set muy reducido dentro de Theano, es que utilizaremos otros desarrollos que se basan en Theano, pero brindan una interfaz mucho más sencilla, abocada exclusivamente a nuestras necesidades: **Lasagne**, una librería que se apoya totalmente en Theano, brindando una interfaz sencilla, y únicamente centrada en las operaciones relacionadas a redes neuronales.

6.2. Herramientas

Categoría	Herramientas
Lenguaje de Programación	Python
Entornos de desarrollo	Geany[15]
Control de Versiones	Git[16]
Administración y control del Proyecto	Google Docs[18]
Elaboración de Documentos	L ^A T _E X[17]
Librerías Principales	Theano Scipy[12] Numpy[13] Lasagne

7. Alcance

- Desarrollo de los módulos de recolección de datos (uno por cada red social analizada).
- Desarrollo del módulo de calificación automática de imágenes.
- Análisis de los resultados obtenidos, y modificaciones consecuentes para mejorar los resultados.
- Integración del módulo de clasificación a Theano o Lasagne (o alguna implementación semejante).

8. Plan de Trabajo

8.1. Equipo de Trabajo

- Tutor: Lic. Luis Algerich.
- Desarrolladores: Martín Buchwald y Ezequiel Genender Peña.

8.2. Metodología

Para la implementación del proyecto se llevará a cabo una metodología basada en SCRUM[19], para la cual se definirán una serie de iteraciones (con fechas a ser determinadas en las sucesivas reuniones con el tutor), con entregables a la finalización de cada una de ellas.

Al comienzo de cada iteración se analizarán las prioridades de los distintos requerimientos en conjunto con el tutor, luego estableciendo fechas para los entregables parciales (de ser considerados necesarios), y al finalizar la misma se realizará una reunión para la presentación del entregable de la iteración.

8.3. Estimación

A continuación se muestra un listado de tareas necesarias para alcanzar los objetivos del proyecto, con los esfuerzos estimados de los mismos. Los esfuerzos, en todos los casos, se encuentran expresados en horas.

Iteración	Descripción	Esf.	Rec.	Total
-	Propuesta del trabajo Profesional	20	2	40
1	Configuración del Entorno	5	2	10
	Diseño de los Módulos de recolección de datos	10	1	10
	Módulo de Recolección de datos de Flickr	20	2	40
	Heurística para las imágenes de Flickr	10	1	10
2	Módulo de Clasificación: Investigación y Análisis	70	2	140
	Módulo de Clasificación: Diseño	10	2	20
3	Módulo de Clasificación: implementación	40	2	80
	Integración entre el módulo de recolección y clasificación	10	1	10
4	Corroboración de resultados y correcciones al clasificador y heurística	20	2	40
5	Módulo de recolección de datos de 500px	10	1	10
	Integración con módulo de clasificación	5	2	10
	Corroboración de resultados y correcciones	10	2	20
6	Publicación y Experimentación con nuevas imágenes en Flickr y 500px	20	2	40
7	Módulos de recolección de datos para redes sociales adicionales e integración	25	2	50
8	Investigación de otros tipos de algoritmos para definir las heurísticas a utilizar, y su implementación.	20	2	40
9	Integración con Theano o Lasagne	25	2	50
-	Reuniones	20	2	40
-	Presentación	25	2	50

Además, se tiene en cuenta el tiempo dedicado a la administración del proyecto, el cual estimaremos alrededor del 10 % del tiempo de desarrollo. El resultado final de la estimación es:

Descripción	Esfuerzo
Iteración 1	70
Iteración 2	160
Iteración 3	90
Iteración 4	40
Iteración 5	40
Iteración 6	40
Iteración 7	50
Iteración 8	40
Iteración 9	50
Otros	130
Administración	70
Esfuerzo Total	780

8.4. Cronograma de entregables

A continuación se hace un cronograma tentativo de entregables al finalizar cada una de las iteraciones. Las mismas pueden ser modificadas en caso de verse necesario, por el tutor o por los desarrolladores del proyecto. La fecha de cada una de las entregas serán determinadas en conjunto con el tutor.

Iteración	Entregables	Hitos
1	<ul style="list-style-type: none"> • Módulo recolección de datos Flickr. • Set de datos de ejemplo. 	<ul style="list-style-type: none"> • Configuración del entorno terminada. • Diseño de módulos de recolección de datos. • Implementación de un módulo de recolección de datos.
2	<ul style="list-style-type: none"> • Implementación básica de módulo de clasificación. 	<ul style="list-style-type: none"> • Análisis y Diseño completos del módulo de clasificación.
3	<ul style="list-style-type: none"> • Módulo de clasificación. 	<ul style="list-style-type: none"> • Implementación del módulo terminada.
4	<ul style="list-style-type: none"> • Resultados del módulo para la red social Flickr. • Modificaciones a los módulos anteriores. 	<ul style="list-style-type: none"> • Integración entre módulos de recolección y clasificación terminada.
5	<ul style="list-style-type: none"> • Módulo de recolección para red social adicional. 	<ul style="list-style-type: none"> • Implementación terminada del nuevo módulo de recolección. • Integración de los módulos adicionales de recolección y clasificación.
6	<ul style="list-style-type: none"> • Informe de resultados obtenidos. 	<ul style="list-style-type: none"> • Publicación experimental de fotos en redes sociales. • Obtención e interpretación de resultados.
7	<ul style="list-style-type: none"> • Módulo de recolección para otras redes sociales. 	<ul style="list-style-type: none"> • Implementación de cada uno de los módulos de recolección de datos adicionales. • Integración de cada uno de los módulos adicionales con módulos de clasificación. • Corrección de errores aplicada.
8	<ul style="list-style-type: none"> • Mejoras a los módulos de recolecciones de datos 	<ul style="list-style-type: none"> • Implementación terminada de los distintos algoritmos que permitan mejorar la forma de asignar la calificación de una imagen.
9	<ul style="list-style-type: none"> • Versión de Theano (o Lasagne) con módulo de calificación de imágenes integrado. 	<ul style="list-style-type: none"> • Integración con Theano (o Lasagne) terminada.

Referencias

- [1] *Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position*. Kunihiko Fukushima.
- [2] *Gradient-Based Learning Applied to Document Recognition*. Yann Lecun, Léon Bottou, Yoshua Bengio, Patrick Haffner.
- [3] *RAPID: Rating Pictorial Aesthetics using Deep Learning*. Xin Lu, Zhe Lin, Hailin Jun, Jianchao Yang, James Wang. The Pennsylvania State University y Adobe Research.
- [4] *A Deep Learning Approach to Document Image Quality Assessment*. Le Kang, Peng Ye, Yi Li, David Doerman. Universidad de Maryland, USA, y NICTA and ANU, Australia.
- [5] *Color Image Quality Assessment Based on Image Quality Parameters Perceived by Human Vision System*. Zhen-Xiang Xie, Zhi-Fang Wang. Chongqing Medical University.
- [6] *Subjective Image Quality Assessment based on Objective Image Quality Measurement Factors*. Hyung-Ju Park, Dong-Hwan Har. IEEE.
- [7] *Image Retargeting Quality Assessment: A Study of Subjective Scores and Objective Metrics*. Lin Ma, Weisi Lin, Chenwei Deng, King Ngi Ngan. IEEE.
- [8] *A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms*. Hamid Rahim Sheikh, Muhammad Farooq Sabir, Alan Conrad Bovik. IEEE.
- [9] *Deep Learning*. Página de noticias y recolección de información sobre software relacionado a Machine Learning.
<http://www.deeplearning.net/>
- [10] *Python*. Lenguaje de programación interpretado.
<http://www.python.org/>
- [11] *Theano*. Librería de Python, basada en NumPy y Scipy, para el desarrollo de algoritmos para problemas de Machine Learning, desarrollada originalmente en la Universidad de Montreal.
<http://www.deeplearning.net/software/theano/>
- [12] *Scipy*. Paquete de Python para computación científica.
<http://www.scipy.org/>
- [13] *Numpy*. Paquete de Python para cálculo y procesamiento numérico.
<http://www.numpy.org/>
- [14] *Lasagne*. Librería para manejo y entrenamiento de redes neuronales en Theano, de manera simplificada.
<http://lasagne.readthedocs.org/>
- [15] *Geany*. Editor de texto.
<http://www.geany.org/>

- [16] *Git*. Sistema de control de versiones.
<http://git-scm.com>
- [17] \LaTeX . Sistema de composición de textos de alta calidad.
www.latex-project.org/
- [18] *Google Docs*. Herramienta para edición de archivos online.
<https://docs.google.com>
- [19] *SCRUM*. Metodología de desarrollo de Software, con ciclo de vida iterativo.
<http://scrummethodology.com/>