

UNIVERSIDAD NACIONAL DE SAN AGUSTIN DE AREQUIPA
ESCUELA DE POSGRADO
UNIDAD DE POSGRADO DE LA FACULTAD DE INGENIERIA DE
PRODUCCION Y SERVICIOS



SISTEMA RECOMENDADOR DE OBJETOS DE APRENDIZAJE, BASADO
EN LA METODOLOGÍA DE DEEP LEARNING, PARA EL
RECONOCIMIENTO DE ESTILOS DE APRENDIZAJE QUE MEJOREN EL
DESEMPEÑO DE LOS ESTUDIANTES EN LA EDUCACIÓN BÁSICA
REGULAR (EBR 2017)

Tesis presentada por el Bachiller:

Torres Aguilar, Ney

Para optar el Grado Académico de Maestro en
Ciencias: Informática, con mención en
Tecnologías de la Información y
Comunicación en Gestión y Educación.

ASESOR: Mg. Juan Carlos Gutiérrez Cáceres

AREQUIPA-PERU

2021

AGRADECIMIENTO

Al Mg. Juan Carlos Gutiérrez Cáceres, por su esfuerzo y dedicación, quien, con sus conocimientos, su experiencia, su paciencia y su motivación ha logrado en mí perfilarme en el campo de la investigación del *Deep Learning*.

DEDICATORIAS

A mi Angelito Martin y Felipito que están en el cielo, a mi esposa por su apoyo, amor y comprensión, de la misma forma a mi Madre por darme el ejemplo de la perseverancia constante.

RESUMEN

En la educación moderna es muy importante diferenciar el estilo de aprendizaje para tratar de mejorar la enseñanza y en consecuencia el aprendizaje, al presentar información adecuada al estilo de aprendizaje de los estudiantes. Por esta razón este trabajo propone utilizar una red *DENSENET* como sistema de recomendación de estilo de aprendizaje, se entrena un modelo con un conjunto de datos de 500 imágenes de dibujos de paisajes. El sistema es capaz de predecir estilos de aprendizaje. Esta propuesta, alcanzó una precisión del 87,77% que es buena para orientar a los estudiantes de secundaria en el proceso de aprendizaje.

ABSTRACT

In modern education it is very important to differentiate the learning style to try to improve teaching and consequently learning, at presenting appropriate information for the learning style of the students. For this reason this work proposes uses a DenseNet as a recommender system of learning style, a model is trained with a dataset of 500 images of landscapes drawings. The system is capable of predicted learning styles. This proposed, reached an accuracy of 87.77% that is good to guide students in high school in the learning process.

TABLA DE CONTENIDO

| | |
|---|------|
| AGRADECIMIENTO | i |
| DEDICATORIAS..... | ii |
| RESUMEN | iii |
| ABSTRACT | iv |
| ÍNDICE DE FIGURA | ix |
| ÍNDICE DE TABLA | xiii |
| GLOSARIO DE TÉRMINOS: | xiv |
| CAPÍTULO I: PLANTEAMIENTO METODOLÓGICO | 1 |
| Introducción: | 1 |
| 1.1. Descripción del problema | 2 |
| Formulación del problema | 2 |
| 1.2. Objetivo de la investigación..... | 3 |
| Objetivos específicos | 3 |
| Variables e indicadores | 3 |
| Hipótesis | 3 |
| 1.3. Justificación | 3 |
| Estructura del documento | 5 |
| CAPÍTULO II: MARCO TEORICO..... | 6 |
| 2.1. Consideraciones iniciales..... | 6 |
| 2.2. Red neuronal: | 6 |
| 2.3. Modelo de una Neurona:..... | 7 |
| 2.3.1. Elementos básicos que componen una red neuronal: | 7 |
| 2.4 Deep Learning: | 9 |
| 2.4.1 Taxonomía: | 13 |

| | |
|---|----|
| 2.5. Redes Convolucionales..... | 14 |
| 2.5.1. Capa convolucional: | 16 |
| 2.5.2. Capa de reducción o pooling: | 17 |
| 2.5.3. Capa clasificadora totalmente conectada:..... | 18 |
| 2.6. Convolución:..... | 19 |
| 2.6.1. Sigmoides:..... | 20 |
| 2.7.2. ReLU (Rectified Lineal Unit)..... | 21 |
| 2.8. Max-Pooling | 21 |
| 2.9. Data augmentation | 24 |
| 2.9.1. Aprendizaje profundo para el aumento de datos: | 26 |
| 2.10 ResNet..... | 27 |
| 2.10.1. Características ResNet: | 29 |
| 2.11. Redes convolucionales densamente conectadas (DENSELY) | 30 |
| 2.11.1. Conectividad densa. | 30 |
| 2.11.2. Función compuesta. | 31 |
| 2.11.3. Agrupación de capas. | 31 |
| 2.11.4. Tasa de crecimiento. | 31 |
| 2.11.5. Capas de cuello de botella. | 31 |
| 2.11.6. Compresión..... | 31 |
| 2.12 Métricas de experimentación | 31 |
| 2.12.1 Matriz de confusión | 31 |
| 2.12.2 Accuracy (precisión)..... | 31 |
| 2.12.3 Loss (pérdida) | 32 |
| 2.12.4 Precisión..... | 32 |
| 2.12.5 Recuperación (Recall)..... | 32 |
| 2.13. Sistema Recomendador (SR)..... | 33 |
| 2.14. Objetos de Aprendizaje OA | 34 |

| | |
|---|-----------|
| 2.14.1. Características de los Objetos de aprendizaje..... | 35 |
| 2.14.2. Modelos de Metadatos | 37 |
| 2.14.2.1 Estándares de Metadatos..... | 37 |
| 2.14.3 Modelos de Empaquetamiento..... | 37 |
| 2.14.3.1. Modelos de empaquetamiento de OA..... | 38 |
| 2.15. Estilos de aprendizajes..... | 39 |
| 2.16.1 Modelos de estilos de aprendizaje | 39 |
| 2.16.2 El Modelo de Aprendizaje Experiencia de Kolb: | 40 |
| 2.25.2.1 Criterios de valoración del test LSI I de Kolb | 42 |
| CAPÍTULO III: DIAGNÓSTICO SITUACIONAL | 45 |
| 3.1 Trabajos que reconocen textos a mano | 45 |
| 3.1.1. Primero: obtención de la “espinas dorsal de las líneas del texto”, que consta:..... | 45 |
| 3.1.2. Segundo: Separación de caracteres conflictivos..... | 47 |
| 3.2 Trabajos que reconocen Objeto dibujo | 48 |
| 3.2.1. Estructura de la red | 48 |
| 3.2.2. Experimentos y resultados | 49 |
| 3.3 Trabajos que reconocen paisajes dibujados a mano | 50 |
| 3.3.1. Metodología..... | 51 |
| 3.3.2. Funciones novedosas en la arquitectura Scene-Net..... | 52 |
| 3.3.2.1. Tamaño de entrada más pequeño..... | 53 |
| 3.3.2.2. Max Pooling superpuesta..... | 53 |
| 3.3.2.3. Afinación Fina. | 53 |
| 3.4 Trabajo para generar estrategias de enseñanza y motivaciones que estimulen el aprendizaje..... | 54 |
| 3.5 Análisis de trabajos relacionados..... | 55 |
| 3.6 Consideraciones finales | 57 |

| | |
|---|----|
| CAPÍTULO IV: PROPUESTA PLANTEADA | 59 |
| 4.1 Propuesta educativa | 59 |
| 4.2 Propuesta computacional | 61 |
| 4.2.1 Etapa experimental: | 62 |
| 4.2.2 Método para el reconocimiento de Estilos de Aprendizaje de imágenes de paisajes | 65 |
| 4.2.2.1. Pre-procesamiento de imágenes: | 65 |
| 4.2.2.2. Extracción de características:..... | 65 |
| 4.2.2.3. Clasificación | 66 |
| 4.2.2.4. Evaluación: | 66 |
| 4.3 Etapa de construcción de la propuesta | 66 |
| 4.3.1 Implementación de la propuesta | 67 |
| CAPÍTULO V: PRUEBAS Y RESULTADOS | 68 |
| 5.1 Base de datos (dataset)..... | 68 |
| 5.1.1 Conjunto de datos de estilos de aprendizaje | 68 |
| 5.2 Pruebas..... | 70 |
| 5.3 Resultados..... | 71 |
| 5.4 Evaluación de la eficiencia | 76 |
| 5.5 Decrecimiento de los modelos..... | 76 |
| 5.6 Análisis de los resultados..... | 77 |
| CONCLUSIONES Y RECOMENDACIONES | 79 |
| Conclusiones | 79 |
| Recomendaciones | 80 |
| BIBLIOGRAFÍA | 81 |

ÍNDICE DE FIGURA

| | |
|--|----|
| Figura 2.1: Red neuronal de una única neurona y una entrada (Durán Suárez, 2017). | 7 |
| Figura 2.2: ejemplo de una red neuronal totalmente conectada (Matich, 2001) | 8 |
| Figura 2.3: Modelo no lineal de una neurona. (Ocampo, 2014)..... | 8 |
| Figura 2.4. Imágenes de muestra de las clases base de 10 dígitos MNIST (Ponti & Paranhos da Costa, 2017) | 11 |
| Figura 2.5: a) un único valor x_1 como entrada, b) vector x como entrada (Ponti & Paranhos da Costa, 2017) | 12 |
| Figura 2.6: Una arquitectura profunda con dos capas ocultas, generando representaciones intermediarios que preceden a la clasificación. (Ponti & Paranhos da Costa, 2017)..... | 13 |
| Figura 2.7. Taxonomía del deep learning a partir de conceptos de RNA (Restrepo Arteaga, 2015) | 13 |
| Figura 2.8: Taxonomía del deep learning a partir del tipo de algoritmo de aprendizaje (Restrepo Arteaga, 2015)..... | 14 |
| Figura 2.9: Filtro convolución 2x2 (Thomas, 2018)..... | 15 |
| Figura 2.10: Estructura- fuente: http://www.diegocalvo.es/red-neuronal-convolucional/ | 15 |
| Figura 2.11: Fuente: http://www.diegocalvo.es/red-neuronal-convolucional | 16 |
| Figura 2.12: Obtención de la matriz de activación. (Durán Suárez, 2017)..... | 17 |
| Figura 2.13: Convolución a una imagen original a una versión comprimida..... | 17 |
| Figura 2.14: Resultado de aplicar diferentes tipos de pooling. (Durán Suárez, 2017) | 18 |
| Figura 2.15: Fully convolutional networks (Loncomilla, 2016)..... | 19 |
| Figura 2.16: Sliding window (Ventana deslizante) | 19 |
| Figura 2.17: función de activación sigmoide..... | 20 |
| Figura 2.18: Función de activación ReLU | 21 |
| Figura 2.19: Ejemplo visual (con algo de relleno) de este tutorial de red convolucional TensorFlow (Thomas, 2018)..... | 22 |
| Figura 2.20: Fuente: tesis doctoral (Shankar, 2017)..... | 23 |
| Figura 2.21: Convolucionales para detectar el dígito 9 en una imagen..... | 23 |

| | |
|---|----|
| Figura 2.22: Operación de agrupación máxima en a) ejemplos pictóricos y b) basados en imágenes usando una agrupación tamaño de 2x2 y zancada de 2. (Salumaa, 2018). | 24 |
| Figura 2.23: Aumento de datos en juego | 25 |
| Figura 2.24: Fuente: Aumento de datos mediante la transformación de preservación semántica para SBIR. | 26 |
| Figura 2.25: Fuente: Artistas e Inteligencia Artificial.nte | 26 |
| Figura 2.26: Fuente UC Berkeley | 27 |
| Figura 2.27: El módulo residual en ResNet (Rosebrock, 2017) | 28 |
| Figura 2.28:(a) El módulo residual original. (b) El módulo residual actualizado usando pre-activación. | 28 |
| Figura 2.29: Red residual (Yeung, 2017) | 29 |
| Figura 2.30: Capas de red para ajustar una asignación residual (Yeung, 2017)..... | 29 |
| Figura 2.31: Arquitectura completa de ResNet (Yeung, 2017) | 30 |
| Figura 2.32: Una red densamente convolucional con una tasa de crecimiento de $k = 4$. Cada capa toma todos los mapas de características anteriores como entrada (Huang, 2017)..... | 30 |
| Figura. 2.33: Bloques densos donde se muestran las capas de transición y cambian los tamaños de los mapas de características mediante convolución y agrupación (Huang, 2017)..... | 31 |
| Figura 2.34: Estructura propuesta del Objeto de Aprendizaje (Cué, 2017)..... | 35 |
| Figura 2.35: Esquema de las capas y las iniciativas más importantes para llegar a una interoperabilidad de contenidos en e-learning | 37 |
| Figura 2.36: Estilos de aprendizaje. Fuente: (Romero, 2010) | 42 |
| Figura 2.37: Cuadrantes de fases | 43 |
| Figura 2.38: Estilo de aprendizaje según puntuación AC-CE y AE-RO..... | 44 |
| Figura 3.1: De arriba abajo: FCN-pool2, FCN-pool3 y FCN-pool4 | 46 |
| Figura 3.2: Representación de los resultados después de cada paso. | 47 |
| Figura 3.3: Ejemplo de caracteres conflictivos..... | 47 |
| Figura 3.4: LAG de tiradas verticales (grafo rojo) | 48 |
| Figura 3.5: Estructura de CNN de dos ramas y diagrama de esquema del modelo de aprendizaje de migración..... | 48 |
| Figura 3.6: Ejemplos de bocetos de escena de nuestro conjunto de datos Scene250 (un ejemplo por categoría)..... | 50 |

| | |
|---|----|
| Figura 3.7: Ejemplos para demostrar los desafíos del reconocimiento de bocetos de escenas. | 51 |
| Figura 3.8. Arquitectura general del modelo Scene-Net. | 51 |
| Figura 3.9: Frecuencia por Estilo de Aprendizaje con base en el Modelo de Kolb de los estudiantes de Diseño Gráfico de FADLUZ. (Ortega, 2019) | 55 |
| Figura 4.1. Test completado por un estudiante de la muestra (Inventario de Estilos de aprendizaje de Kolb)..... | 59 |
| Figura 4.2. Dibujo de una casa en un paisaje, por el estudiante que completo el Test. | 59 |
| Figura 4.3. Datos generada por Google Forms para obtener las puntuaciones de los cuatro estilos de aprendizaje..... | 60 |
| Figura 4.4. Resultados del Test de Kolb aplicados a los estudiantes de secundaria. . | 60 |
| Fuente: Elaboración propia a partir de datos obtenidos en la investigación..... | 60 |
| Figura 4.5. Metodología empleada | 61 |
| Figura 4.6. Ejemplos de tareas en la etapa del pre-procesamiento de datos..... | 63 |
| Figura 4.7. Arquitectura general de CNN para la extracción de características de una imagen | 63 |
| Figura 4.8. Marco de trabajo para la clasificación..... | 64 |
| Figura 4.9. Marco de trabajo para la evaluación..... | 64 |
| Figura 4.10. Marco de trabajo usado para el pre-procesamiento de imágenes en el reconocimiento de expresiones faciales | 65 |
| Figura 4.11. Arquitectura DenseNet https://github.com/karthikziffer/Keras_Densenet_CIFAR10_ | 66 |
| Figura 4.12: Propuesta de investigación: La red Densenet se utiliza para aprender estilos de aprendizaje a partir de una base de datos de bocetos de estudiantes. | 67 |
| Figura 5.1: Ejemplos de la base de datos de los dos estilos donde (a) Grupo RETAILER (Divergente-Asimilador)y (b) Grupo SIMPLISTIC (Convergente-Acomodador). | 69 |
| Figura 5.2: Evidencia de la deducción del modelo entrenado, dando da como resultado grupo de imágenes de estilo Detallista. | 70 |
| Figura 5.3: Evidencia de la deducción del modelo entrenado, dando da como resultado grupo de imágenes de estilo Simplista..... | 71 |

| | |
|--|----|
| Figura 5.04: Resultado de precisión de Densenet de la evolución del equipo de prueba desde el proceso de entrenamiento. | 78 |
|--|----|

ÍNDICE DE TABLA

| | |
|---|----|
| Tabla 1.1 Variables e indicadores | 3 |
| Tabla 2.1 Esquema de la matriz de confusión | 31 |
| Tabla 2.2 Integración de modelos de estilos de aprendizaje..... | 40 |
| Tabla 2.3 Los estilos de aprender y sus características generales | 41 |
| Tabla 2.4 Ítems test de Kolb según grupo y dimensiones del modelo..... | 42 |
| Tabla 2.5 Diferencias de puntuaciones | 43 |
| Tabla 3.1. Comparación de la precisión media (MAP) | 49 |
| Tabla 3.1 Arquitectura detallada de Scene-Net | 52 |
| Tabla 3.3. comparativa de trabajos relacionados con la investigación..... | 55 |
| Tabla 5.1 Distribución | 69 |
| Tabla 5.2 Configuración de los modelos | 72 |
| Tabla 5.3 Configuración de los modelos | 73 |
| Tabla 5.4 Configuración de los modelos | 74 |
| Tabla 5.5 Configuración de los modelos | 74 |
| Tabla 5.6 Resultados del proceso de entrenamiento de las redes | 75 |
| Tabla 5.7 Decrecimiento de ConvNet..... | 77 |
| Tabla 5.8 Crecimiento y decrecimiento de DenseNet | 77 |
| Tabla 5.9 Crecimiento de ConvNet | 78 |

GLOSARIO DE TÉRMINOS:

OA: Objetos de aprendizaje

DL: Deep learning

SR: Sistema Recomendador

CNN: Convolutional Neural Networks

MLP: Perceptron multicapas

RNA: Redes neuronales artificiales

DenseNet: Dense Convolutional Network

Deep Learning: Aprendizaje profundo

Machine Learning: Aprendizaje automático por parte de máquinas

High School: Escuela secundaria

Simplistic: Simplista

Retailer: Detallista

CAPÍTULO I: PLANTEAMIENTO METODOLÓGICO

INTRODUCCIÓN:

Actualmente, existe una preocupación por una formación personalizada que adapte un subconjunto de rasgos que determinan las características del entorno de enseñanza, como, por ejemplo, adaptar la presentación según el estilo de aprendizaje del alumno. Los distintos estilos de aprendizaje se refieren a las diferencias entre la forma en que la información se transforma en conocimiento. Por ello, se puede intentar mejorar la enseñanza y consecuentemente el aprendizaje presentando la información de la forma adecuada al estilo de aprendizaje de los estudiantes (León O. A., 2016).

Ante esta realidad es necesario desarrollar un sistema adaptativo que tenga en cuenta diversos factores, y tomar decisiones que afectarán al sistema, antes de proceder a diseñar la estructura de la información relacionada con las actividades a realizar por los estudiantes (por ejemplo los objetos de aprendizaje), así como las posibles relaciones entre ellas, los diferentes contenidos que serán ofrecidos a cada tipo de estudiante en cada actividad, y los vínculos entre actividades y contenidos (Schilardi, León, Segura, & Polenta, 2016), que mejoren los desempeños de los estudiantes en las diferentes áreas de la educación básica regular.

Las infinitas aplicaciones científicas de los algoritmos de *Deep Learning* (aprendizaje profundo) permiten que los modelos computacionales compuestos de múltiples capas de procesamiento aprendan representaciones de datos con múltiples niveles de abstracción, estos métodos han mejorado dramáticamente el estado de arte en reconocimiento de voz, reconocimiento visual de objetos, detección de objetos y muchos otros dominios como el descubrimiento de fármacos y la genómica (LeCun, 2015).

Dada la potencialidad de aplicaciones de la metodología Deep Learning (aprendizaje profundo) en diferentes campos, podría implementarse en el ámbito educativo, integrándose en los sistemas de recomendación para la detección de estilos de aprendizaje...a partir de bocetos o dibujos a mano alzada, como complemento a la selección de recursos didácticos que potencian las competencias de los estudiantes. y

fomentar actitudes positivas en el área, permitiendo así el logro esperado de aprendizajes, y la reducción significativa de actitudes negativas de los estudiantes.

La principal motivación de este proyecto, es la detección de los estilos de aprendizaje, mediante la implementación de Dense Convolutional Network (DenseNet) (Huang, 2017), a partir de bocetos o dibujos a mano alzada, en sistemas recomendadores de objetos de aprendizaje, que han tomado gran importancia en los procesos de enseñanza aprendizaje; siendo una herramienta que aportan a la actividad docente y el aprendizaje de los estudiantes.

1.1. DESCRIPCIÓN DEL PROBLEMA

Luego de realizar el proceso de observación referente al desarrollo de las competencias en el *High School* en la planificación de los procesos de enseñanza aprendizaje la gran mayoría de docentes realizan una educación intransigente y no aprovechan sus talentos, desconoce el nivel y ritmo de aprendizaje de sus estudiantes e imparten las mismas tareas, con los mismos recursos sin descartar los conocimientos para todos, ni se preocupan en establecer estrategias psicopedagógicas para desarrollar sus competencias en las diferentes áreas curriculares .

Por todo lo anterior se hace necesario implementar un sistema de apoyo a la actividad pedagógica del docente capaz de recomendar objetos de aprendizaje personalizados que mejoren los desempeños de los estudiantes en las áreas de ciencias del nivel secundario.

FORMULACIÓN DEL PROBLEMA

Si se implementa un sistema recomendador de objetos de aprendizaje basado en la metodología de *deep learning*, entonces se podrá facilitar el reconocimiento de estilos de aprendizaje que mejoren el desempeño de los estudiantes de la educación básica regular del *High School*.

¿Qué elementos deben tenerse en cuenta, en una propuesta basada en un sistema recomendador de objetos de aprendizaje basado en *deep learning*, para el reconocimiento de estilos de aprendizaje que mejoren el desempeño de los estudiantes de la educación básica regular?

1.2. OBJETIVO DE LA INVESTIGACIÓN

Proponer un sistema recomendador de objetos de aprendizaje, basado en *deep learning*, para la selección óptima de estilos de aprendizaje que mejore el desempeño de los estudiantes en la educación básica regular del nivel secundario.

OBJETIVOS ESPECÍFICOS

- Identificar la metodología de estilos de aprendizajes para implementar objetos de aprendizaje que mejoren el desempeño en las diferentes áreas.
- Analizar arquitecturas de *deep learning* para definir los estilos de aprendizajes.
- Implementar la arquitectura Dense Convolutional Network (DenseNet)
- Realizar las pruebas y resultados.

VARIABLES E INDICADORES

Sistema recomendador de objetos de aprendizaje basado en *deep learning*, que mejoren el desempeño de los estudiantes según sus estilos de aprendizaje.

Tabla 1.1 Variables e indicadores

| Variable | Dimensiones | Indicadores | Técnicas/Instrumentos |
|---|-------------------------------|--|---------------------------------|
| V.I.: Sistema recomendador de objetos-aprendizaje | Arquitectura de deep learning | Arquitecturas de Red (CNN, ResNet, DenseNet) | Modelo de red DenseNet |
| V.D.: Desempeño de los estudiantes. | Estilos de aprendizajes | Estilo detallista y simplista | Test de estilos de aprendizaje. |

HIPÓTESIS

Es posible proponer un sistema recomendador de objetos de aprendizaje basado en *deep learning*, para reconocer los estilos de aprendizaje.

1.3. JUSTIFICACIÓN

Los sistemas informáticos de apoyo a los docentes para gestionar, impartir y optimizar la enseñanza de forma interactiva a los estudiantes, ayuda en el proceso de enseñanza-aprendizaje. (Canales Velis, 2016), se han desarrollado sistemas de clasificación de

recursos educativos basado en el perfil del usuario, tomando en cuenta estilos de aprendizajes y las competencias que el estudiante debe lograr en la educación media superior (Bautista, 2015).

Actualmente en las instituciones educativas se observa grandes problemas de aprendizaje en los estudiantes, que impiden a los docentes realizar una buena enseñanza en las diferentes áreas curriculares. Si se pudiera implementar sistemas informáticos de apoyo para la enseñanza, que integren los estilos aprendizaje se precisaría la evidencia de desempeño de los estudiantes, garantizando la calidad de los aprendizajes.

Machine deep learning technique (técnica de aprendizaje de máquinas), define redes neuronales basadas en el reconocimiento de patrones de los datos de entrada y lograr resultados con un nivel de confianza alto y eficiente en el tiempo (Contreras, 2016), el éxito de los métodos de *deep learning* está en la composición de funciones, que realizan transformaciones sucesivas a partir de vector de entrada. (Ponti M. y., 2018), entre las principales aplicaciones de *deep learning* se encuentra la mejor comprensión de enfermedades y terapias genéticas, análisis de imágenes médicas, como radiográficas y resonancia magnéticas, aumentando la precisión de diagnósticos, en un menor tiempo y con un menor coste que los métodos tradicionales, exploración de la posibilidad de reutilización de fármacos ya conocidos y probados para su uso contra nuevas enfermedades, identificación en textos de sentimientos positivos y negativos, temas, palabras claves, localización de caras e identificación de emociones faciales (Ávila, 2018).

En este trabajo se propone aplicar métodos de *Deep Learning* (aprendizaje profundo) que buscan descubrir un modelo implementado a través de reglas y parámetros utilizando un conjunto de datos, un método para guiar el aprendizaje del modelo. Al final de este proceso de aprendizaje, se genera como salida una función capaz de recibir nuevos datos de entrada y proporcionar una representación adecuada del problema en cuestión (Ponti & da Costa, 2017). En esta investigación, se describirán las redes neuronales profundas, particularmente las *Convolutional Neural Networks* (redes neuronales convolucionales) CNN. Cuyo propósito es extraer todas las características de una imagen y luego utilizar esas características para detectar o clasificar los objetos en una imagen (Massiris, 2018).

Recientemente, el reconocimiento de bocetos ha sido un área de investigación muy popular debido a sus amplias aplicaciones. Sin embargo, no existe un punto de referencia previo para el bosquejo de la escena ni un trabajo de investigación relacionado con la comprensión del bosquejo de la escena compleja. Los trabajos de investigaciones anteriores se centran en el reconocimiento de bocetos de un solo objeto y emplea esquemas tradicionales de orientación fotográfica para esta tarea (Ye, 2016).

Recientemente, con la popularidad de la CNN profundas, se propone implementar el modelo CNN profundo y avanzado para la detección de los estilos de aprendizaje, mediante la aplicación de Dense Convolutional Network (DenseNet) (Huang, 2017), a partir de bocetos o dibujos a mano alzada de bocetos de escena o paisaje libre, de estudiantes de *high school*, en sistemas recomendadores de objetos de aprendizaje, que han tomado gran importancia en los procesos de enseñanza aprendizaje; siendo una herramienta que aportan a la actividad docente y el aprendizaje de los estudiantes.

ESTRUCTURA DEL DOCUMENTO

Para abordar la temática, en este trabajo de investigación se estructura en cinco capítulos, donde se realiza una introducción al tema de investigación, en el capítulo I: se analiza el problema hallado, el contexto y la motivación de investigación para plantear el objetivo de la investigación y las hipótesis correspondientes. El capítulo II el marco teórico, donde se contempla los antecedentes del estudio previo a la investigación, de igual manera se puntualizan algunas consideraciones teóricas con una reflexión crítica y el capítulo III se realizará el diagnóstico del contexto de la investigación realizada. En el capítulo IV se propone la metodología que será empleada para el desarrollo de la investigación, en el capítulo V se validará la propuesta y los resultados de la investigación y en el capítulo VI se evaluará la propuesta según las ventajas y desventajas.

CAPÍTULO II: MARCO TEORICO

2.1. CONSIDERACIONES INICIALES

Debido a su constitución y a sus fundamentos, las redes neuronales artificiales presentan un gran número de características semejantes a la del cerebro, en el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan ciertos resultados específicos. Una red neuronal no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de pesos en los enlaces mediante el aprendizaje. También existen redes que continúan aprendiendo a lo largo de su vida, después de completado su período de entrenamiento.

Existen muchos enfoques bien establecidos para el aprendizaje automático por parte de máquinas (Machine Learning), cada uno de ellos utiliza una estructura algorítmica diferente para optimizar las predicciones basadas en los datos recibidos. Machine Learning es un amplio campo con una compleja taxonomía de algoritmos que se agrupan, en general, en tres grandes categorías: aprendizaje supervisado, aprendizaje no supervisado y reinforcement Learning.

Caso concreto de Deep Learning su estructura algorítmica está compuesto de múltiples capas de procesamiento para aprender representaciones de dato, con múltiples niveles de abstracción que realizan una serie de transformaciones lineales y no lineales que a partir de los datos de entrada generen que una salida próxima a la esperada.

2.2. RED NEURONAL:

Sistema computacional compuesto por un gran número de elementos simples, elementos de procesos muy interconectados, las cuales procesan información por medio de su estado dinámico como respuesta a entradas externas.

De la forma que están constituidos presentan un gran número de características semejantes a las del cerebro. Son capaces de aprender a la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que presentan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas.

La capacidad de aprendizaje adaptativo permite aprender a llevar acabo ciertas tareas mediante entrenamientos con ejemplos ilustrativos, son dinámicos, porque son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones. (Matich, 2001)

2.3. MODELO DE UNA NEURONA:

Se comenzará describiendo la red neuronal más simple que existen, para explicar el funcionamiento de las redes neuronales; una red formada por una única neurona y una única entrada ver la Figura 2.1. (Durán Suárez, 2017)

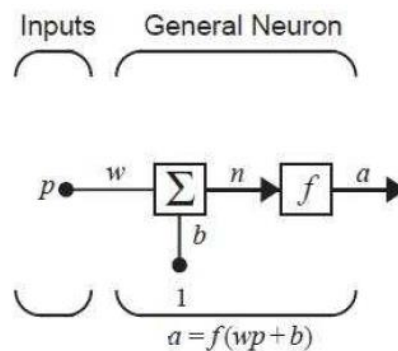


Figura 2.1: Red neuronal de una única neurona y una entrada (Durán Suárez, 2017).

Entradas: $p \cdot w ; 1 \cdot b$

p : *escalar*

w : *peso*

1 : *valor*

b : *bias*

Salida de la función de transferencia: se tiene la salida “a” de la neurona:

$$a = f(wp + b) \quad (2.1)$$

2.3.1. Elementos básicos que componen una red neuronal:

Los elementos básicos que compone una neurona (Matich, 2001), se detallan a continuación, estas puede verse, en la Figura 2.2, un esquema de una red neuronal:

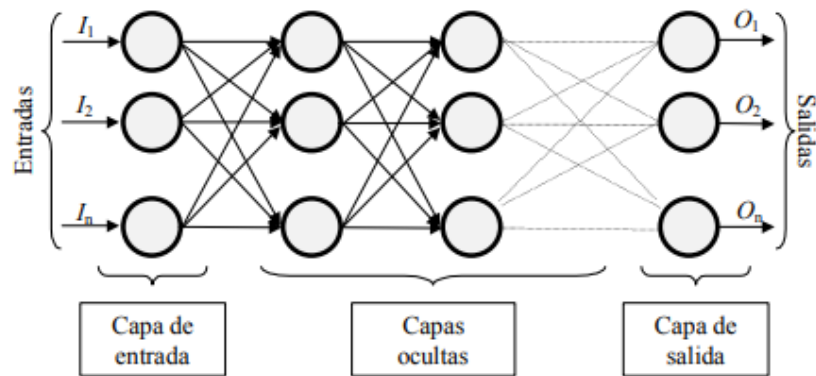


Figura 2.2: ejemplo de una red neuronal totalmente conectada (Matich, 2001)

Los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida”. Cabe mencionar que la capa oculta puede estar constituida por varias capas.

Una neurona es una unidad de procesamiento de información que es fundamental para la red neuronal, el cual forma las bases para diseñar redes neuronales artificiales. En la Figura 2.3, se representa el modelo de una neurona donde se identifica tres elementos básicos (Ocampo, 2014):

- Un conjunto de sinapsis o conexiones
- Un sumador para sumar las señales de entrada
- Una función de activación, esta función normaliza el rango de la salida, obteniéndose un intervalo cerrado; comúnmente (0,1) o (-1,1).

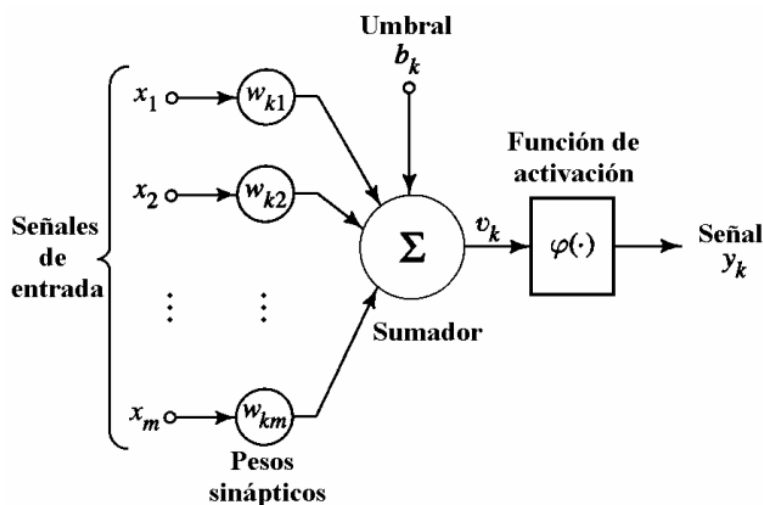


Figura 2.3: Modelo no lineal de una neurona. (Ocampo, 2014)

- Umbral: El umbral b_k tiene el efecto de subir o bajar la entrada neta de la función de activación, dependiendo si es positivo o negativo, respectivamente.

2.4 DEEP LEARNING:

Deep learning es un conjunto de técnicas y procedimientos algorítmicos basados en Machine Learning (aprendizaje de máquinas) para lograr que una máquina aprenda de la misma forma que lo hace un ser humano. Son algoritmos cuyo propósito es simular el comportamiento que lleva a cabo el cerebro para reconocer imágenes, palabras o sonidos, estos algoritmos funcionan en base a un proceso por capas, donde el aprendizaje profundo simula el funcionamiento básico del cerebro, que se realiza a través de las neuronas. El aprendizaje profundo simula el funcionamiento básico del cerebro, que se realiza a través de las neuronas. En *Deep Learning*, esas neuronas serían las capas. Se puede implementar con lenguajes de desarrollo como *R*, *C++*, *Python*, etc. (Navarro, 2015)

Entre las mejores aplicaciones está en la comprensión de enfermedades y terapias genéticas, análisis de imágenes médicas, como radiográficas y resonancia magnéticas, aumentando la precisión de diagnósticos, en un menor tiempo y con un menor coste que los métodos tradicionales, exploración de la posibilidad de reutilización de fármacos ya conocidos y probados para su uso contra nuevas enfermedades, identificación en textos de sentimientos positivos y negativos, temas, palabras claves, localización de caras e identificación de emociones faciales. (Ávila, 2018)

Los métodos que utilizan *Deep Learning* buscan descubrir un modelo (reglas de búsqueda, parámetros) utilizando un conjunto de datos (ejemplos) y un método para guiar el aprendizaje del modelo a partir de esos ejemplos. Al final del proceso de aprendizaje se tiene una función capaz de recibir por entrada los datos nuevos y proporcionar como salida una presentación adecuada para el problema. (Ponti & Paranhos da Costa, 2017)

La diferencia entre *Machine Learning* y *Deep learning* es como aprende la función $f(\cdot)$, a partir de un conjunto de parámetros genera el resultado deseado. En *Deep learning* tenemos métodos que aprenden $f(\cdot)$ por medio de las composiciones de funciones, ejemplo:

$$f(x) = f_L(f_1(x_1)) \dots \quad (2.2)$$

$f_l(\cdot)$: función de la capa l

x_l : vector de datos de la capa l

Generando como salida un vector: x_{l+1}

Para ser más preciso cada función hace uso de parámetros para realizar la transformación de los datos de entradas, vamos a denotar conjunto de parámetros (como una matriz) por w_l , relacionado a cada función f_l , el cual sería:

$$f_L(\dots f_2(f_1(x_1, W_1); W_2) \dots), W_L) \quad (2.3)$$

x_1 : datos de entrada

w_l : conjunto de parámetros

En la idea central de *Deep learning* es aprender representaciones de datos sucesivos e intermedios, es decir, los x_l , $l = 1 \dots l_n$. Los algoritmos de *Deep learning* resuelve el problema de encontrar los parámetros W directamente a partir de los datos y definen cada representación como combinaciones de otras (anteriores) y simples. Así la profundidad permite aprender una secuencia de funciones que transforman vectores mapeados de un espacio a otro, hasta alcanzar el resultado deseado.

Es por eso que la jerarquía de las representaciones es de suma importancia: cada función opera en una entrada que genera una representación que luego se pasa a la siguiente función. La hipótesis en DL es que, si tenemos suficientes capas L , espacios con una dimensionalidad suficientemente alta, es decir, el número de parámetros W en cada función y suficientes datos para conocer los parámetros W_l para todos l , entonces podremos ajustar el alcance de las relaciones en los datos originales, encontrando así la representación más adecuada para la tarea deseada.

A continuación, utilizaremos el ejemplo de reconocimiento de dígitos, pasando de un modelo superficial a un modelo profundo; tomemos como ejemplo el problema de ordenar imágenes que contienen dígitos numéricos de 10 dígitos (de 0 a 9). Para esto, utilizaremos la base de datos MNIST que tiene imágenes de 28×28 , ver Figura 2.4. Construyendo una arquitectura de red neuronal simple, podríamos tratar de resolver el problema de la siguiente forma:

- **Entrada:** Vectorizamos la imagen de entrada 28x28 para obtener un vector x de tamaño 784 x 1.
- **Neuronas:** Creamos 10 neuronas de salida, cada uno representa representando la probabilidad de entrada perteneciente a una de las 10 clases.
- **Pesos (parámetros):** Así como una red MLP, cada neurona de salida está asociada a pesos y términos bias que se aplican en los datos para generar una combinación línea como salida
- **Clasificación:** Se utiliza la función Softmax para que la salida sea interpretada como probabilidad permitiendo asignar un ejemplo a la clase con mayor probabilidad.

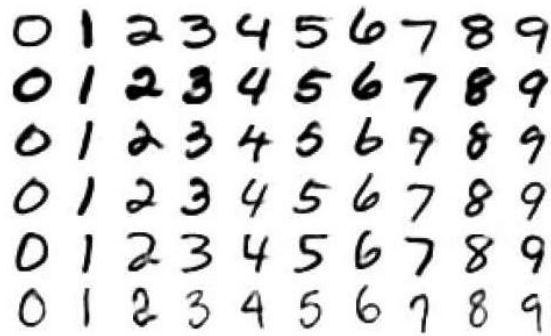


Figura 2.4. Imágenes de muestra de las clases base de 10 dígitos MNIST (Ponti & Paranhos da Costa, 2017)

Recordemos configurar una red neuronal y definir la notación. En nuestra arquitectura superficial, cada neurona j procesa una imagen de entrada en la forma $w_j^t x + b_j$. Para ilustrar, si tenemos una entrada de 4 valores en el vector, la neurona j produciría el siguiente resultado:

$$w_j^t x + b_j = (w_{j,1}x_1 + b_1) + (w_{j,2}x_2 + b_2) + (w_{j,3}x_3 + b_3) + (w_{j,4}x_4 + b_4) \quad (2.4)$$

Como se tiene sólo una capa necesitamos que ese valor resultante de la neurona j sea el puntaje de la red neuronal para la clase j . Así de los valores generados por el producto vectorial y sumados los bias, aplicamos la función softmax (función de activación), que es una forma de obtener valores normalizados para el intervalo 0-1 para cada clase c . La probabilidad de predecir y cómo pertenecer a la clase c , dada una imagen de entrada x ,

un conjunto de pesos w y los términos bias b , ambos relativos a la neurona de la clase c se define como:

$$P(y = c|x; w_c; b_c) = \text{softmax}_c(x^t w_c + b_c) = \frac{e^{x^t w_c + b_c}}{\sum_j |e^{x^t w_j + b_j}|} \quad (2.05)$$

En la figura 2.5. Se muestra dos casos: a) con entrada escalar y mientras que b) se tiene un vector como entrada:

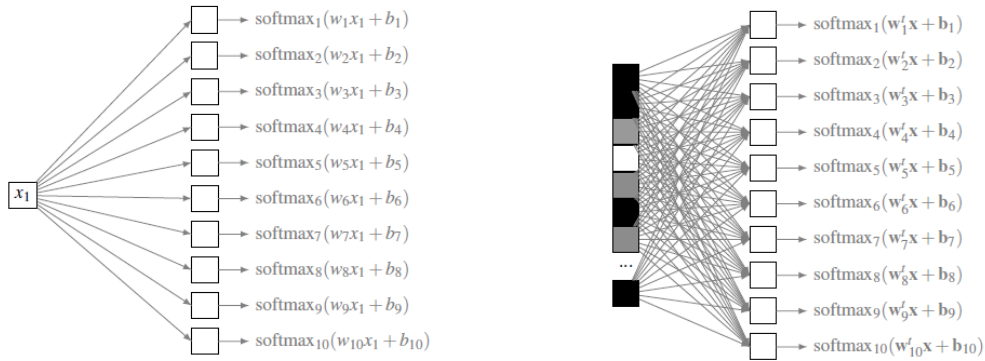


Figura 1.5: a) un único valor x_1 como entrada, b) vector x como entrada (Ponti & Paranhos da Costa, 2017)

Para mejorar los resultados se utiliza una arquitectura profunda, haciendo uso de la composición de funciones. Añadiremos 2 capas nuevas entre la entrada y la salida, estas capas se conocen como las capas ocultas. En *Deep learning* se conoce como *full connected (FC)*, que no es más que una capa oculta, así como utilizada en redes de *perceptron multicas (MLP)*. Ilustramos una arquitectura en la Figura 2.6. Por lo tanto, tendremos una predicción alcanzada por una composición en la forma:

$$\hat{y} = f(x) = f_3(f_2(f_1(x_1; W_1; b_1); W_2; b_2)), W_3; b_3), \quad (2.6)$$

En que $f_1(x_1) = x_2$, $f_2(x_2) = x_3$ y finalmente $f_3(x_3) = y$.

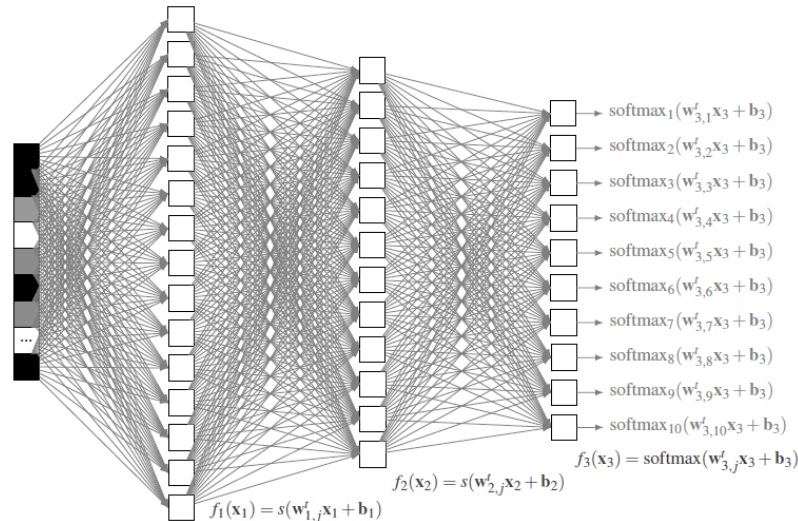


Figura 2.6: Una arquitectura profunda con dos capas ocultas, generando representaciones intermedias que preceden a la clasificación. (Ponti & Paranhos da Costa, 2017)

2.4.1 Taxonomía:

Deep learning puede clasificarse de varias formas, se puede destacar la taxonomía a partir de conceptos; clasifican los conceptos de acuerdo a la arquitectura de RNA (redes neuronales artificiales) que se implementan y por el tipo de entrenamiento de las arquitecturas figura 2.7. Como también, por el tipo de redes de acuerdo a su algoritmo más común que la conforman, tal como se muestra en los siguientes diseños figura 2.8 (Restrepo Arteaga, 2015)

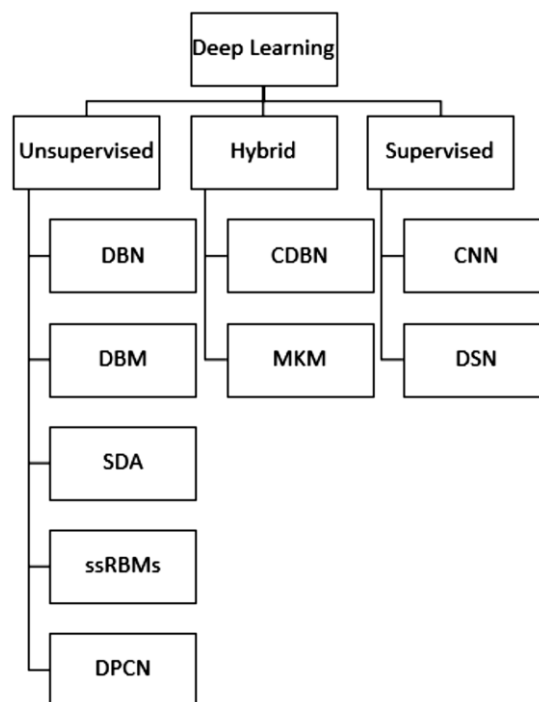


Figura 2.7. Taxonomía del deep learning a partir de conceptos de RNA (Restrepo Arteaga, 2015)

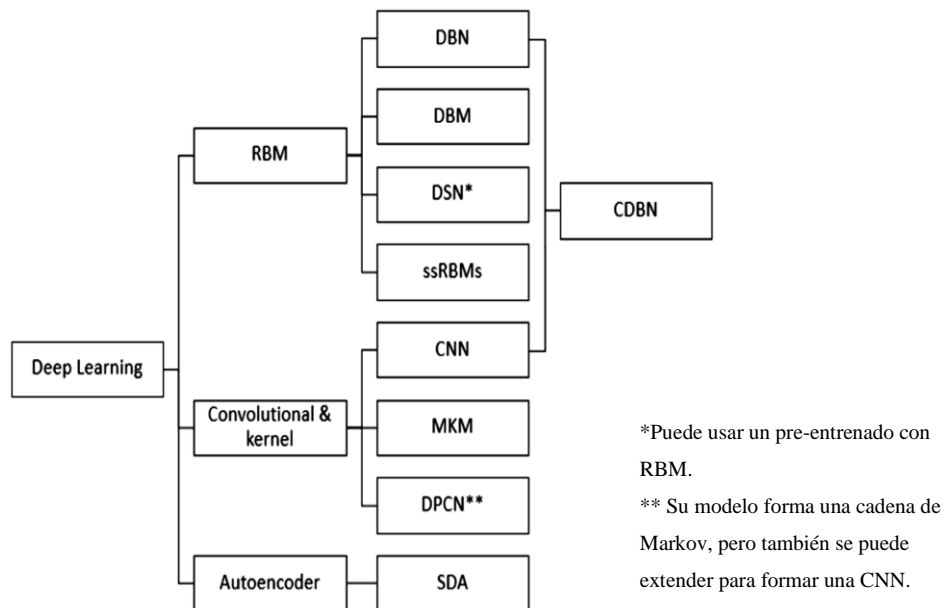


Figura 2.8: Taxonomía del deep learning a partir del tipo de algoritmo de aprendizaje (Restrepo Arteaga, 2015)

Esta taxonomía clasifica las redes de acuerdo a su algoritmo más común, ya que, de no ser así, la mayoría de las redes serían de algoritmos híbridos, pues pueden tener un pre-entrenamiento de un tipo diferente al entrenamiento formal.

2.5. REDES CONVOLUCIONALES

Las redes neuronales convolucionales trabajan dividiendo y modelando la información en partes más pequeñas, y combinando esta información en las capas más profundas de la red. Por ejemplo, en el caso del tratamiento de una imagen, las primeras capas tratarían de detectar los bordes de las figuras. Las siguientes capas buscarían combinar los patrones de detección de bordes para conseguir formas más simples, y aplicar patrones de posición de objetos, iluminación. Por último, en las últimas capas se intentará hacer coincidir la imagen con todos los patrones descubiertos, para conseguir una predicción final de la suma de todos ellos. Así es como las redes neuronales convolucionales consiguen modelar una gran cantidad de datos, dividiendo previamente el problema en partes para conseguir predicciones más sencillas y precisas (Antona Cortés, 2017).

Las redes neuronales convolucionales actúan como un "filtro en movimiento" que pasa a través de la imagen. Este filtro en movimiento, o convolución, se aplica a una cierta vecindad de nodos (que pueden ser los nodos de entrada, es decir, los píxeles), como se

muestra a continuación en la figura 2.9, donde el filtro aplicado es 0.5 x el valor del nodo: (Thomas, 2018)

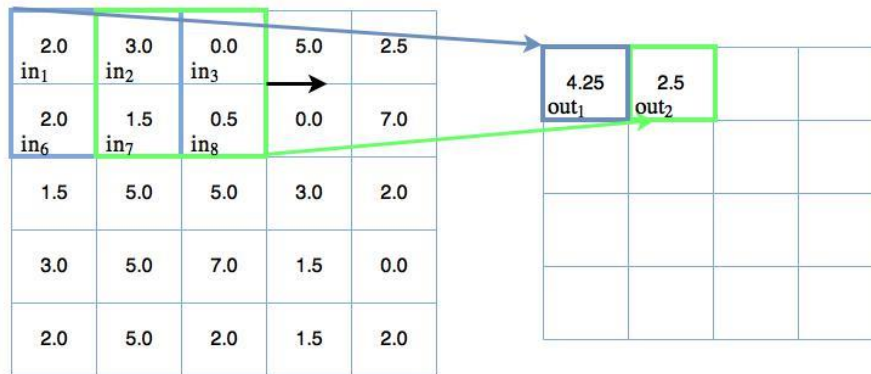


Figura 2.9: Filtro convolución 2x2 (Thomas, 2018)

Mover el filtro 2×2 (todos los pesos = 0.5) se realizaron los siguientes cálculos:

$$\begin{aligned} out_1 &= 0.5i_{n1} + 0.5i_{n2} + 0.5i_{n6} + 0.5i_{n7} & (2.7) \\ &= 0.5 \times 2.0 + 0.5 \times 3.0 + 0.5 \times 2.0 + 0.5 \times 1.5 \\ &= 4.25 \end{aligned}$$

$$\begin{aligned} out_2 &= 0.5i_{n2} + 0.5i_{n3} + 0.5i_{n7} + 0.5i_{n8} & (2.8) \\ &= 0.5 \times 3.0 + 0.5 \times 0.0 + 0.5 \times 1.5 + 0.5 \times 0.5 \\ &= 2.5 \end{aligned}$$

Este tipo de red es una variación de un perceptrón multicapa, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones.

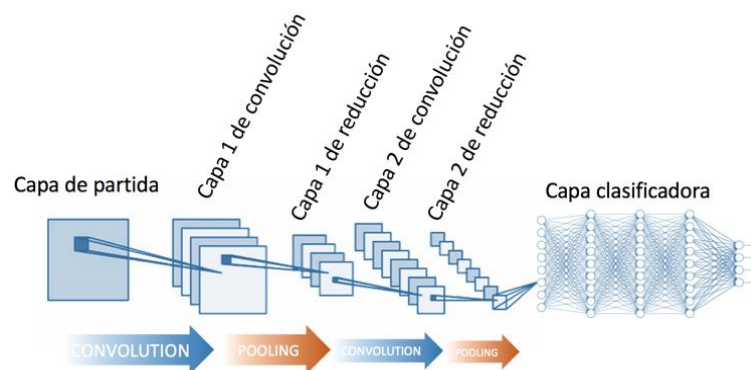


Figura 2.10: Estructura- fuente: <http://www.diegocalvo.es/red-neuronal-convolucional/>

En general, todas las redes neuronales convolucionales están formadas por una estructura compuesta por 3 tipos de capas, ver figura 2.10: (Loncomilla, 2016).

- Capa convolucional
- Capa de reducción o pooling
- Capa clasificadora totalmente conectada

2.5.1. Capa convolucional:

Esta capa le da nombre de la red. En vez de utilizar la multiplicación de matrices como en las redes neuronales estándar, se aplica una operación llamada convolución. Esta operación recibe como entrada una imagen, y sobre ella aplica un filtro que nos devuelve su mapa de características, reduciendo así el tamaño de sus parámetros ver figura 2.11.

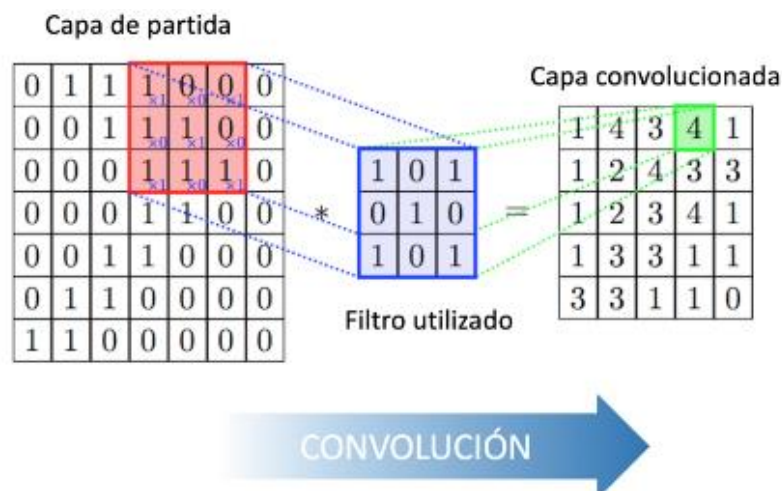


Figura 2.11: Fuente: <http://www.diegocalvo.es/red-neuronal-convolucional>

Una vez se ha recorrido toda la imagen se obtiene la matriz de activación completa que contiene las características que se buscan en la imagen para cada filtro, como se observa este proceso con mayor claridad en la figura 2.12.

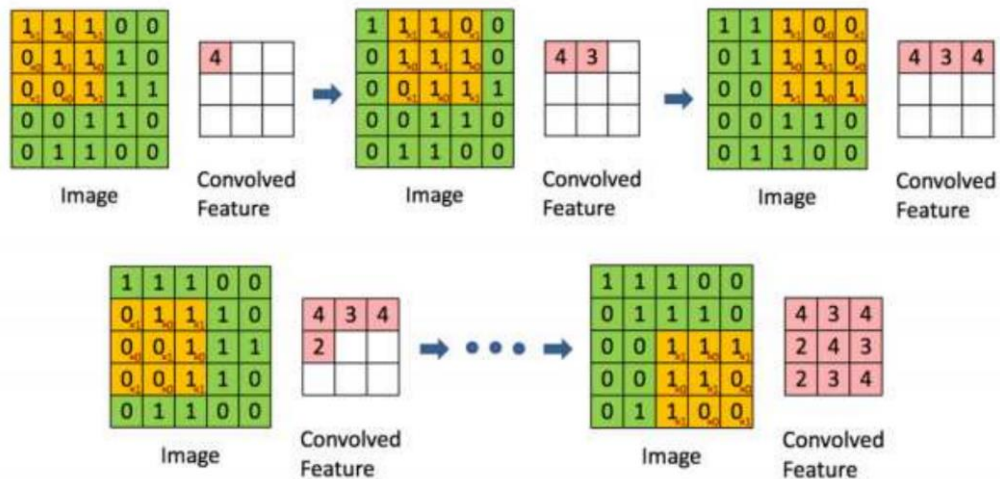


Figura 2.12: Obtención de la matriz de activación. (Durán Suárez, 2017)

Por ejemplo, si el kernel calcula la diferencia entre el brillo de un píxel y su vecino, la magnitud de la convolución va a ser mayor en las áreas de bordes o contrastes fuertes (diferencia mayor), y menor en las áreas sin contrastes (diferencia cero). La convolución convierte la imagen original en una versión comprimida, donde cada punto representa la presencia o ausencia de algún rasgo particular, por ejemplo bordes o siluetas. Esta nueva información alimenta a una capa de entrada, y el proceso sigue igual que en una RN (redes neuronales) convencional, este proceso se observa con mayor claridad en la figura 2.13. (Perez, 2018)

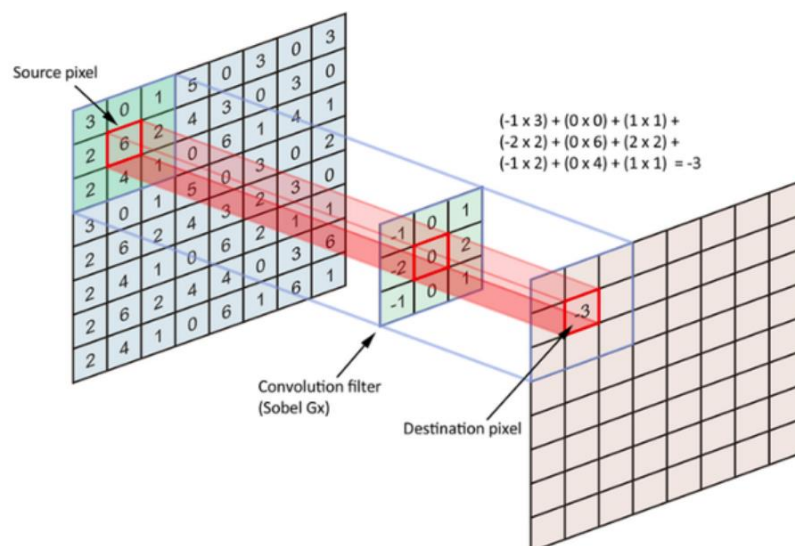


Figura 2.13: Convolución a una imagen original a una versión comprimida (Perez, 2018)

2.5.2. Capa de reducción o pooling:

Esta capa se coloca generalmente después de la capa convolucional. Es la encargada de reducir la cantidad de parámetros a analizar reduciendo las dimensiones espaciales

(ancho x alto), quedándose de esta forma con las características más comunes. La operación que lleva a cabo esta capa también se llama reducción de muestreo o submuestreo, ya que la reducción de tamaño implica también una pérdida de información. Sin embargo, para una red neuronal, este tipo de pérdida puede ser beneficioso debido a:

- La reducción del tamaño provoca una menor sobrecarga de cálculo en las próximas capas de la red.
- Reduce habitualmente el sobreajuste.

La operación que se suele aplicar en esta capa es “max-pooling”, que divide la imagen de entrada en un conjunto de rectángulos, y respecto a cada uno de ellos, se queda con el valor máximo. Además, como se comentó en el párrafo, es importante destacar que existen dos tipos de *pooling*: *mean-pooling* o *average-pooling* y *max-pooling*. En la Figura 2.14, pueden verse los efectos de aplicar un tipo de pooling u otro.

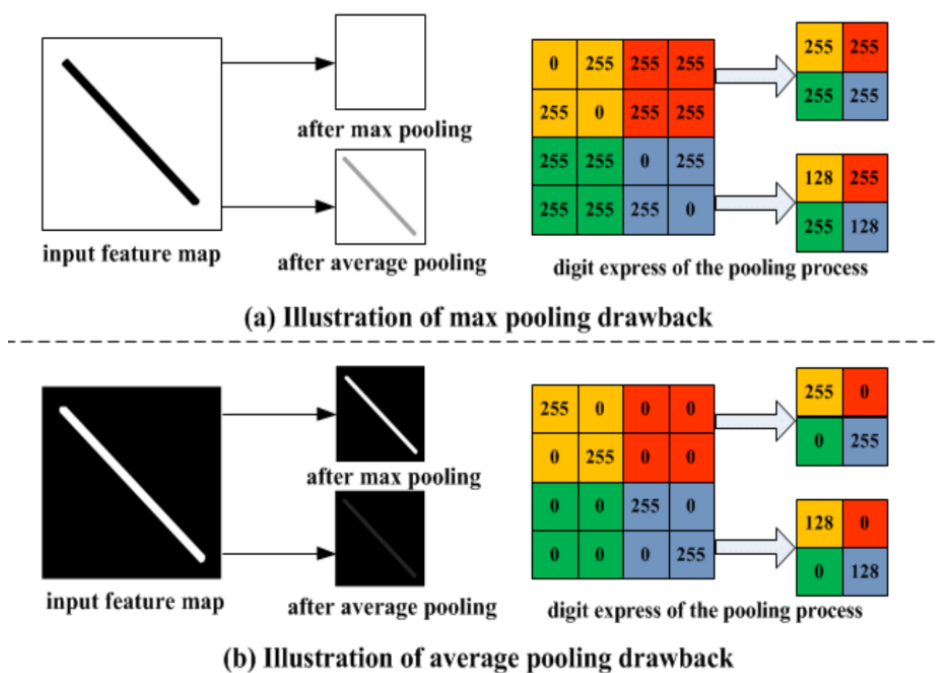


Figura 2.14: Resultado de aplicar diferentes tipos de pooling. (Durán Suárez, 2017)

2.5.3. Capa clasificadora totalmente conectada:

Una vez que la imagen ha pasado tanto por las capas convolucionales como las de pooling y se han extraído sus características más destacadas, los datos llegan a la fase de clasificación. Para ello, las redes convolucionales normalmente utilizan capas completamente conectadas en las que cada píxel se trata como una neurona independiente.

Las neuronas de esta fase funcionan como las de un perceptrón multicapa, donde la salida de cada neurona se calcula multiplicando la salida de la capa anterior por el peso de la conexión, y aplicando a este dato una función de activación. En la figura 2.15, se observa:

- Un encoder (codificador) y un decoder (decodificador)
- Comprimen la información
- Entregan una salida por píxel

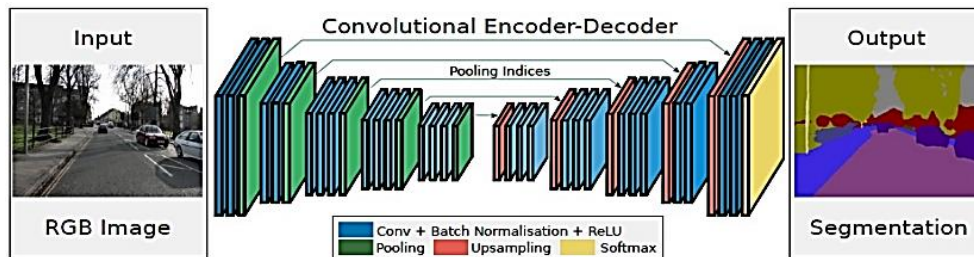


Figura 2.15: Fully convolutional networks (Loncomilla, 2016)

2.6. CONVOLUCIÓN:

En la convolución se realizan operaciones de productos y sumas entre la capa de partida y los n filtros (o kernel) que genera un mapa de características. Las características extraídas corresponden a cada posible ubicación del filtro en la imagen original.

La ventaja es que el mismo filtro (= neurona) sirve para extraer la misma característica en cualquier parte de la entrada, con esto que consigue reducir el número de conexiones y el número de parámetros a entrenar en comparación con una red multicapa de conexión total. (Calvo D. , 2017)

En el ejemplo cada píxel de salida es una combinación lineal de los píxeles de entrada, figura 2.16.

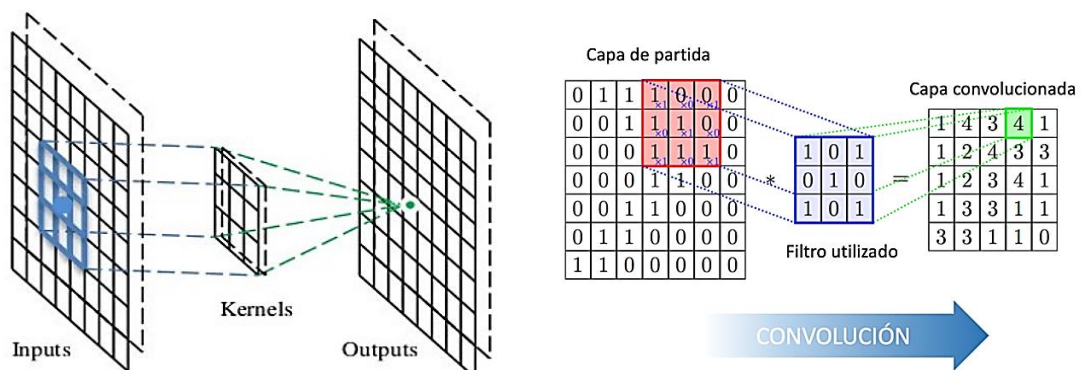


Figura 2.16: Sliding window (Ventana deslizante). (Calvo D. , 2017)

Después de aplicar la convolución se les aplica a los mapas de características una función de activación.

La función de activación recomendada es sigmoide y ReLU, seleccionando una tasa de aprendizaje adecuada y monitorizar la fracción de neuronas muertas.

La función de activación utiliza la misma suma ponderada de la entrada anterior, y la transforma una vez más como salida.

$$z = b + \sum_i w_w x_i \quad (2.9)$$

$Z =$ suma ponderada de entrada

2.6.1. Sigmoide:

Transforma los valores introducidos a una escala (0,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0. (Calvo D., 2018)

$$f(x) = \frac{1}{1-e^{-x}} \quad (2.10)$$

$e = 2,71828$ (constante exponencial)

Primero establecemos que la variable z equivale a nuestra suma ponderada de entrada y después la pasamos a través de la función sigmoide.

$$z = b + \sum_i w_i x_i \quad (2.11)$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (2.12)$$

Aunque la ecuación parece complicada y arbitraria, en realidad tiene una forma bastante simple. La podemos ver si trazamos el valor de $\sigma(z)$ como función de la entrada z , ver figura 2.26.

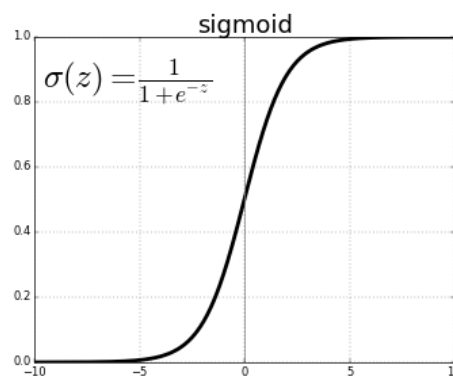


Figura 2.17: función de activación sigmoide. (Calvo D., 2018)

Podemos ver que $\sigma(z)$ actúa como una especie de función que comprime, comprimiendo la salida a un rango de 0 a 1. En el centro, donde $z = 0$,

$$\sigma(0) = \frac{1}{(1+e^0)} = \frac{1}{2} \quad (2.12)$$

Para valores negativos grandes de z , el término e^{-z} en el denominador crece exponencialmente, y $\sigma(z)$ se aproxima a 0. Al contrario, valores positivos grandes de z reducen e^{-z} hacia 0, y $\sigma(z)$ se aproxima a 1. (Alvarado, 2018)

2.7.2. ReLU (Rectified Lineal Unit)

La función ReLU transforma los valores introducidos anulando los valores negativos y dejando los positivos tal y como entran. Figura 2.18.

$$R(z) = \max(0, z) = \begin{cases} 0 & \text{for } z < 0 \\ z & \text{for } z \geq 0 \end{cases} \quad (2.13)$$

Se define como: $R(z) = \max(0, z)$

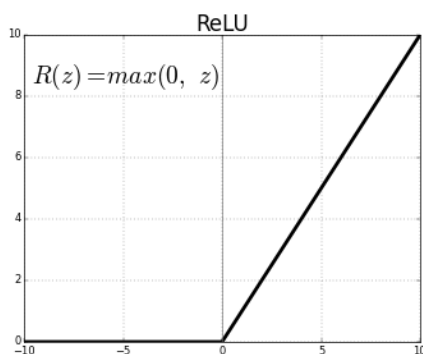


Figura 2.18: Función de activación ReLU. (Calvo D. , 2018)

En otras palabras, las ReLUs permiten el paso de todos los valores positivos sin cambiarlos, pero asigna todos los valores negativos a 0.

2.8. MAX-POOLING

Max Max Pooling ("Elegir el máximo") es un atajo para comprimir el espacio de datos sin afectar mucho su información. Esto ayuda a acelerar el procesamiento y entrenamiento de una CNN. (Perez G. , 2021)

La agrupación máxima (Max Pooling) es la aplicación de una ventana móvil a través de un espacio de entrada 2D, donde el valor máximo dentro de esa ventana es la salida. La agrupación máxima se realiza aplicando un filtro máximo a (generalmente) subregiones

no superpuestas de la representación inicial. Max pooling hace dos cosas principales, ver figura 2.19:

- Reduce la cantidad de parámetros dentro del modelo (como puede observar anteriormente, la salida es más pequeña que la entrada): esto se denomina muestreo descendente o submuestreo.
- Generaliza los resultados de un filtro convolucional, lo que hace que la detección de características sea invariable a los cambios de escala u orientación.

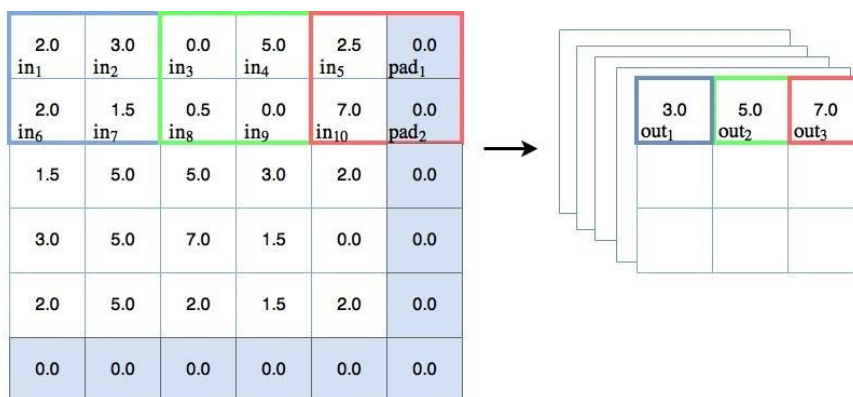


Figura 2.19: Ejemplo visual (con algo de relleno) de este tutorial de red convolucional TensorFlow (Thomas, 2018)

Como puede ver, cada ventana móvil de color captura el valor máximo dentro del cuadrado 2x2 y lo envía al lado derecho.

Otro ejemplo de Max Pooling, es la estrategia de reducción de muestras en redes neuronales convolucionales, ver figura 2.20.

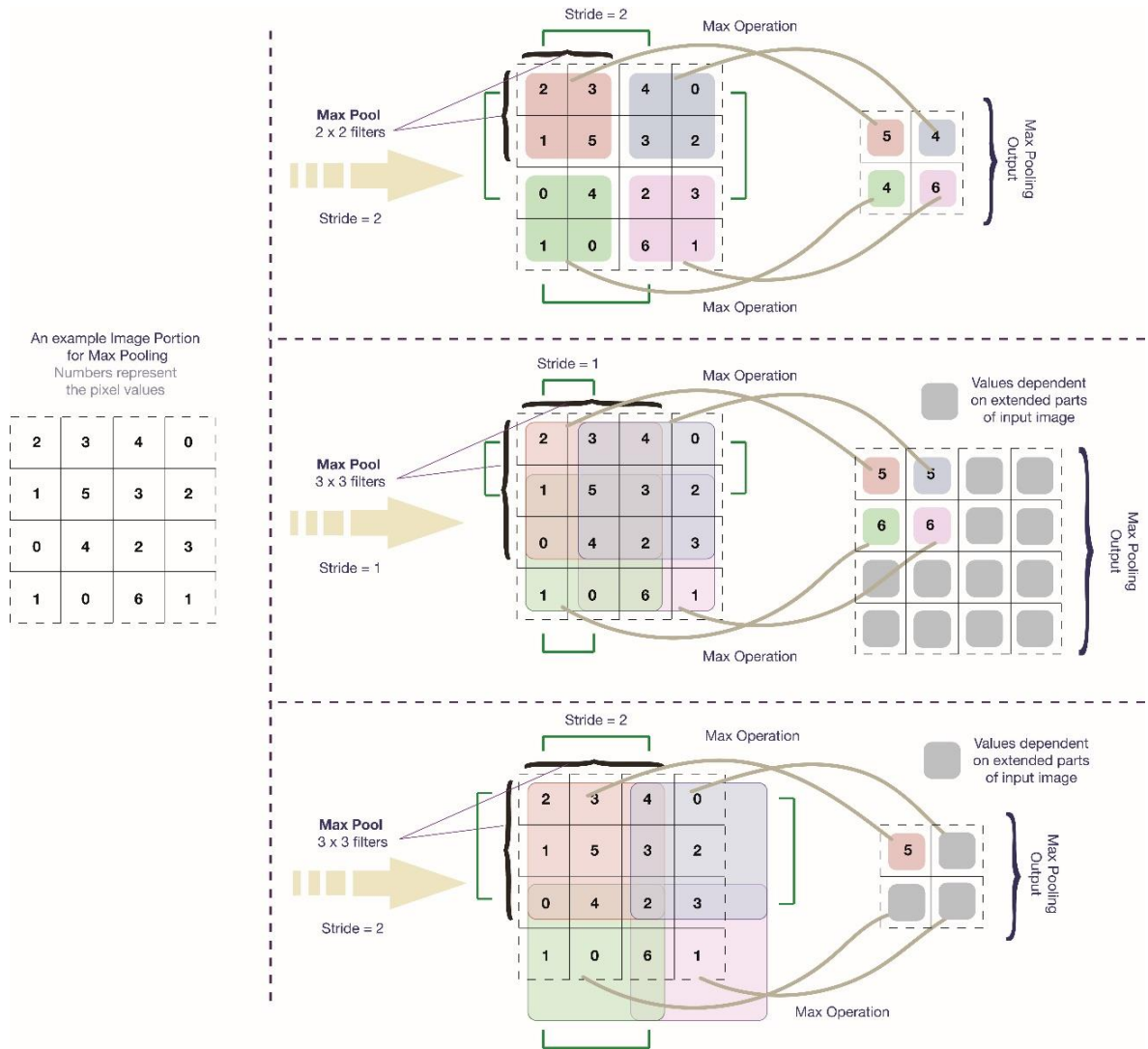


Figura 2.20: Fuente: tesis doctoral (Shankar, 2017)

Aquí en la figura, mostramos la operación sobre el espacio de píxeles. Alternativamente podemos hacer una operación similar en algún otro espacio matemático. Además, uno puede cambiar la operación de llevar 'Máx' a otra cosa, por ejemplo, tomar un 'Promedio' (esto es lo que se hace en la agrupación promedio).

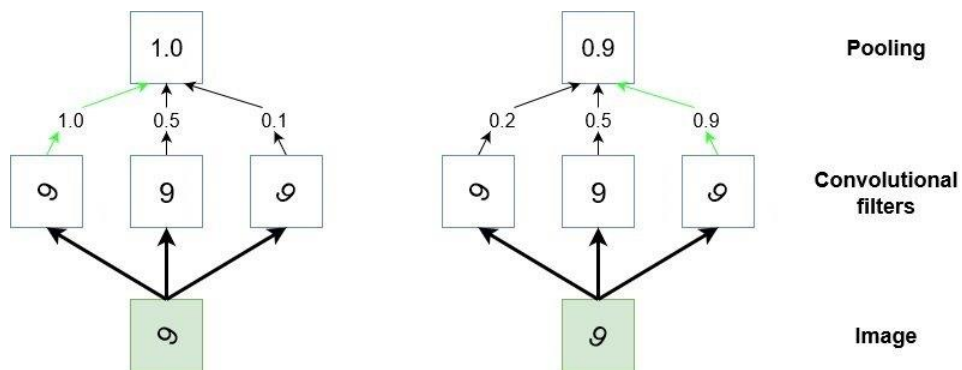


Figura 2.21: Convolucionales para detectar el dígito 9 en una imagen. . (Calvo D. , 2018)

En la figura 2.21, puede ver una representación estilizada de varios mapas de características convolucionales que se han sintonizado durante el entrenamiento para detectar el dígito "9" en una imagen, pero en diferentes orientaciones. La agrupación máxima permite una generalización que detecta un "9" sin importar la orientación o tamaño, dentro de la entrada. Es un tipo de movimiento de datos de bajo nivel a información de nivel superior. (Thomas, 2018)

Por ejemplo, en la figura 2.22, si tenemos una matriz 4×4 que representa nuestra entrada inicial. Digamos, también, que tenemos un filtro 2×2 que ejecutaremos sobre nuestra entrada. Tendremos un paso de 2 (lo que significa que (dx, dy) para pasar por encima de nuestra entrada será $(2, 2)$) y no se superpondrán regiones.

Para cada una de las regiones representadas por el filtro, tomaremos el máximo de esa región y crearemos una nueva matriz de salida donde cada elemento sea el máximo de una región en la entrada original.

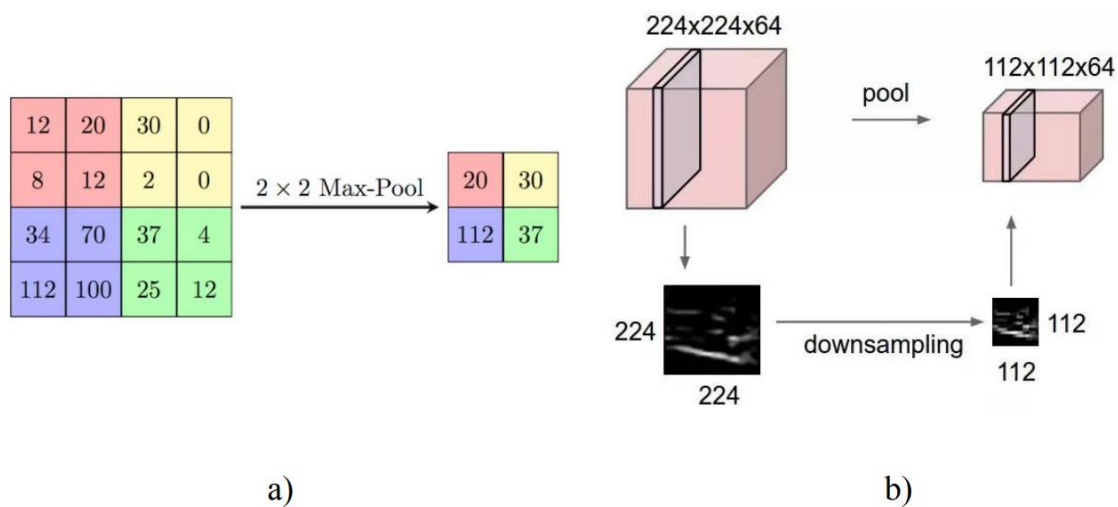


Figura 2.22: Operación de agrupación máxima en a) ejemplos pictóricos y b) basados en imágenes usando una agrupación tamaño de 2×2 y zancada de 2. (Salumaa, 2018)

2.9. DATA AUGMENTATION

El aumento de datos es una forma de crear nuevos "datos" con diferentes orientaciones. Los beneficios de esto son dos: (Virdee, 2018)

- Capacidad de generar más datos a partir de datos limitados
- Evita el ajuste excesivo.

Trabajar con datos limitados tiene sus propios desafíos, usar el aumento de datos puede tener resultados positivos solo si las técnicas de aumento mejoran el conjunto de datos actual.

Para obtener más datos, solo tenemos que realizar pequeñas modificaciones en nuestro conjunto de datos existente. Cambios menores como en la figura 2.23, volteos o traslaciones o rotaciones. Nuestra red neuronal pensaría que estas son imágenes distintas de todos modos. (Raj, 2018)

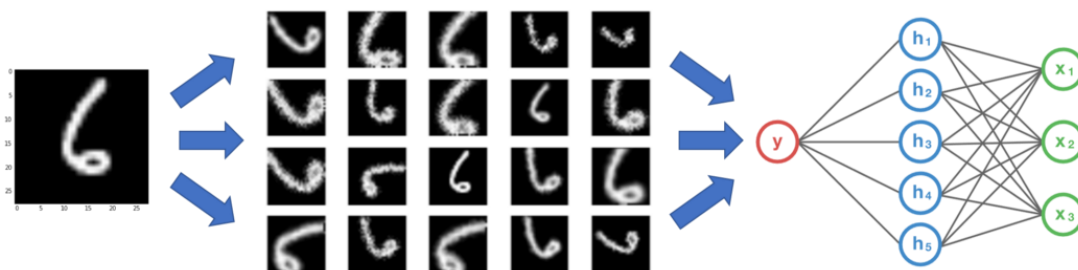


Figura 2.23: Aumento de datos en juego. (Raj, 2018)

Se dice que una red neuronal convolucional que puede clasificar los objetos de manera robusta, incluso si se coloca en diferentes orientaciones, tiene la propiedad denominada invarianza. Más específicamente, una CNN puede ser invariante a la traducción, el punto de vista, el tamaño o la iluminación (o una combinación de lo anterior). Esta es esencialmente la premisa de data augmentation.

Puede haber muchas variaciones que pueden afectar los resultados, como el tamaño del conjunto de datos, las técnicas de aumento, el tamaño del lote, el tamaño de la imagen y los parámetros de entrenamiento, por nombrar algunos. Figura 2.24.

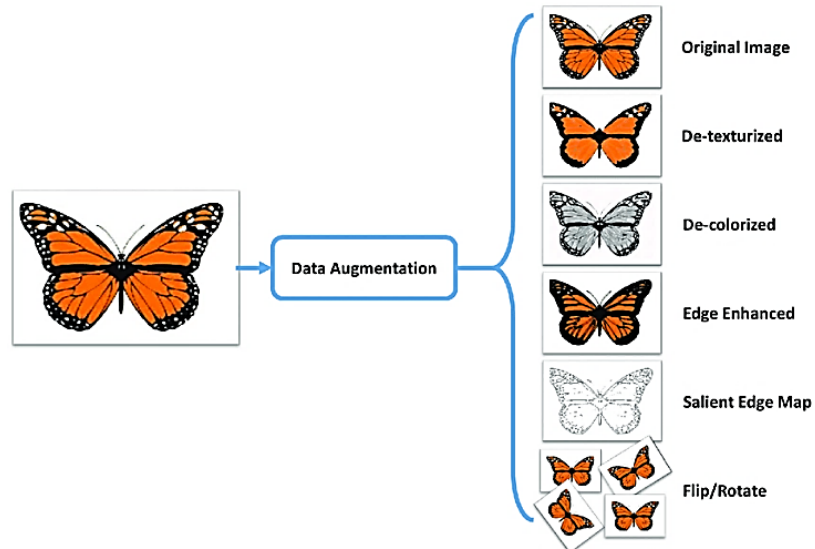


Figura 2.24: Fuente: Aumento de datos mediante la transformación de preservación semántica para SBIR. (Raj, 2018)

2.9.1. Aprendizaje profundo para el aumento de datos:

Pasando de lo simple a lo complejo, hay algunas cosas más interesantes que solo giros y rotaciones que puede hacer a su conjunto de datos para hacerlo más robusto, como son: (gage, 2018)

a) Transferencia de estilo neuronal:

Las redes neuronales han demostrado ser eficaces para transferir elementos estilísticos de una imagen a otra, como "Starry Stanford" aquí:

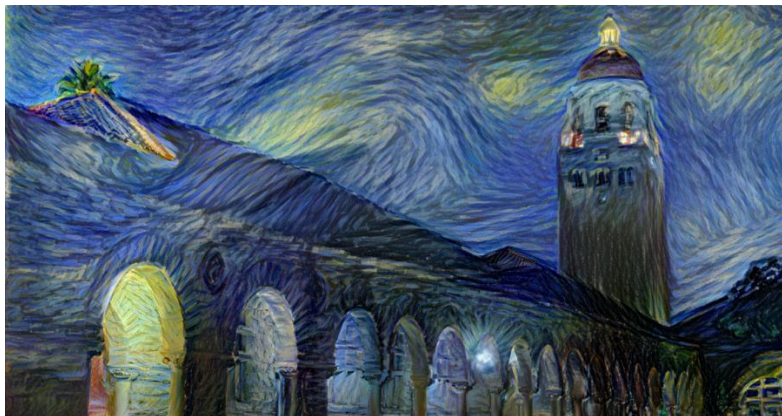


Figura 2.25: Fuente: Artistas e Inteligencia Artificial.nte. (gage, 2018)

Puede utilizar redes pretrained que transfieren estilos exteriores a sus imágenes de entrenamiento como parte de una tubería de aumento de conjunto de datos.

b) Redes de Publicidad Generativa:

Un nuevo tipo de algoritmo llamado GAN tiene la capacidad para generar contenido (de todo tipo) que es bastante bueno. Ejemplo de imágenes usando algoritmos GAN para la traducción de imagen a imagen y se obtiene estos resultados.

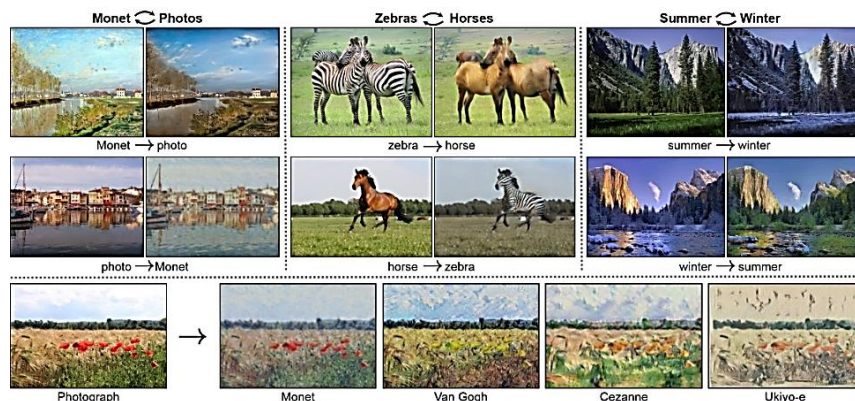


Figura 2.26: Fuente UC Berkeley. (gage, 2018)

c) AutoAugment de Google:

Usa Aprendizaje automático para aumentar su conjunto de datos. La idea es que los aumentos correctos dependen de su conjunto de datos, y se pueden aprender a través de un modelo: aunque los aumentos reales en sí son bastante simples.

2.10 RESNET

A diferencia de las arquitecturas de red secuenciales tradicionales como AlexNet, OverFeat y VGG, ResNet es, en cambio, una forma de "arquitectura particular" que se basa en módulos de microarquitectura (también llamadas "arquitecturas de red en red").

El término microarquitectura se refiere al conjunto de "bloques de construcción" utilizados para construir la red. Una colección de bloques de construcción de microarquitectura (junto con sus capas estándar CONV, POOL, etc.) conduce a la macroarquitectura (es decir, la red final en sí misma). (Rosebrock, 2017)

Las ResNets desarrolladas en la figura 2.27, son arquitecturas modularizadas que apilan bloques de construcción de la misma forma de conexión

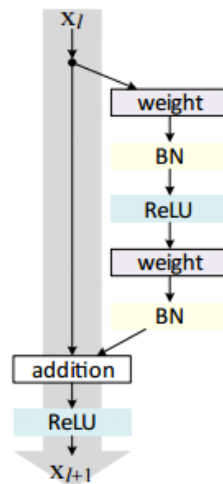


Figura 2.27: El módulo residual en ResNet (Rosebrock, 2017)

Se puede obtener una mayor precisión al actualizar el módulo residual para usar asignaciones de identidad. Las redes residuales profundas (ResNets) se puede expresar en una forma general: (He, 2016)

$$\begin{aligned}
 y_l &= h(x_l) + \mathcal{F}(x_l, w_l) \\
 x_{l+1} &= f(y_l)
 \end{aligned}
 \quad (2.14)$$

Donde x_l y x_{l+1} son entradas y salidas de la unidad l -ésima, y \mathcal{F} es una función residual. En, $h(x_l) = x_l$ es un mapeo de identidad y \mathcal{F} es una función ReLU. Figura 2.28

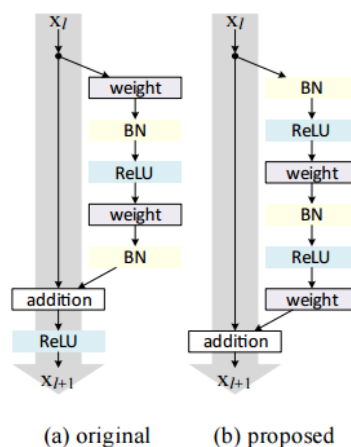


Figura 2.28:(a) El módulo residual original. (b) El módulo residual actualizado usando pre-activación. (Rosebrock, 2017)

Aunque ResNet es mucho más profundo que VGG16 y VGG19, el tamaño del modelo es en realidad sustancialmente menor debido al uso de la agrupación promedio global en

lugar de capas totalmente conectadas, lo que reduce el tamaño del modelo hasta 102 MB para ResNet50. (Yeung, 2017)

2.10.1. Características ResNet:

- Las capas extras reducen efectivamente el error.
- Se trabaja con residuales.

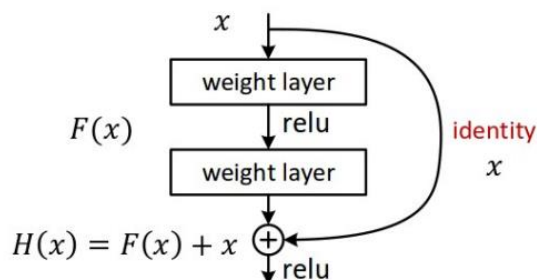


Figura 2.29: Red residual (Yeung, 2017)

Ejemplo:

Los modelos más profundos son más difíciles de optimizar. El modelo más profundo debería ser capaz de funcionar al menos tan bien como el modelo menos profundo.

Una solución por construcción es copiar las capas aprendidas del modelo menos profundo y establecer capas adicionales para el mapeo de identidad.

Para ello usaremos capas para adaptarse a los residuos $F(x) = H(x) - x$, en lugar de $H(x)$ directamente.

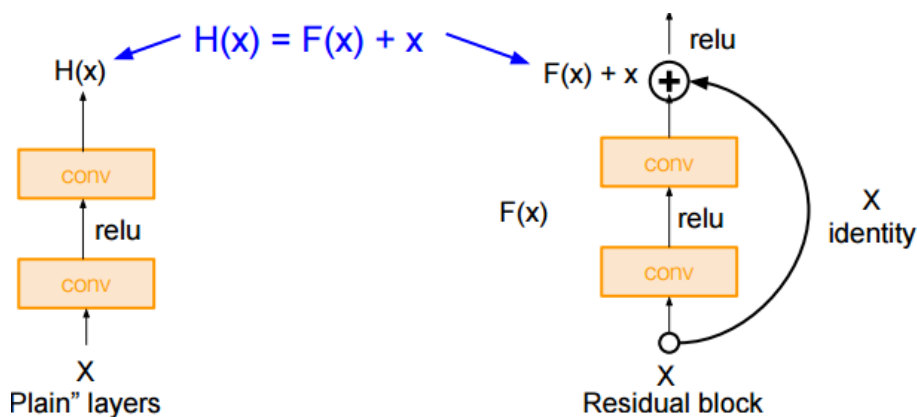


Figura 2.30: Capas de red para ajustar una asignación residual (Yeung, 2017)

En el siguiente modelo tenemos la arquitectura completa de ResNet, con grupo de bloques residuales y cada bloque residual tiene dos capas de conv 3x3

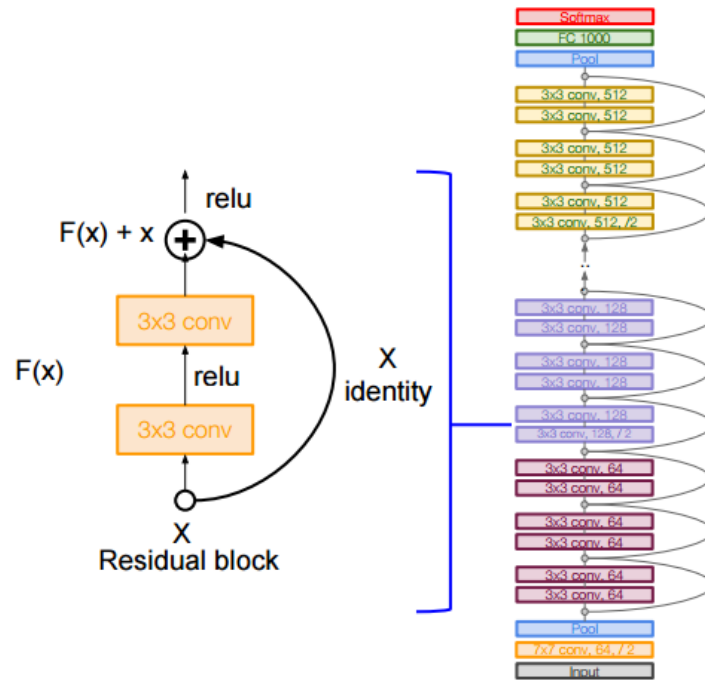


Figura 2.31: Arquitectura completa de ResNet (Yeung, 2017)

2.11. REDES CONVOLUCIONALES DENSAMENTE CONECTADAS (DENSELY)

Según (Huang, 2017), las redes densas se describen por conectividad densa, función compuesta, capas de agrupación, tasa de crecimiento, capas de cuello de botella y compresión, que se describen a continuación:

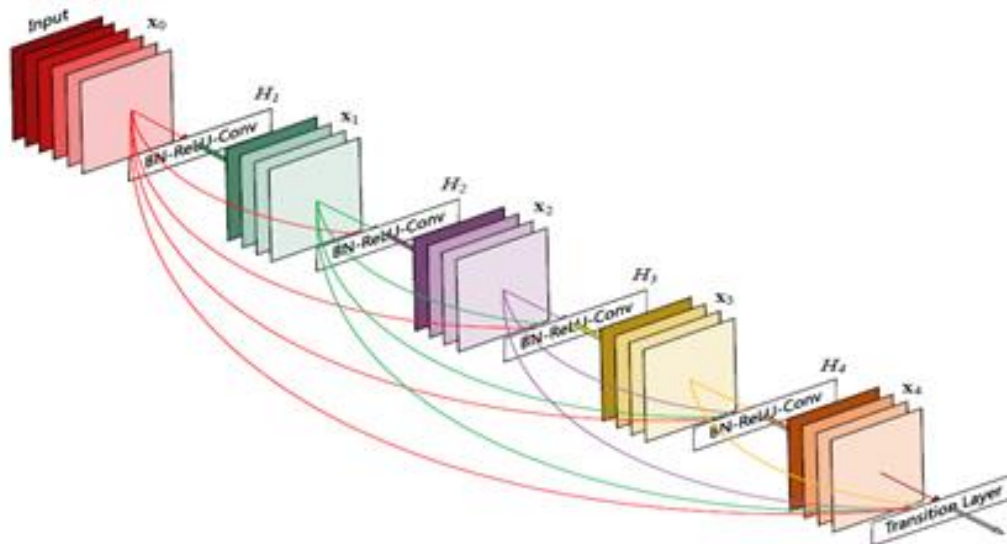


Figura 2.32: Una red densamente convolucional con una tasa de crecimiento de $k = 4$. Cada capa toma todos los mapas de características anteriores como entrada (Huang, 2017).

2.11.1. Conectividad densa. La Fig. 1. muestra el esquema de DenseNet. donde, la ℓ^{th} capa recibe los mapas de características de todas las capas precedentes. Este es el motivo por el cual se denomina Red convolucional densa (DenseNet).

2.11.2. Función compuesta. Consta de tres operaciones: normalización por lotes (BN) (Szegedy S. I., 2015), seguida de una unidad lineal rectificada (ReLU) (X. Glorot, 2011) y una convolución de 3×3 (Conv).

2.11.3. Agrupación de capas. Un paso importante de CNN es el muestreo descendente que cambia el tamaño de los mapas de características. Para simplificar la arquitectura, la red se divide en varios bloques densos y densamente conectados. Las capas de transición consisten en una normalización por lotes y una operación convolucional de 1×1 y un agrupamiento promedio de 2×2 .



Figura. 2.33: Bloques densos donde se muestran las capas de transición y cambian los tamaños de los mapas de características mediante convolución y agrupación (Huang, 2017).

2.11.4. Tasa de crecimiento. La función produce mapas de características, donde la capa tiene mapas de características, donde están los canales de las imágenes.

2.11.5. Capas de cuello de botella. La convolución se introduce como capa de cuello de botella antes de cada convolución para reducir los mapas de características.

2.11.6. Compresión. Finalmente, un bloque denso que, de los mapas de características, se reduce por factor, lo que se denomina factor de compresión.

2.12 MÉTRICAS DE EXPERIMENTACIÓN

2.12.1 Matriz de confusión

Es una herramienta que permite la visualización, el rendimiento del modelo clasificador una vez entrenado (Vega Arias, 2019), ver tabla 2.1.

Tabla 2.1 Esquema de la matriz de confusión

| | | Predicted class | |
|--------------|-------------|-----------------|----------------|
| | | Class = Yes | Class = No |
| Actual Class | Class = Yes | True Positive | False Negative |
| | Class = No | False Positive | True Negative |

2.12.2 Accuracy (precisión)

La precisión mide la precisión del modelo al clasificar una muestra como positiva.

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}} \quad (2.15)$$

2.12.3 Loss (pérdida)

Valor de pérdida implica qué tan mal o bien se comporta un modelo después de cada iteración de optimización.

2.12.4 Precisión

Mide la precisión del modelo al clasificar una muestra como positiva. Cuando el modelo hace muchas clasificaciones positivas incorrectas, o pocas clasificaciones positivas correctas, esto aumenta el denominador y hace que la precisión sea pequeña. Por otro lado, la precisión es alta cuando:

- El modelo hace muchas clasificaciones positivas correctas (maximizar el positivo verdadero).
- El modelo realiza menos clasificaciones positivas incorrectas (minimiza los falsos positivos).

$$Precision = \frac{True_{positive}}{True_{positive} + False_{positive}} \quad (2.16)$$

2.12.5 Recuperación (Recall)

El Recall mide la capacidad del modelo para detectar muestras positivas. Cuanto mayor sea el recuerdo, más muestras positivas se detectarán.

La recuperación solo se preocupa por cómo se clasifican las muestras positivas. Esto es independiente de cómo se clasifiquen las muestras negativas.

$$Recall = \frac{True_{positive}}{True_{positive} + False_{negative}} \quad (2.17)$$

2.13. SISTEMA RECOMENDADOR (SR)

Los sistemas recomendadores cumplen el objetivo de explorar y filtrar las mejores opciones a partir de un perfil de usuario, considerando una importante cantidad de posibilidades diferentes. Esto involucra la construcción de un perfil de usuario que contenga preferencias, características, etc., el cual puede ser obtenido de forma implícita o explícita (Deco C. B., 2010).

Los elementos partícipes en la recomendación son (Sánchez Bocanegra, 2017):

- Perfil de usuario: Propiedades asociadas al usuario (por ejemplo, pacientes que padecen de Diabetes Tipo 1 con niveles en sangre de 11.2 mmol/L. El sistema recomendador recopila toda esa información y lo incluye en su banco de datos. Esta información facilita la personalización de la recomendación).
- Ítems: Elementos que un usuario busca de acuerdo a sus necesidades (por ejemplo: control de medicación, dosificación, prospecto, videos que describen el padecimiento o la enfermedad, etc.), los ítems representan la esencia de la recomendación ya que son los elementos que se ofrecen a los usuarios.
- Contexto: Define aquellas propiedades externas que añaden información al sistema (por ejemplo, la versión del sistema operativo, el navegador que utiliza, la hora de la consulta, tiempo que tarda en la elección de un ítem, etc.)
- Dominio: El perfil de los usuarios, los ítems y el contexto son los principales elementos de un sistema recomendador (por ejemplo, pertenecen al dominio de Diabetes toda la información relativa a pacientes, investigadores, profesionales sanitarios y a todo el conjunto de información de diabetes). Los métodos recomendadores buscan la mayor aproximación con un dominio particular que pueda ser útil entre las entidades partícipes.

Ejemplo: Lógica del funcionamiento de algunos de los sistemas de recomendación que han sido integrados a ciertos portales. (Ocampo-Botello, 2014):

- Recomendadores colaborativos: El primer recomendador colaborativo al cual haremos referencia será www.MovieLens.com, que es un recomendador de películas. Un usuario de esta tienda valora una película en una escala de 1 a 5 estrellas, siendo 1 malísima y 5 indica que esa película se tiene que ver; entonces el sistema utiliza valoraciones de otros usuarios de acuerdo a una

categoría y el sistema recomendará algunas en las que el usuario podría estar interesado. Este sistema recomendador emplea algoritmos basados en productos que dan mejores resultados:

- Recomendadores basados en contenido: Son aquellos que realizan la recomendación con base en la descripción del producto.
- Recomendadores basados en conocimiento: Sistemas basados en conocimiento, necesitan de un historial sobre un usuario, con tan solo unas preguntas se puede hacer la recomendación.
- Recomendadores híbridos: Como primer lugar, mencionaremos a la poderosa empresa de comercio electrónico y que además desarrolla algoritmos de recomendación complejos Amazon, que empezó siendo una librería en línea, es un ejemplo de sistema de recomendación híbrido, que mezcla los enfoques basados en contenido y colaborativos. Por ejemplo, Amazon utiliza algoritmos de recomendación personalizada para cada cliente. La tienda cambia radicalmente dependiendo de quien la visita, hace búsquedas para alguien que es un amante de las mascotas, libros, videojuegos y la tecnología, entre otras cosas. Esto es posible gracias al perfil del usuario, ya que se almacena con el historial de navegación, los productos adquiridos y las preferencias marcadas en el perfil.

2.14. OBJETOS DE APRENDIZAJE OA

Los objetos de aprendizaje son elementos de un nuevo tipo de instrucción basada en computadora. Basado en el paradigma orientado a objetos de la informática. Orientación a objetos altamente valora la creación de componentes (llamados "objetos") que pueden reutilizarse en múltiples contextos. Esta es la idea fundamental detrás de los objetos de aprendizaje: los diseñadores instructivos pueden construir pequeños (en relación con el tamaño de un curso completo) componentes de instrucción que se pueden reutilizar varias veces en diferentes aprendizajes contextos, además, los objetos de aprendizaje se entienden generalmente como entidades digitales. entregables a través de Internet, lo que significa que cualquier número de personas puede acceder a ellos y usarlos simultáneamente. (Wiley, 2000)

Los OA como son los recursos digitales de software que apoyan el proceso de enseñanza y aprendizaje, descritos mediante metadatos que incorporan consideraciones

tecnológicas y pedagógicas orientadas a la efectividad del proceso de enseñanza y aprendizaje, ver figura 2.34. (Sprock, 2015)

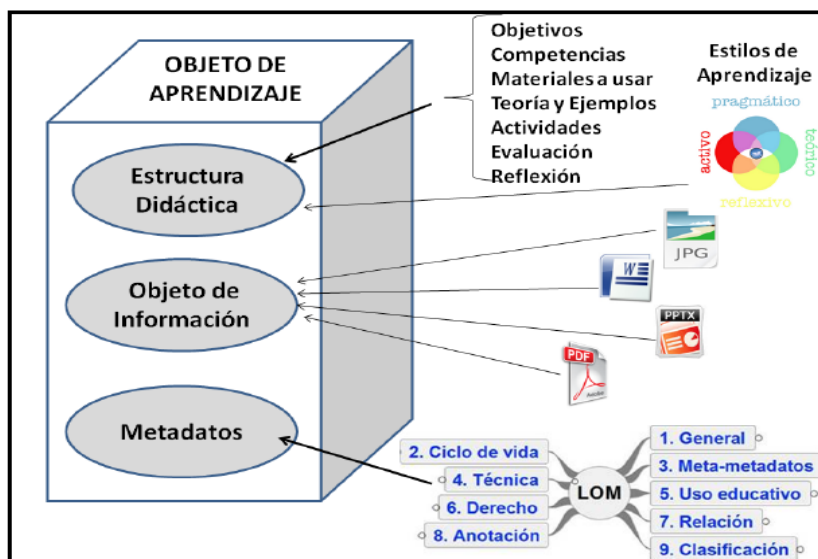


Figura 2.34: Estructura propuesta del Objeto de Aprendizaje (Cué, 2017)

A partir del análisis de las cualidades con las que debe cumplir un OA, se considera como: una unidad didáctica digital independiente, cuya estructura está formada por un objetivo de aprendizaje específico, un contenido, actividades y una autoevaluación, y que puede ser reutilizada en diferentes contextos tecnológicos (Repositorios, Entornos virtuales de enseñanza y aprendizaje) y educativos. Además cuenta con unos metadatos que propicien su localización dentro de los repositorios y permitan abordar su contextualización. (Maldonado M, 2015)

2.14.1. Características de los Objetos de aprendizaje

Los OA deben de cumplir con ciertas características que permitan su reutilización en diversas situaciones sin problemas de compatibilidad con otras plataformas. Para que esto sea posible, los OAs deben cumplir con ciertos requisitos técnicos y funcionales (Morgado, 2010):

- No se puedan dividir en unidades más pequeñas, es decir, debe contener una unidad mínima de aprendizaje.
- Con sentido en sí mismas (no dependan de otros).
- Susceptibles de ser combinados con otros OAs para componer una unidad superior (capítulo, bloque, unidad didáctica, etc.).

- Accesibles dinámicamente a través de una base de datos o repositorio. Esto es posible debido a la información de los OAs contenida en los metadatos.
- Interoperables, duraderas y capaces de soportar cambios tecnológicos sin ser rediseñados.

Para que los requisitos mencionados sean factibles, los OAs deben cumplir principalmente con las características de: Interoperabilidad, Accesibilidad, Reusabilidad y Granularidad.

a. Interoperabilidad: La posibilidad de que los OAs sean reutilizados refleja que deben tener la capacidad de ser interoperables, es decir, importados y exportados en cualquier tipo de plataformas, lo cual los hace también más durables. Para que esto sea posible es necesario estructurar la información de manera uniforme con la ayuda de estándares y especificaciones e-learning.

b. Accesibilidad: La Accesibilidad es entendida en este contexto como la capacidad de ser buscado y localizado a través de los metadatos (datos sobre los datos) de los objetos, de esta manera es posible conocer las características de los objetos desde diversos puntos de vista para su reutilización.

c. Reusabilidad: Se trata de una de las características más importantes de los OAs. El concepto de reusabilidad de OAs es descrito como la posibilidad y adecuación para que el objeto sea usable en futuros escenarios.

d. Granularidad: El tamaño o granularidad de los OAs es otra importante característica porque está estrechamente relacionado a la capacidad de reutilización. Un tamaño no adecuado podría dificultar o impedir la reusabilidad del OA, perdiendo con ello su principal característica. De acuerdo a la diversidad de definiciones de OAs, se pueden encontrar ejemplos de OAs con un nivel de granularidad muy pequeño como una imagen, o de un tamaño muy grande como un software.

2.14.2. Modelos de Metadatos

Existen distintos modelos de metadatos, cada uno de ellos con distintos esquemas de descripción, donde cada objeto se describe por medio de una serie de par de atributo-valor, es el que puede servir para recuperar la información.

2.14.2.1 Estándares de Metadatos

El uso de estándares en el diseño y producción de OA ofrece ventajas al momento de almacenar, recuperar y distribuirlos. Según la International Organization for Standardization (ISO), la estandarización es la actividad que tiene por objeto establecer, ante problemas reales o potenciales, disposiciones destinadas a usos comunes y repetidos, con el fin de obtener un nivel de ordenamiento óptimo en un contexto dado, que puede ser tecnológico, político o económico.

La generación de estándares de metadatos es una inversión en cuanto a la futura interoperabilidad ya que expande las posibilidades de las distintas partes para trabajar efectivamente en el largo plazo, sin importar el cambio de tecnología.

Ejemplo: Adaptación del esquema representativo de las capas y las iniciativas más relevantes para llegar a la interoperabilidad de contenidos en e-learning.

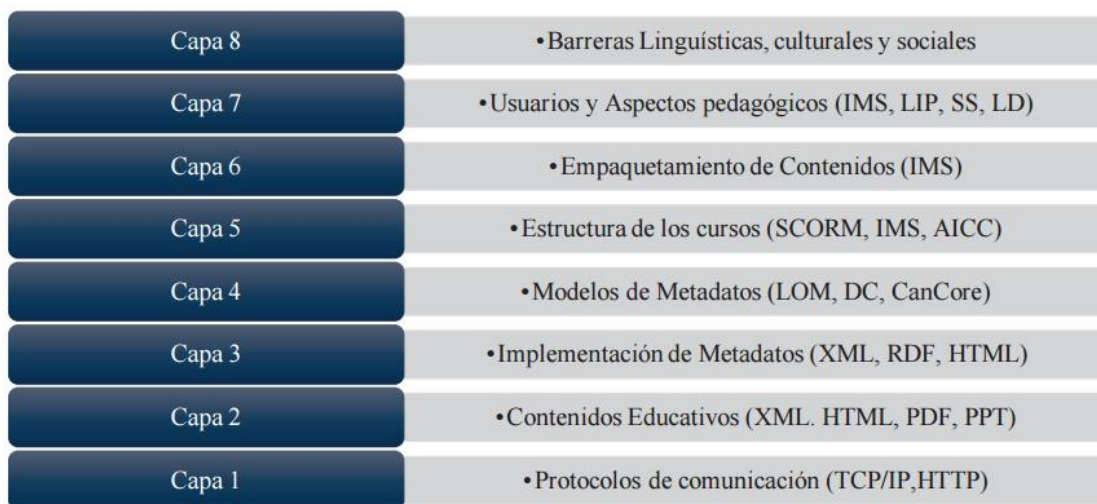


Figura 2.35: Esquema de las capas y las iniciativas más importantes para llegar a una interoperabilidad de contenidos en e-learning. (Morgado, 2010)

2.14.3 Modelos de Empaquetamiento

Dentro de las características relevantes con las que debe contar un OA están la interoperabilidad y la capacidad de ser publicable. Estas características cobran relevancia

cuando se desea almacenar, buscar, recuperar y compartir los OA a través de distintos sistemas informáticos. El empaquetamiento del OA facilita la interoperabilidad de este tipo de material educativo sobre distintos EVEAS.

2.14.3.1. Modelos de empaquetamiento de OA

El empaquetamiento de contenidos de un OA indica el formato en el que deben agruparse las colecciones de archivos con material educativo y detalla la sintaxis de un archivo en el que se describen y estructuran los contenidos de un determinado paquete de contenido. A este fichero se le denomina manifiesto del paquete y suele incluir metadatos.

Con este fin, el IMS (IMS Global Learning Consortium) definió un tipo de fichero XML para la descripción de los contenidos de los OA, a partir de las recomendaciones del LTSC (Learning Technology Standards Committee) y la AICC (Aviation Industry CBT Committee). De tal modo que cualquier EVEA pueda cargar el OA, leyendo su fichero de configuración IMSMANIFEST.XML. Para visualizar un OA, empaquetado bajo un modelo es necesario contar con un editor o una plataforma de aprendizaje que lo permita, siendo los EVEAs los más utilizados.

- **Learning resource meta-data specification:**

Actualmente IMS es el principal promotor y desarrollador de especificaciones abiertas orientadas al elearning, y que cubren más aspectos de la enseñanza virtual. Su objetivo es el de lograr interoperabilidad de aplicaciones y servicios en e-learning a partir de sus especificaciones.

Esta especificación conjuntamente con la guía desarrollada por ADL en su modelo de referencia SCORM, y por IEEE LTSC con su propuesta de metadatos para OA, son los que están teniendo una mayor repercusión en e-learning.

- **ADL/Sharable Content Object Reference Model (SCORM)**

Actualmente es el modelo más utilizado en la industria y que cuenta con mayor cantidad de herramientas que lo soportan. Es un perfil de aplicación, ya que combina muchas especificaciones (IMS, IEEE) y las particulariza para un caso concreto

La utilización de SCORM permite el empaquetamiento del contenido, actividades y metadatos, con esto se propicia la reutilización y se facilita la migración de OA entre diferentes EVEAS compatibles con el estándar.

2.15. ESTILOS DE APRENDIZAJES

El término “estilo de aprendizaje” se refiere al hecho de que cada persona utiliza su propio método o estrategias para aprender. Aunque las estrategias varían según lo que se quiera aprender, cada uno tiende a desarrollar ciertas preferencias o tendencias globales, tendencias que definen un estilo de aprendizaje. (de Mola Garay, 2011). Son los rasgos cognitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables de cómo los alumnos perciben interacciones y responden a sus ambientes de aprendizaje. (González Clavero, 2011). Es decir, tienen que ver con la forma en que los estudiantes estructuran los contenidos, forman y utilizan conceptos, interpretan la información, resuelven los problemas, seleccionan medios de representación (visual, auditivo, kinestésico), etc. Los rasgos afectivos se vinculan con las motivaciones y expectativas que influyen en el aprendizaje, mientras que los rasgos fisiológicos están relacionados con el género y ritmos biológicos, como puede ser el de sueño-vigilia, del estudiante. (Cazau, 2005)

2.16.1 Modelos de estilos de aprendizaje

Se han desarrollado distintos modelos y teorías sobre estilos de aprendizaje los cuales ofrecen un marco conceptual que permite entender los comportamientos diarios en el aula, cómo se relacionan con la forma en que están aprendiendo los alumnos y el tipo de acción que puede resultar más eficaz en un momento dado. Las primeras investigaciones acerca de los estilos de aprendizaje vieron la luz en la década de 1950, cuando desde la Psicología Witkin 1954, citado por Cabrera y Fariñas (2005) devino uno de los pioneros en el estudio de los “estilos cognitivos”, teorías que más tarde la Pedagogía adoptó y moldeó para sí. Desde entonces y hasta la actualidad las definiciones de estilos de aprendizaje han resultado disímiles. (González Clavero, 2011).

En principio, resulta posible integrar los modelos de estilos de aprendizaje. Un avance en este sentido es la propuesta de Martha M. Perea Robayo (2003), para la cual en general las teorías sobre los estilos de aprendizaje confluyen en cuatro categorías, tal como se aprecia en la siguiente tabla 2.14. (Cazau, 2005).

Tabla 2.2 Integración de modelos de estilos de aprendizaje

| CATEGORÍAS DE ESTILOS DE APRENDIZAJE | MODELOS TEÓRICOS |
|--|---|
| Selección de la información o tipos de estímulos que generan mayor atención: Estilos visual, auditivo y kinestésico. | Felder y Silverman Programación Neurolingüística |
| Tipo de información desde la cual se prefiere iniciar el proceso: <ul style="list-style-type: none"> • Experiencias directas y concretas, estilo intuitivo y activo • Experiencias abstractas que parten de ideas, estilo sensitivo y teórico. | Felder y Silverman Kolb |
| Procesamiento de la información o forma de organizarla, relacionarla y comprenderla: <ul style="list-style-type: none"> • Estilo secuencial y predominancia cortical y límbica izquierda • Estilo global con predominancia cortical derecha. | Felder y Silverman Hermman |
| La forma de trabajar con la información: <ul style="list-style-type: none"> • Estilo activo y pragmático • Estilo teórico y reflexivo. | Felder y Silverman Kolb |

2.16.2 El Modelo de Aprendizaje Experiencia de Kolb:

Propone un modelo para examinar la forma cómo aprenden los sujetos a partir de la experiencia, y que toma en cuenta los aportes tanto de Jung, Lewin así como de Piaget (Kolb, 1979, 1981), también considera que el aprendizaje constituye un modelo cíclico, con 4 estados básicos o fases. (Mayaute, 1992):

- Experiencia Concreta (EC) de una situación de aprendizaje
- Observación Reflexiva (OR) de un fenómeno relevante
- Conceptualización Abstracta (CA) acerca de lo que significa aquello que es observado
- Experimentación Activa (EA) de las hipótesis planteadas.

A partir de la preferencia por una determinada fase de aprendizaje, Kolb definió los siguientes tipos de Aprendizaje, el cual está basado en el aprendizaje que toma como eje central la experiencia directa del estudiante (Romero, 2010), los cuatro tipos son:

Tabla 2.3 Los estilos de aprender y sus características generales

| ESTILO DE APRENDER | CARACTERÍSTICAS |
|---------------------------|---|
| Divergente | Modalidades EC y OR: Agilidad imaginativa, visualiza situaciones concretas de diversas perspectivas, formula ideas emotivo(a). y se interesa por las personas. |
| Asimilador | Modalidades CA (conceptualización activa) y OR (observación reflexiva): Habilidad para crear modelos teóricos, razonamiento inductivo; le interesan menos las personas y más los conceptos abstractos. |
| Convergente | Modalidades CA y EA: Aplicación práctica de las Ideas, pruebas de Inteligencia de una contestación; soluciona un problema o pregunta; razonamiento hipotético deductivo; poco emotivo(a); prefiere los objetos a las personas. |
| Acomodador | Modalidades EC (experiencia concreta) y EA (experiencia activa): llevar a cabo planes; Involucrarse en experiencias nuevas; arriesgado(a) e intuitivo(a); depende de otras personas; cómodo con la gente. |

De acuerdo con David Kolb (1984), son necesarias dos dimensiones para que ocurra el aprendizaje. La primera es descrita como percepción del medio y la segunda, como procesamiento; de la misma forma como un proceso de cuatro etapas: experiencia concreta/conceptualización abstracta y experiencia activa/observación reflexiva. De la combinación de estas etapas resultan cuatro estilos de aprendizaje: convergente, divergente, asimilador y acomodador, figura 2.36.



Figura 2.36: Estilos de aprendizaje. Fuente: (Romero, 2010)

Concibe el aprendizaje como un proceso de cuatro etapas: experiencia concreta/conceptualización abstracta y experiencia activa/observación reflexiva. De la combinación de estas etapas resultan cuatro estilos de aprendizaje: convergente, divergente, asimilador y acomodador.

2.25.2.1 Criterios de valoración del test LSI I de Kolb

Para cada estudiante que resolvió el test LSI de Kolb (Learning Style Inventory) (Anexo 1), se reporta la puntuación total en cada uno de los siguientes grupos de afirmaciones (Castañeda, 2012).

Tabla 2.4 Ítems test de Kolb según grupo y dimensiones del modelo

| Grupo 1 | Grupo 2 | Grupo 3 | Grupo 4 |
|-------------------|-------------------|-------------------|-------------------|
| 1A | 1B | 2B | 2A |
| 2C | 2D | 3D | 3C |
| 3B | 3A | 4C | 6B |
| 4A | 6C | 6D | 7D |
| 8D | 8C | 8B | 8A |
| 9B | 9A | 9C | 9D |
| Total CE = | Total RO = | Total AC = | Total AE = |

Una vez obtenidas las puntuaciones se hallan las diferencias entre las siguientes puntuaciones como se indica a continuación: AC- CE y AE- RO.

Tabla 2.5 Diferencias de puntuaciones

| | | |
|-------------|-------------------------|---------------|
| Adaptador | Si $AC - CE < -2\gamma$ | $AE - RO > 3$ |
| Convergente | Si $AC - CE > -2\gamma$ | $AE - RO > 3$ |
| Divergente | Si $AC - CE < 2\gamma$ | $AE - RO < 3$ |
| Asimilador | Si $AC - CE > 2\gamma$ | $AE - RO < 3$ |

Estas dimensiones representan dos modos dialécticos de captar y transformar la experiencia. La determinación del estilo de cada sujeto se determinó a partir de las puntuaciones AC-CE y AE-RO y la interceptación de estas puntuaciones en alguno de los cuadrantes que se ilustran en la figura 2.37.

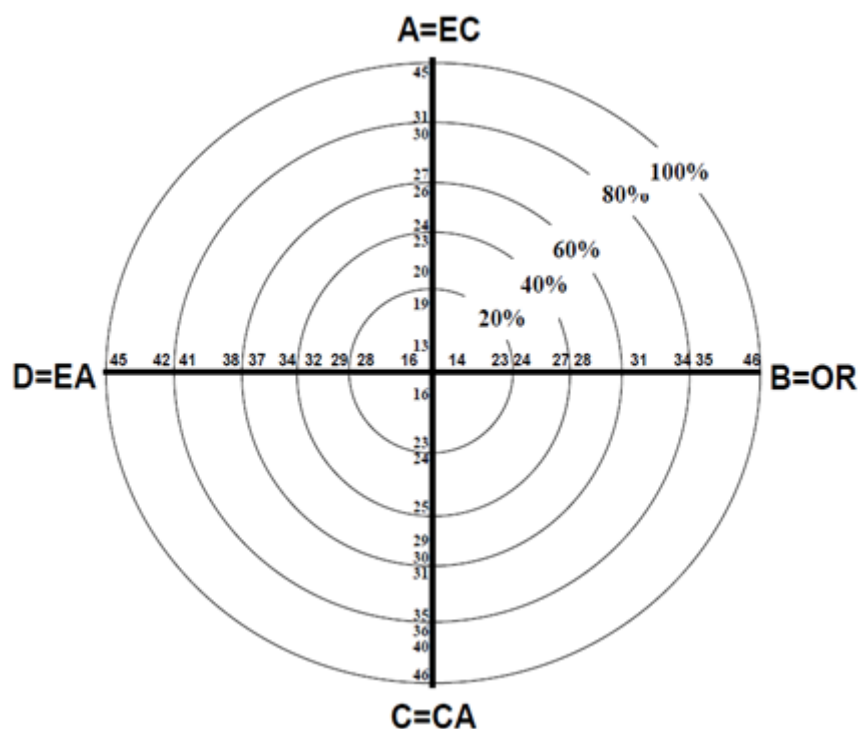


Figura 2.37: Cuadrantes de fases. (Castañeda, 2012)

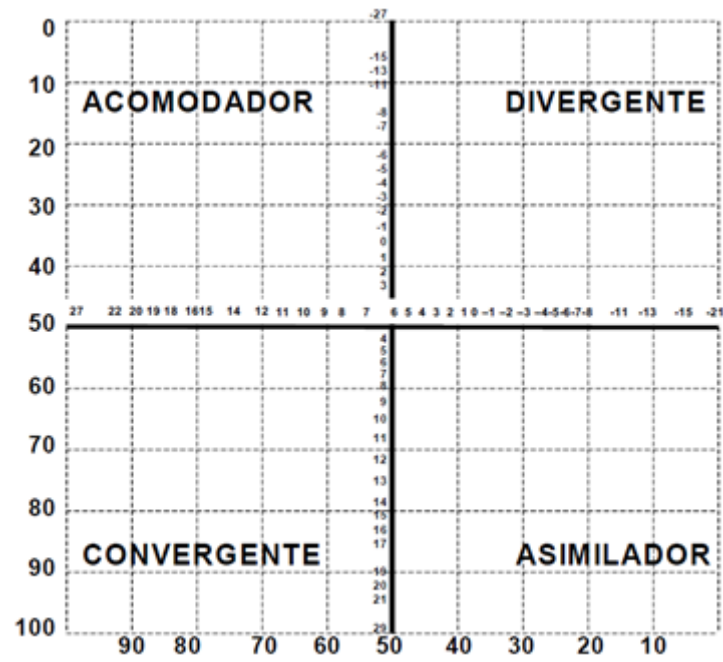


Figura 2.38: Estilo de aprendizaje según puntuación AC-CE y AE-RO. (Castañeda, 2012)

CAPÍTULO III: DIAGNÓSTICO SITUACIONAL

3.1 TRABAJOS QUE RECONOCEN TEXTOS A MANO

Este proyecto trata de utilizar las ventajas de la Inteligencia Artificial y el Deep Learning para el reconocimiento óptico de caracteres, o cualquier otra en la que se necesite distinguir las líneas de un texto manuscrito, como documentos antiguos hasta la traducción simultánea de texto manuscrito (Mendoza Sánchez, 2020), la base de datos utilizada para el entrenamiento ha sido la que se ofreció a los concursantes del ICDAR 2013 Handwriting Segmentation Contest. (Concurso de segmentación de escritura a mano - 150 imágenes escritas en griego e inglés, así como 50 imágenes escritas en idioma bengalí indio), su diseño se basa en la arquitectura de segmentación de líneas de texto utilizando una red totalmente convolucional en imágenes de documentos escritos a mano (Vo, 2017), para ello propone dos fases:

3.1.1. Primero: obtención de la “espina dorsal de las líneas del texto”, que consta:

- a. El primer bloque que encuentra la imagen, es una red completamente convolucional (FCN) con 2 salidas diferentes, se tratan de dos mapas que tratan de predecir la posibilidad de que un determinado píxel del espacio pertenezca al texto o que pertenezca al fondo, en cuanto a la arquitectura utilizada, barajaron la posibilidad de usar la familia de redes FCNpool, todas estas utilizan filtros de 3x3 para aprender los “aspectos tempranos” (early features). La diferencia entre las 3 es que estos aspectos tempranos son compuestos por capas de pooling diferentes.

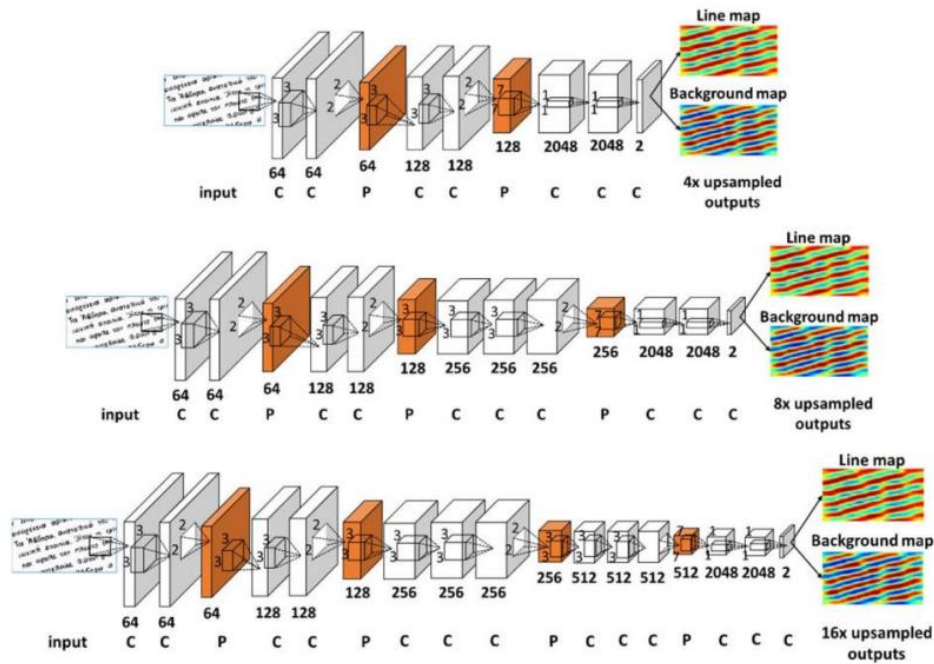


Figura 3.1: De arriba abajo: FCN-pool2, FCN-pool3 y FCN-pool4, (Mendoza Sánchez, 2020)

- b. El segundo paso consiste en deslizar estos dos mapas a lo largo de la imagen e integrar, dando como resultado un nuevo mapa. Las líneas del texto estarán localizadas en las zonas de mayor energía de este mapa.
- c. Lo siguiente que hacen es utilizar un “thresholding algorithm” (algoritmo con umbral) que permite binarizar el mapa de energía. Esto nos da como resultado una nueva imagen donde se puede ver la dirección general principal del texto.
- d. Por último, se debe de sacar el esqueleto o espina dorsal de cada línea de texto. En esto hay un pequeño problema a la hora de discernir entre aquellos mapas de energía binarizados que tienen dos líneas diferentes conectadas entre sí por un error de escritura. Por ello se ejecuta un algoritmo que permite tras varias iteraciones, extraer y fusionar las espinas dorsales.

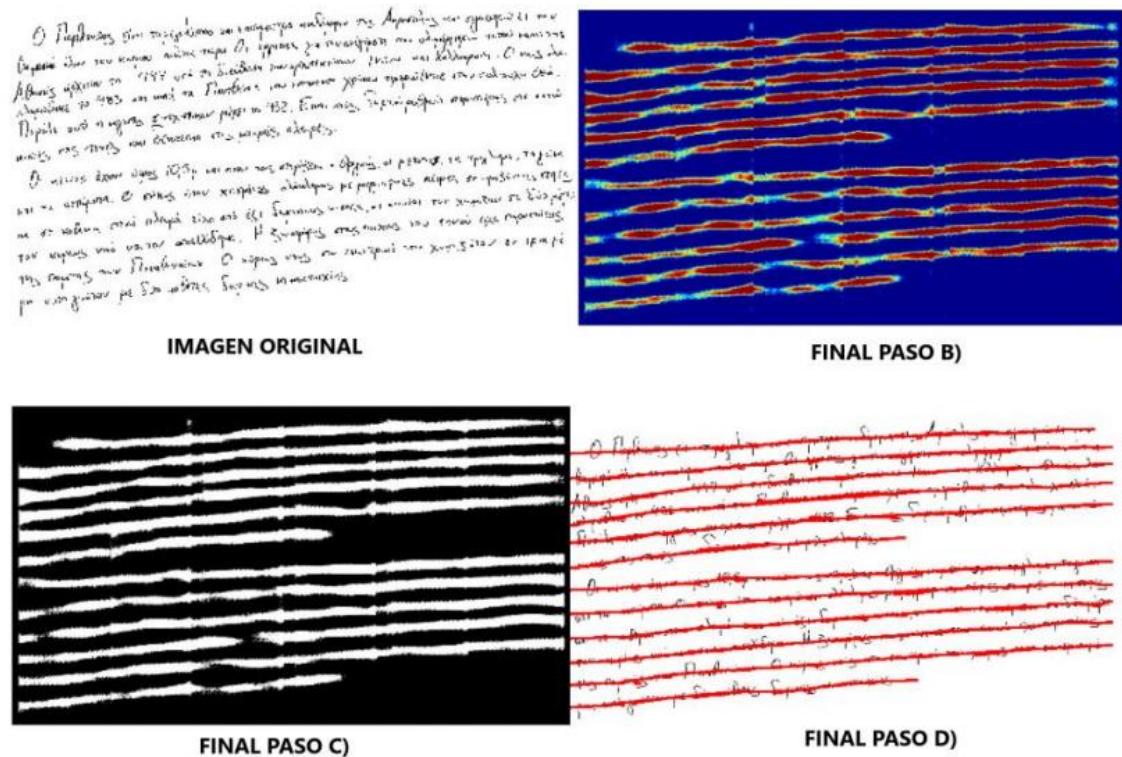


Figura 3.2: Representación de los resultados después de cada paso. (Vo, 2017)

3.1.2. Segundo: Separación de caracteres conflictivos.

Aquellas que por un error durante la escritura del documento se tocan entre sí pero que pertenecen a dos líneas diferentes, para ello se utilizan la línea que surge en el medio de las dos espaldas dorsales para ayudar en su algoritmo de separación.

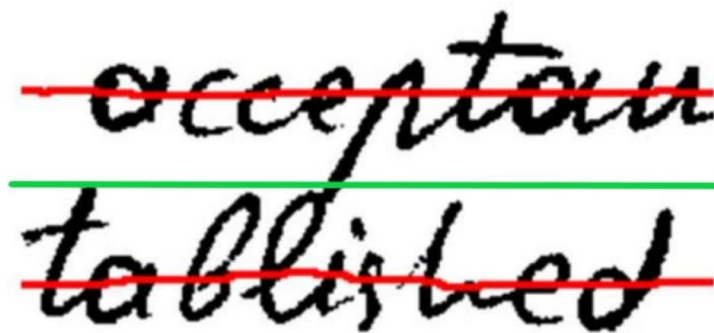


Figura 3.3: Ejemplo de caracteres conflictivos. (Vo, 2017)

El algoritmo de separación que emplean se apoya en la teoría de grafos y en la tecnología LAG – Line adjacency graphs (grafos de líneas adyacentes).

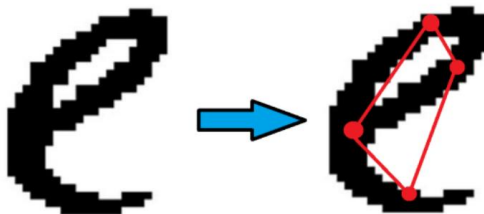


Figura 3.4: LAG de tiradas verticales (grafo rojo). (Vo, 2017)

Se concluye que, aunque el concurso se realizó en 2013 y se utilizó una tecnología mucho más avanzada que la que había por entonces, la red neuronal hubiese con el algoritmo de referencia alcanzo una puntuación calculada de FM=95.60% (métrica final), por lo que concluyen que está entre los mejor que los competidores en el concurso de ICDAR 2013 en la sección de líneas de texto.

3.2 TRABAJOS QUE RECONOCEN OBJETO DIBUJO

Encontrar imágenes naturales de acuerdo con las características y reglas definidas por las personas, la misma imagen dibujada a mano puede representar muchas cosas diferentes, debido a la relación semántica de mapeo de categorías "uno a muchos" entre la imagen dibujada a mano y la imagen natural, que es la inherente ambigüedad de la imagen dibujada a mano. Además, el dibujo de las personas tiene muchas características diferentes, por lo que los resultados de la recuperación generalmente no pueden coincidir completamente con su intención. (Qi, 2019)

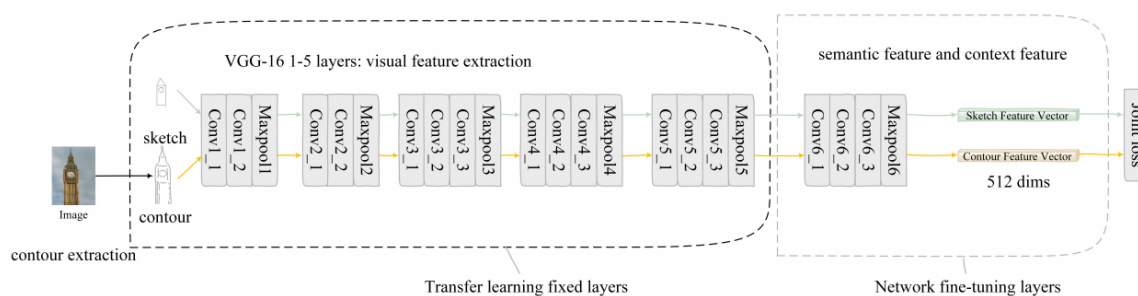


Figura 3.5: Estructura de CNN de dos ramas y diagrama de esquema del modelo de aprendizaje de migración. (Qi, 2019)

3.2.1. Estructura de la red

Para los desafíos mencionados anteriormente, se propone una arquitectura que se muestra en la Fig. 3.5, donde, en la fase de extracción de características, primero establece un modelo de sistema y adopta una estructura CNN de dos ramas. Cada capa de la red comparte parámetros de peso. Por lo tanto, el modelo se puede aproximar como la misma red neuronal que recibe dos datos de entrada al mismo tiempo. Los datos de entrada

secundarios son un par de imágenes dibujadas a mano y una imagen de color, ambas son la matriz. Después del preprocesamiento, la salida final cada vez se empareja con dos vectores propios.

3.2.2. Experimentos y resultados

Como se muestra en la Tabla 3.1, el modelo Deep network (La red profunda) permite extraer características globales e información de contexto, en comparación con los resultados obtenidos de varias redes neuronales poco profundas. Los experimentos muestran que el modelo general tiene una gran capacidad de generalización con una precisión media media de 0,64 en el conjunto de datos de Flickr15 K. El modelo de migración puede realizar una división de espacio vectorial semántica de imágenes de trazos fino, que satisface perfectamente los requisitos de recuperación personalizados mediante la entrada de imágenes basadas en bocetos dibujados a mano.

Tabla 3.1. Comparación de la precisión media (MAP)

| Methods | MAP |
|--------------------------------------|---------------|
| Ours | 0.6449 |
| AFM+QE [18] | 0.579 |
| Triplet(fine-tuned final model) [17] | 0.3617 |
| Sketchy triplet [11] | 0.3591 |
| Query-adaptive re-ranking CNN [9] | 0.3230 |
| Triplet loss CNN [10] | 0.2445 |
| Siamese CNN [8] | 0.1954 |
| PeceptualEdge [12] | 0.1837 |
| GF-HOG [6] | 0.1222 |
| HOG [7] | 0.1093 |
| SIFT [5] | 0.0911 |
| SSIM [4] | 0.0957 |
| ShapeContext [13] | 0.0814 |
| StructureTensor [12] | 0.0798 |

3.3 TRABAJOS QUE RECONOCEN PAISAJES DIBUJADOS A MANO

La creación de un sistema capaz de reconocer automáticamente bocetos de humanos, que es un problema de investigación importante y desafiante. Tiene muchas aplicaciones en la interacción humana con computadora, multimedia y visión por computadora. La mayoría de los métodos anteriores se centran en el reconocimiento de croquis de un solo objeto. En comparación con el reconocimiento de bocetos de un solo objeto, la comprensión del boceto de escena es más desafiante debido a la complejidad, variación e incertidumbre del boceto de escena:

- Complejidad: los bocetos de escenas suelen contener múltiples objetos (Fig. 3-6), lo que introduce una gran dificultad para que la computadora reconozca todos los objetos en una escena y comprenda la información semántica implicada.
- Variación: los dibujantes desde diferentes posiciones de visualización pueden generar bocetos totalmente diferentes de la misma escena (Fig. 3-7: Aula).
- Incertidumbre: mil personas pueden dibujar la misma escena en una mil formas diferentes. Por ejemplo, pueden aparecer diferentes objetos en la misma escena (Fig. 3-7: Desierto). Los bocetos de “Río” pueden ser de varios trazos para representar la forma del río o un retrato con finos detalles de plantas a lo largo del río (Fig. 3-7: Río).

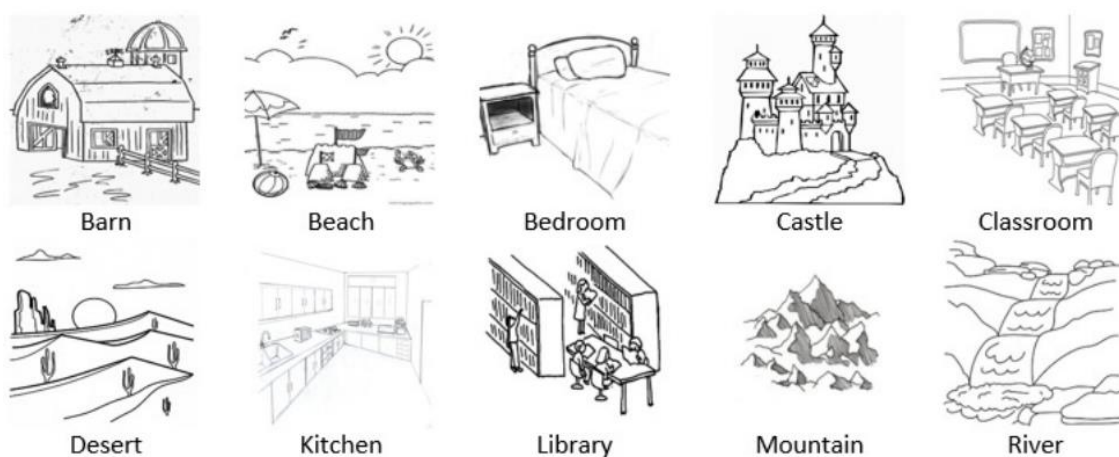


Figura 3.6: Ejemplos de bocetos de escena de nuestro conjunto de datos Scene250 (un ejemplo por categoría). (Ye, 2016)

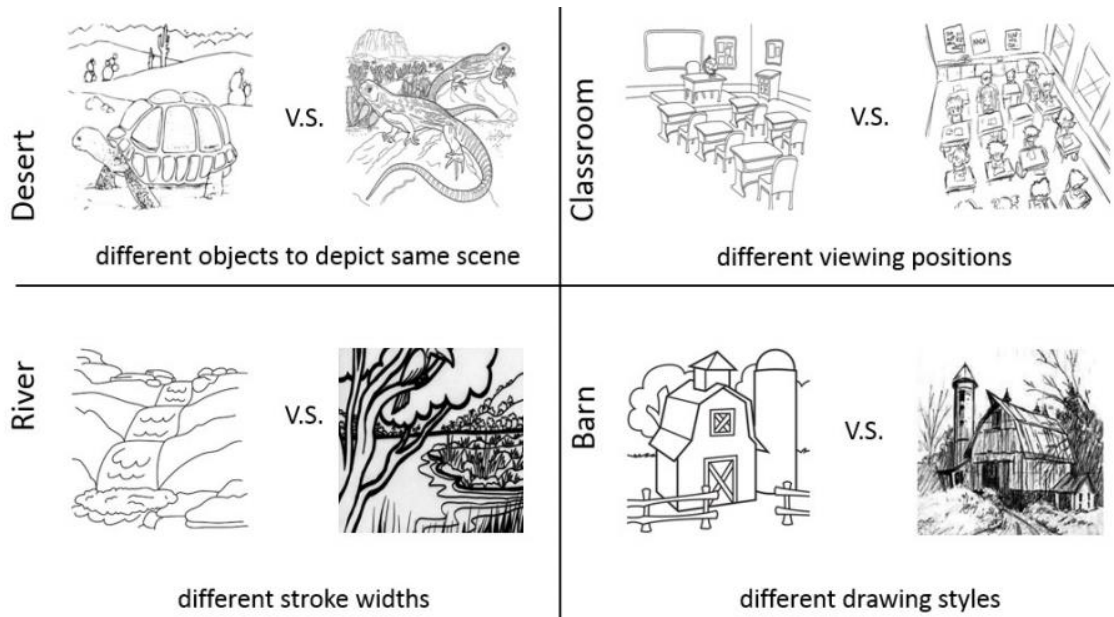


Figura 3.7: Ejemplos para demostrar los desafíos del reconocimiento de bocetos de escenas. (Ye, 2016)

En este trabajo se hace el primer intento por abordar este problema de estudiar la comprensión del boceto de la escena humana; para ellos propone un modelo de red neuronal convolucional profunda (CNN) “Scene-Net” y construye un novedoso sistema de reconocimiento de bocetos de escenas basado en este modelo. El sistema ha sido probado en el conjunto de datos recopilados de bocetos de escenas y comparado con otros enfoques de reconocimiento de bocetos y CNN de última generación (Ye, 2016).

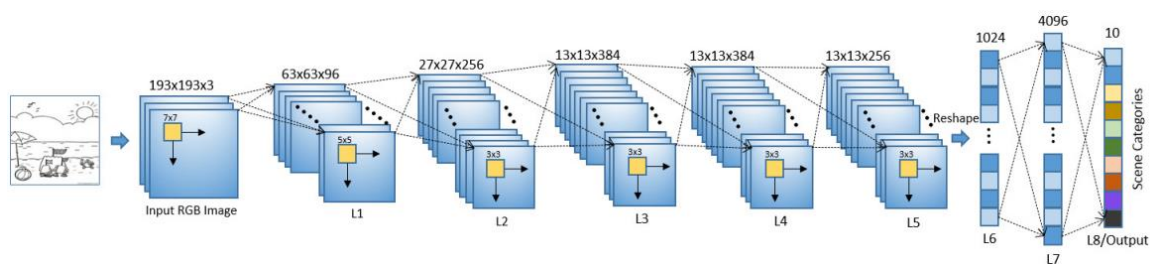


Figura 3.8. Arquitectura general del modelo Scene-Net. (Ye, 2016)

3.3.1. Metodología

En este trabajo se propone la metodología el modelo Scene-Net. que sigue el esquema de diseño utilizado comúnmente en otras CNN modernas, emplea características novedosas, p. Ej. ajuste fino, reducción eficiente de sobreajuste y mejora el rendimiento de reconocimiento.

Propone un modelo de CNN “Scene-Net”, cuya arquitectura se ilustra en la Fig. IV.1, mientras que los parámetros más detallados se pueden encontrar en la Tabla IV.1. Como se muestra en la Fig. IV.1; la misma contiene cinco capas convolucionales (L1~L5) seguidas de tres capas completamente conectadas (fully-connected) (L6~L8). Cada capa tiene un ReLU Excepto la capa L8.

Tabla 3.1 Arquitectura detallada de Scene-Net

| Index | Layer | Type | Filter Size | Filter Num | Stride | Pad | Output Size |
|-------|-------|--|-------------|------------|--------|-----|-------------|
| 0 | | Input | - | - | - | - | 193 × 193 |
| 1 | L1 | Conv | 7 × 7 | 96 | 3 | 0 | 63 × 63 |
| 2 | | ReLU | - | - | - | - | 63 × 63 |
| 3 | | LRN($n = 5, \alpha = 10^{-4}, \beta = 0.75$) | - | - | - | - | 63 × 63 |
| 4 | | Maxpool | 3 × 3 | - | 2 | 0 | 31 × 31 |
| 6 | L2 | Conv | 5 × 5 | 256 | 1 | 0 | 27 × 27 |
| 7 | | ReLU | - | - | - | - | 27 × 27 |
| 8 | | LRN($n = 5, \alpha = 10^{-4}, \beta = 0.75$) | - | - | - | - | 27 × 27 |
| 9 | | Maxpool | 3 × 3 | - | 2 | 0 | 13 × 13 |
| 10 | L3 | Conv | 3 × 3 | 384 | 1 | 1 | 13 × 13 |
| 11 | | ReLU | - | - | - | - | 13 × 13 |
| 12 | L4 | Conv | 3 × 3 | 384 | 1 | 1 | 13 × 13 |
| 13 | | ReLU | - | - | - | - | 13 × 13 |
| 12 | L5 | Conv | 3 × 3 | 256 | 1 | 1 | 13 × 13 |
| 13 | | ReLU | - | - | - | - | 13 × 13 |
| 14 | | Maxpool | 3 × 3 | - | 2 | 0 | 6 × 6 |
| 15 | L6 | Fully-connected | 6 × 6 | 1024 | 1 | 0 | 1 × 1 |
| 16 | | ReLU | - | - | - | - | 1 × 1 |
| 17 | | Dropout(0.5) | - | - | 1 | 0 | 1 × 1 |
| 18 | L7 | Fully-connected | 1 × 1 | 4096 | 1 | 0 | 1 × 1 |
| 19 | | ReLU | - | - | - | - | 1 × 1 |
| 20 | | Dropout(0.5) | - | - | 1 | 0 | 1 × 1 |
| 18 | L8 | Fully-connected | 1 × 1 | 10 | 1 | 0 | 1 × 1 |
| 19 | | Softmax loss | - | - | - | - | 1 × 1 |

3.3.2. Funciones novedosas en la arquitectura Scene-Net.

Características novedosas aplicadas en Scene-Net. Aunque pocos trabajos de investigación previos de CNN discuten en detalle por qué establecer un parámetro usando un valor específico, y también es imposible verificar exhaustivamente la efectividad de cada hiperparámetro, se propone puntos que son diferentes a la configuración de los modelos profundos de CNN convencionales, también presenta el ajuste fino, una técnica

muy importante para aliviar el problema de sobreajuste en nuestra pequeña escala Scene250.

3.3.2.1. Tamaño de entrada más pequeño.

El tamaño de entrada en CNN profundo es uno de los parámetros más críticos, ya que todas las capas siguientes dependen de la capa de entrada, se encontró que un tamaño de entrada más pequeño es más apropiado para el reconocimiento de bocetos de escenas. La razón principal es que los bocetos de escenas suelen contener varios objetos dentro de una imagen. Un tamaño de entrada más pequeño combinado con un aumento de datos aleatorio puede capturar características más sofisticadas. Con este fin, se utiliza un tamaño de entrada de 193×193 en Scene-Net.

3.3.2.2. Max Pooling superpuesta.

Algunas CNN profundas recientes emplean Max Pooling de 2×2 con paso 2. Este tipo de agrupación máxima no superpuesta puede generar cierta invariancia espacial y reducir el tamaño de la capa en un 75%. Sin embargo, la agrupación máxima superpuesta, aplicada en el modelo Scene-Net, aporta una mejora para el sistema de reconocimiento y reducir el tamaño de la capa en un 50%.

3.3.2.3. Afinación Fina.

El ajuste fino es un método de uso común en estudios profundos de CNN para evitar el ajuste excesivo en conjuntos de datos a pequeña escala. Pre-entrena un modelo de CNN en un gran conjunto de datos auxiliares y aprende los pesos iniciales del modelo. Luego, reanuda el proceso de entrenamiento del mismo modelo de CNN en el conjunto de datos de destino (generalmente uno pequeño) con los pesos iniciales aprendidos.

En la propuesta planteada, primero se entrenó previamente el modelo Scene-Net en el conjunto de datos de “TU Berlin” y se obtuvo los pesos de aprendizaje iniciales para el modelo Scene-Net. Luego, se reanuda el proceso de entrenamiento en el conjunto de datos Scene250 para ajustar el modelo Scene-Net para la tarea de reconocimiento de bocetos

de escenas. Debido a que se quiso predecir 10 categorías de escenas en el paso de ajuste fino en lugar de 250 categorías de objetos en el entrenamiento previo, se cambió el número de salida de la última capa de 250 a 10.

También se disminuyó la tasa de aprendizaje para ralentizar el entrenamiento afinado. El motivo es que, después del entrenamiento previo, se obtuvo el modelo de clasificación estable y solo se quiso aplicar este modelo a una nueva tarea de reconocimiento, por lo tanto, la tasa de aprendizaje debe reducirse para evitar cambiar su estabilidad. En los experimentos, se encontró que la aplicación de un ajuste fino aumentaría significativamente la precisión del reconocimiento y reduciría el problema de sobreajuste.

3.4 TRABAJO PARA GENERAR ESTRATEGIAS DE ENSEÑANZA Y MOTIVACIONES QUE ESTIMULEN EL APRENDIZAJE.

La investigación se llevó en el Programa de Diseño Gráfico de la Facultad de Arquitectura y Diseño de la Universidad del Zulia, para evaluar los estilos de aprendizaje presentes en los estudiantes adscritos al programa de Diseño Gráfico y generar estrategias de enseñanza y motivacionales que estimulen su aprendizaje, para ello se aplican el Test de Kolb (1985), el formulario fue adaptado y alojado de forma asíncrona en el servicio Google Forms en el dominio de las cuentas educativas institucionales de la Universidad del Zulia (fad.luz.edu.ve).

Luego de aplicar el Test de Kolb se procedió a identificar los modelos de aprendizaje y la más predominante de los estudiantes del Programa de Diseño Gráfico, es el modelo de aprendizaje de la etapa Observación reflexiva (OR), al calcular la frecuencia de aparición se obtuvo un porcentaje general de 44,9% y si consideramos solo las modalidades de Procesamiento es de un 89,8%; de igual forma el modelo de aprendizaje Conceptualización Abstracta (CA) con un porcentaje general de 31,4% al sólo observar las frecuencias para la modalidad de Percepción, la CA cuenta con un 62,7%.

Cuando se procedió a identificar el estilo de aprendizaje, la mayoría de ellos se inclina hacia el estilo Asimilador con un 51%, este estilo de aprendizaje posee un enfoque conciso, lógico y preciso; quienes lo poseen, se destacan por su capacidad para comprender la información, organizándola con un formato claro y lógico.

El segundo estilo de aprendizaje que prevalece es Divergente con un 32%, seguido con porcentajes menores, los estilos de aprendizaje Convergente y Acomodador con un 14% y 3% respectivamente, Por otro lado, se aclara que los estudiantes con estilo Divergente prefieren observar lo que sucede más que actuar, tienen habilidad para observar un mismo fenómeno desde diversas perspectivas. Del mismo modo, pueden escuchar con amplitud mental, considerando distintos puntos de vista sobre una misma cuestión. (Ortega, 2019)

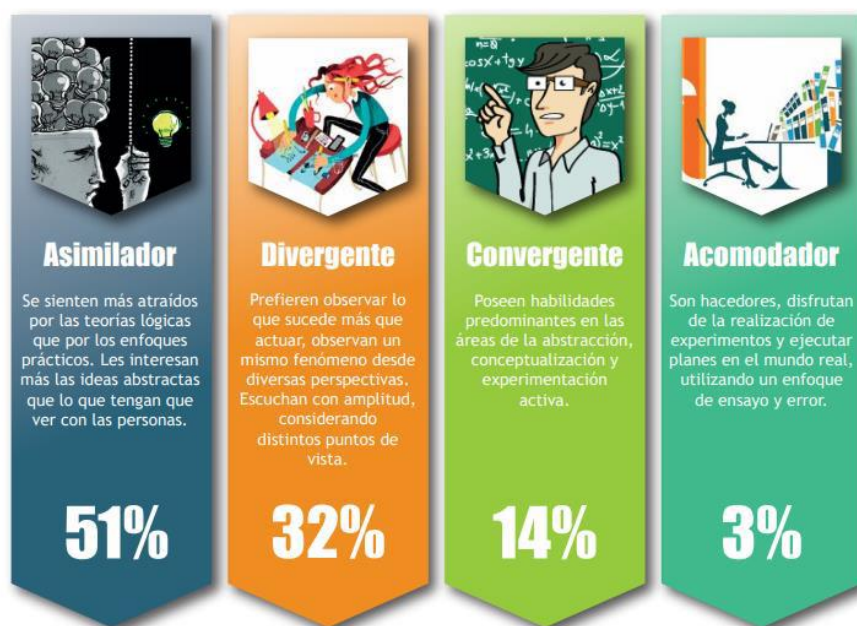


Figura 3.9: Frecuencia por Estilo de Aprendizaje con base en el Modelo de Kolb de los estudiantes de Diseño Gráfico de FADLUZ. (Ortega, 2019)

3.5 ANÁLISIS DE TRABAJOS RELACIONADOS

Tabla 3.3. comparativa de trabajos relacionados con la investigación

| Nº | AUTOR(AÑO) | TEMA | METODOLOGÍA | PRESIÓN % | DATA BASE |
|----|---|---|---|-----------|--|
| 1 | Vo, QN, Kim, SH, Yang, HJ y Lee, GS. (2017) | Segmentación de líneas de texto utilizando una red totalmente convolucional en imágenes de documentos escritos a mano | arquitectura de segmentación de líneas de texto utilizando una red totalmente convolucional en imágenes de documentos escritos a mano | 95.60 % | ICDAR 2013 Benchmarking Dataset. (150 imágenes escritas en griego e inglés, así como 50 imágenes escritas en |

| | | | | | |
|---|--|--|--|--|--|
| | | | | | idioma bengalí indio) |
| 2 | Qi, Q., Huo, Q., Wang, J., Sun, H., Cao, Y. y Liao, J. (2019) | Recuperación de imágenes personalizada basada en bocetos mediante red neuronal convolucional y aprendizaje de transferencia profunda | modelo Deep network (La red profunda) | Precisión 64 % | Flickr15 K (330 bocetos dibujados a mano, cada imagen natural pertenece a una subcategoría definida, cada subcategoría tiene un total de 60 según la división semántica) |
| 3 | Ye, Y. (2016) | Aplicación del aprendizaje profundo al reconocimiento de bocetos de escenas y la recuperación de modelos 3D basados en bocetos. | modelo de CNN "Scene-Net" M | Rendimiento: 66.67 % | Dataset "Scene250" (250 bocetos de escenas en 10 categorías, que consta de 4 categorías interiores comunes y 6 al aire libre comunes.) |
| 4 | Ortega, E., Romero, I. I. C., Paredes, Í., & Canquiz, L. (2019). | Estilos de aprendizaje: estrategias de enseñanza en LUZ. | Test Estilos de aprendizaje de Kolb (1985) | Asimilador con un 51% seguido del Divergente con un 32%. | Base de Datos generada por Google Forms con los puntajes para cada modalidad de aprendizaje y puntajes combinados. |

3.6 CONSIDERACIONES FINALES

A partir de los artículos evaluados, en detección de líneas de documentos escaneados, presentan una red completamente convolucional (FCN) para predecir la estructura de las líneas de texto en imágenes de documentos, obteniendo una estimación aproximada de líneas de texto; a partir del cual se construyen las cadenas de texto que pasan por los caracteres en cada línea de texto; luego los caracteres en contacto deben separarse y asignarse a diferentes líneas de texto para completar la segmentación.

En el caso de recuperación de imágenes personalizada basada en bocetos, se tiene en cuenta, las imágenes naturales de acuerdo con las características y reglas definidas por los seres humanos, la misma imagen dibujada a mano puede representar muchas cosas diferentes, debido a la relación semántica de mapeo de categorías "uno a muchos" entre la imagen dibujada a mano y la imagen natural, que es la ambigüedad inherente de imagen dibujada a mano. Para ello, se propone una arquitectura de recuperación de imágenes basada en bocetos (SBIR) personalizada, incluyendo una red neuronal convolucional profunda (deep full convolutional neural network) y completa como modelo general y un modelo personalizado que utiliza el aprendizaje de transferencia (transfer learning) para lograr una característica semántica de imagen de línea fina (achieve fine-grained). Los experimentos realizados muestran que el modelo tiene una gran capacidad de generalización con una precisión media de 0,64 en el conjunto de datos de Flickr15K.

La mayoría de los métodos anteriores se centran en el reconocimiento de bocetos de un solo objeto, en este trabajo, se hace el estudio de bocetos de escenas a mano alzada; para lo cual se crea un dataset de bocetos de escenas "Scene250" y se explora el esquema de aprendizaje profundo para comprender los bocetos de escenas. Se propone un modelo de red neuronal convolucional (CNN) profunda "Scene-Net" y construye un novedoso sistema de reconocimiento de bocetos de escenas basado en este modelo, los resultados experimentales demuestran la eficacia de predecir categorías de las escenas de bocetos externos e internos.

Por otro lado, la investigación realizada a los estudiantes de Diseño Gráfico de FADLUZ de los estilos de aprendizaje, resalta el modelo de aprendizaje de Observación reflexiva

(OR) en un 89,8%, de igual forma el modelo de aprendizaje Conceptualización Abstracta (CA) en un 62,7%. Cuando se procedió a identificar el estilo de aprendizaje, la mayoría de ellos se inclina hacia el Estilo Asimilador con un 51%, y al Estilo de aprendizaje Divergente con un 32%, seguido con porcentajes menores, los estilos de aprendizaje Convergente y Acomodador con un 14% y 3% respectivamente.

Por otro lado, se aclara que los estudiantes con Estilos Asimilador (EA) posee un enfoque conciso, lógico y preciso; quienes lo poseen, se destacan por su capacidad para comprender la información, organizándola con un formato claro y lógico. De igual forma los de Estilo Divergente (ED) prefieren observar lo que sucede más que actuar, tienen habilidad para observar un mismo fenómeno desde diversas perspectivas, también pueden escuchar con amplitud mental, considerando distintos puntos de vista sobre una misma cuestión.

En la presente investigación, se tendrá en cuenta estos criterios a la hora de procesar los dibujos a mano alzada de los estudiantes aplicados el Test y generar una base de datos para la red neuronal a implementar con la metodología de Deep Learning. También cabe precisar, de los artículos analizados podemos deducir que, no se encontró trabajos de investigación que detecten estilos de aprendizajes a partir de una imagen, ni mucho menos de dibujos a mano alzada; solo detección de líneas de texto de documentos escaneados, detección y selección de imágenes semánticas a partir de un boceto y en caso bocetos de escenas se implementó arquitecturas para la detección de categorías internas y externas.

CAPÍTULO IV: PROPUESTA PLANTEADA

En este capítulo se describe la propuesta del proyecto que busca predecir estilos de aprendizaje de los estudiantes de High School a partir de imágenes a mano alzada, para ello se consideró la parte educativa referente a los estilos de aprendizaje de Kolb y en la parte computacional la implementación de la metodología de Deep Learning.

4.1 PROPUESTA EDUCATIVA

Se describe las estrategias y procedimientos desarrollados para la selección de los Estilos de aprendizaje mediante el Test de Kolb y la propuesta de dibujo de un paisaje luego de la aplicación del Test.

En la aplicación de Test se utilizó el Inventario de Estilos de aprendizaje de Kolb adaptado en una Ficha “Test de Estilos de Aprendizaje” Fig. 4.1, de igual forma se construyó un formulario en Google Forms (<https://forms.gle/XqFu2rY76nFrCa5cA>), cuyos resultados se muestran en una base de datos automática generada por la aplicación y disponible en una Hoja de Cálculo en Google Docs, Fig. 4.3.

Luego de finalizar el llenado del Test (inventario de Estilos de aprendizaje de Kolb) se le solicitó a los estudiantes que realicen un dibujo de un paisaje Fig. 4.2

Figura 4.1. Test completado por un estudiante de la muestra (Inventario de Estilos de aprendizaje de Kolb). (Elaboración propia)



Figura 4.2. Dibujo de una casa en un paisaje, por el estudiante que completo el Test. (Elaboración propia)

En la Fig. 4.3, se observan la base de datos generada por Google Forms en las cuales se procedió a introducir columnas con las fórmulas que permitieron obtener las

Por otro lado la investigación realizada a los estudiantes de Diseño Gráfico de FADLUZ de los estilos de aprendizaje (Ortega, 2019), se concluye que el modelo de aprendizaje de Observación reflexiva (OR) resaltó en un 89,8%, donde la mayoría de ellos se inclina hacia el Estilo de aprendizaje Asimilador con un 51% y Divergente con un 32%, lo cual permitió orientar estrategias con un alto nivel de objetividad y practicidad en los procesos de diseño.

Dicha investigación servirá de referencia para implementar la base de datos de este proyecto de investigación, según los resultados la mayoría de ellos se inclina hacia los Estilo de aprendizaje Asimilador (EA) y Divergente (ED); quienes lo poseen se destacan por su capacidad para comprender la información, organizándola con un formato claro y lógico, crean modelos y son precisos, tienen habilidad para observar un mismo fenómeno desde diversas perspectivas, considerando distintos puntos de vista sobre una misma cuestión y son creativos; a comparación de estilos de aprendizaje Convergente y Acomodador con porcentajes menores de 14% y 3% respectivamente.

Por ello, para fines de investigación las imágenes se agruparán en imágenes “RETAILER” que agrupara a los estilos de aprendizaje (asimilador y divergente) y de igual forma en imágenes “SIMPLISTIC” agrupando los estilos de aprendizaje (convergente y acomodador).

4.2 PROPUESTA COMPUTACIONAL

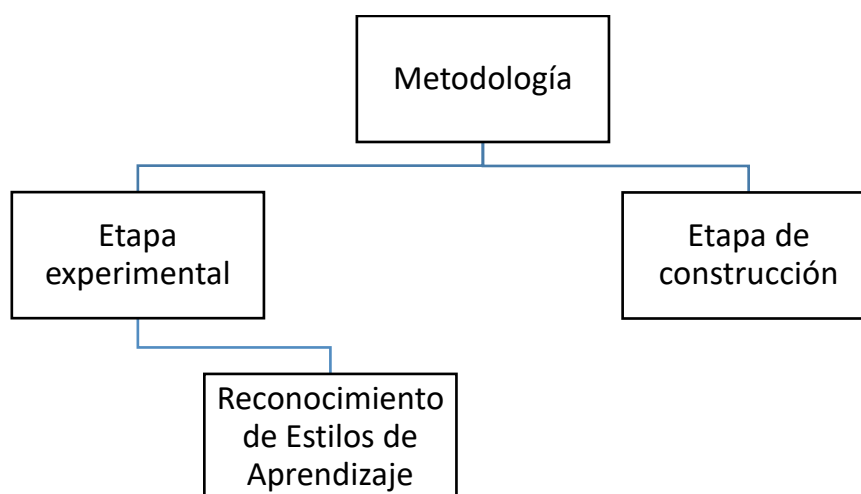


Figura 4.5. Metodología empleada. (Elaboración propia)

4.2.1 Etapa experimental:

En esta etapa experimental se describen los procedimientos de experimentación a llevarse a cabo para la clasificación y predicción de los estilos de aprendizaje; esta etapa experimental consiste en cinco pasos:

- Implementación de la base de datos
- Pre_procesamiento de datos
- Extracción de características
- Clasificación de EA (CNN)
- Evaluación

a. Implementación de la base de datos

Se selecciona en carpetas etiquetadas ejemplos de estilos de aprendizaje clasificados y predichos, que servirá de entrenamiento para el aprendizaje de los modelos, para lo cual se toma en cuenta:

- Tipo de imagen: Las imágenes deben tener los elementos mínimos necesarios que permitan percibir un paisaje de un tipo de estilo de aprendizaje.
- Diversidad: Es imprescindible obtener una variedad de imágenes distintas por cada estilo de aprendizaje.
- Etiquetado: Por ser un tipo de aprendizaje supervisado, las imágenes serán etiquetadas.
- Cantidad de estilos: Los estilos de aprendizaje a considerar son Divergente (ED), Asimilador (EA), Convergente (EC), Acomodador (EA).

b. Pre-procesamiento de datos: Se requiere preparar todas las imágenes para facilitar su procesamiento, como:

- Escala de color: las imágenes de color pasan a escala gris para reducir a un solo plano y minimizar recursos.
- Detección de paisaje: Se requiere ubicar el paisaje en sí, es imprescindible minimizar los datos a analizar y eliminar datos irrelevantes.
- Recorte: Con el propósito de eliminar ruidos y reducir recursos, la imagen se recortará de tal manera que solo se utilizará el paisaje en sí.
- Estandarizar imagen: se normaliza a un mismo tamaño todas las imágenes.

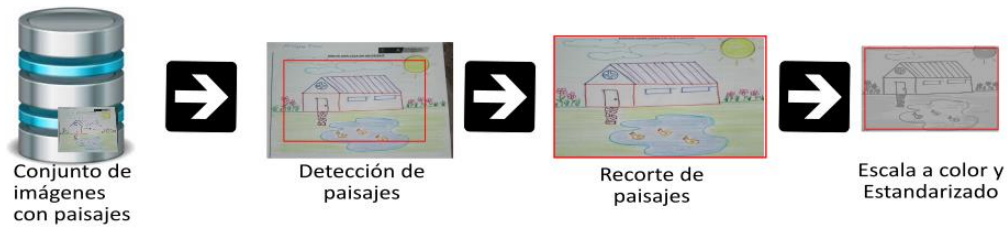


Figura 4.6. Ejemplos de tareas en la etapa del pre-procesamiento de datos. (Elaboración propia)

- c. Extracción de características: Para ello se utilizará una convolucional Neural Network (CNN) para la extracción de características, según los siguientes puntos:
- Entrada: la imagen que solo contiene el paisaje a analizar con un tamaño cuadrado $n \times n$.
 - Procesamiento: la imagen que pasa por una serie de transformaciones conformado por convoluciones y submuestreos.
 - Salida: Se obtiene un vector ($1 \times m$) que resumen las características discriminativas de la imagen.

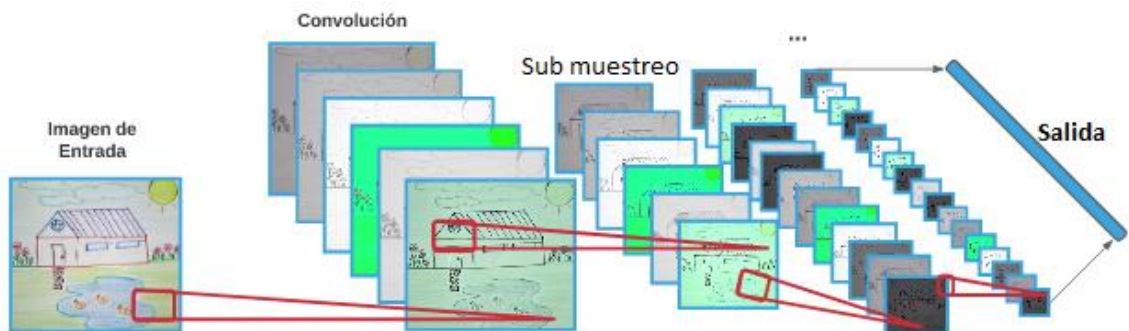


Figura 4.7. Arquitectura general de CNN para la extracción de características de una imagen. (Elaboración propia)

- d. Clasificación:

los vectores de características de cada imagen obtenidos en el paso anterior, sirven como datos de entrenamiento para los algoritmos de aprendizaje, teniendo como objetivo el de aprender modelos capaces de clasificar correctamente nuevas imágenes de estilos de aprendizaje, para llevar a cabo este paso se describe los siguientes puntos:

- Entrada: vectores de características de cada imagen del conjunto de ejemplos de entrenamiento.

- Procesamiento: vectores de características pasa por un algoritmo de aprendizaje que sirve como entrenamiento para conocer y aprender el problema.
- Salida: crea un modelo capaz de clasificar correctamente nuevos ejemplos, sin haberlos usado en el entrenamiento.



Figura 4.8. Marco de trabajo para la clasificación. (Elaboración propia)

- e. Evaluación: por último, se evalúa el modelo aprendido usando conjuntos de imágenes de pruebas, además de aplicar medidas de evaluación. Para llevar a cabo este paso, se sigue los siguientes puntos:
- Entrada: conjunto de imágenes de prueba etiquetadas.
 - Proceso: las imágenes pasan por el modelo aprendido y son clasificadas.
 - Evaluación: verificación de las clases proporcionadas por el modelo aprendido para cada imagen con su correspondencia a las etiquetas.

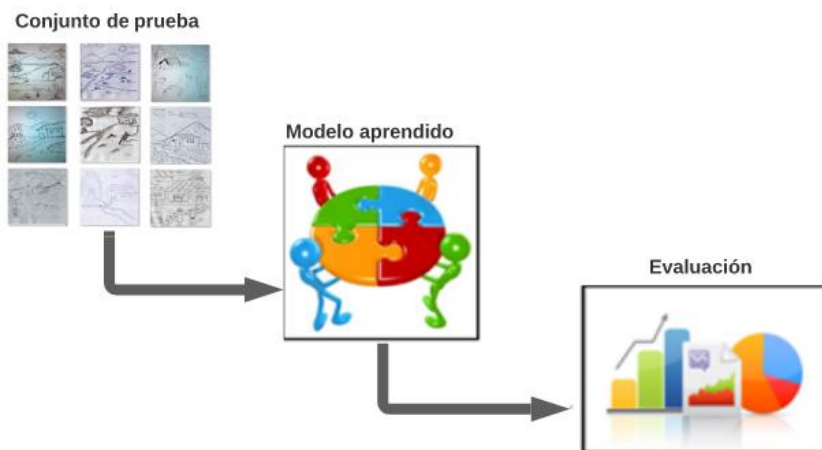


Figura 4.9. Marco de trabajo para la evaluación. (Elaboración propia)

4.2.2 Método para el reconocimiento de Estilos de Aprendizaje de imágenes de paisajes

Empleando la metodología de la etapa experimental (4.2.1.), a continuación, se describen los pasos esenciales para llevar a cabo esta etapa:

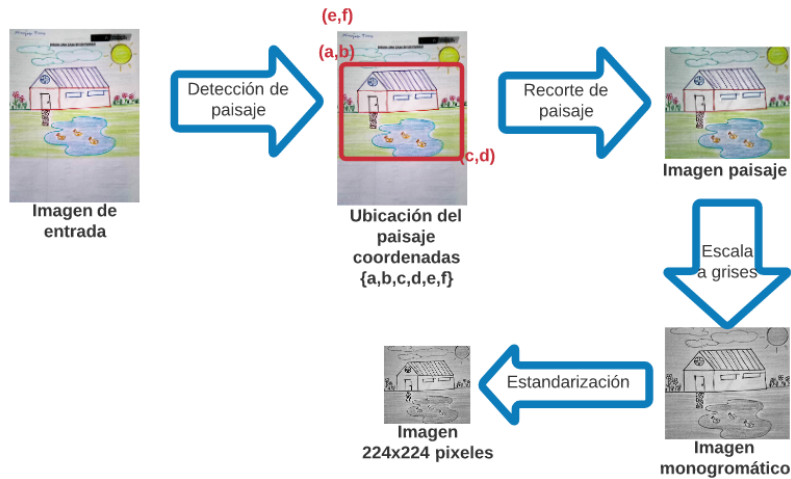


Figura 4.10. Marco de trabajo usado para el pre-procesamiento de imágenes en el reconocimiento de expresiones faciales. (Elaboración propia)

4.2.2.1. Pre-procesamiento de imágenes:

La imagen digital es procesada por un algoritmo de procesamiento *torchvision*, esta biblioteca es parte del proyecto PyTorch, este paquete consta de conjuntos de datos populares, arquitecturas de modelos y transformaciones de imágenes comunes para la visión por computadora, donde se realizó las transformaciones a la imagen para los datos de entrenamiento y los datos de prueba.

4.2.2.2. Extracción de características:

Se utiliza DenseNets con la finalidad de explotar el potencial de la red a través de la reutilización de características de la imagen, produciendo modelos condensados que son fáciles de entrenar y altamente eficientes en parámetros.

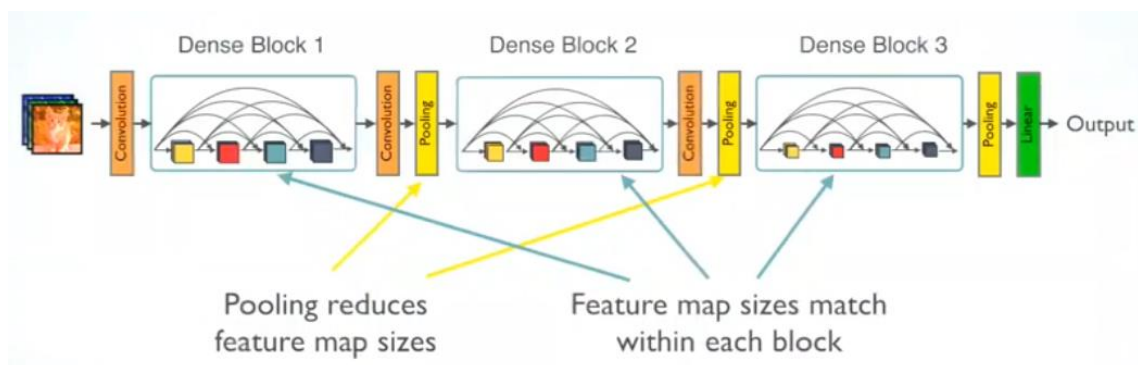


Figura 4.11. Arquitectura DenseNet, https://github.com/karthikziffer/Keras_Densenet_CIFAR10

En la Fig. 4.11. se muestra gráficamente su proceso, donde el dato de entrada es la imagen digital obtenida en el paso anterior, la imagen pasa por la Red convolucional densa, normalización por lotes (BN) y unidad lineal rectificada (ReLU)

4.2.2.3. Clasificación

Se emplean algoritmos de aprendizaje (*DenseNets, matplotlib.pyplot, numpy, torch, torchvision, torch.nn.functional, datasets, transforms, models*) y otros

4.2.2.4. Evaluación:

Se utilizó la validación cruzada en los subdirectorios de Train y Validation, repitiendo tres veces la validación modificada, usando las medidas de exactitud, precisión, para evaluar el desempeño.

4.3 ETAPA DE CONSTRUCCIÓN DE LA PROPUESTA

En este trabajo para el reconocimiento de estilos de aprendizaje en dibujos a mano alzada, proponemos la Dense Convolutional Network (DenseNet), que conecta cada capa con todas las demás en una forma de retroalimentación.

Es por ello, proponemos la arquitectura DenseNET para la tarea de reconocimiento de dibujos de paisajes libres a mano alzada, implementada en la plataforma de colaboratory, para lograr un alto rendimiento a comparación de otras arquitecturas entrenadas CNN y ResNet.

4.3.1 Implementación de la propuesta

La Fig. 4.12, muestra la propuesta de investigación, donde el *DenseNet* se utiliza para entrenar una red para pronosticar estilos de aprendizaje a partir del paisaje de bocetos.

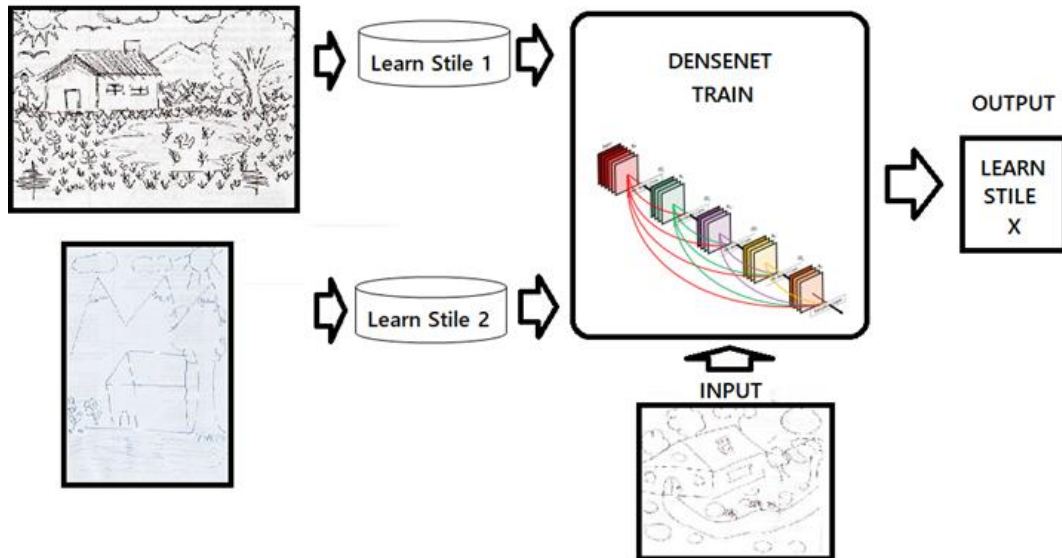


Figura 4.12: Propuesta de investigación: La red Densenet se utiliza para aprender estilos de aprendizaje a partir de una base de datos de bocetos de estudiantes. (Elaboración propia)

CAPÍTULO V: PRUEBAS Y RESULTADOS

5.1 BASE DE DATOS (DATASET)

Para la investigación se utilizó la base de datos de imágenes dibujadas a mano alzada sin ninguna precisión de técnicas de dibujo a mano alzada, por parte de los estudiantes de High School, luego de completar el inventario de estilos de aprendizaje de Kolb.

Esta base de datos contiene imágenes que tienen diferentes características entre sí, como resoluciones y formatos diferentes las cuales fueron clasificados según los estilos de aprendizaje; lo resaltante de esta base de datos es realizar la evaluación de la propuesta con imágenes inéditas de diferentes calidades. En la Figura 4.02 se observa ejemplos de la base de datos.

5.1.1 Conjunto de datos de estilos de aprendizaje

Para la presente investigación se utilizó la colección de 500 imágenes de dos grupos de estilos de aprendizaje diferentes, contando con 250 imágenes de cada estilo..., lo que corresponde a los dibujos de estudiantes de nivel elemental de la escuela XXI. Luego de completar el inventario de estilos de aprendizaje de Kolb; las imágenes fueron realizadas artesanalmente en hoja A4, donde se les pidió a los estudiantes que dibujaran un paisaje, lo importante es que no se les dio ninguna otra indicación, dejándoles libres para elegir el lugar del entorno e inspiración, luego se escanearon las imágenes y guardadas en formato “JPG” utilizando el programa editor de imágenes Paint y fueron etiquetadas en los dos grupos de estilos, siendo el primer grupo RETAILER (Divergente-Asimilador) y el segundo grupo SIMPLISTIC (Convergente-Acomodador), ver Fig. 5.1.



(a)



(b)

Figura 5.1: Ejemplos de la base de datos de los dos estilos donde (a) Grupo RETAILER (Divergente-Asimilador) y (b) Grupo SIMPLISTIC (Convergente-Acomodador). Elaboración propia.

En la tabla 5.1 se muestra como se ha distribuido la base de datos luego de aplicar el algoritmo de Data Augmentation (aumento de datos) y el Target Size (dimensión), de igual forma la cantidad de imágenes de entrenamiento (Training) y la cantidad de imágenes de validación (Validation) y cantidad de imágenes de pruebas (Test).

Tabla 5.1 Distribución

| Base de dato | Dimensión | Training | Validation | Test | Total |
|----------------|-----------|----------|------------|------|-------|
| Imagen de E.A. | 225x225 | 2224 | 880 | | 3104 |

5.2 PRUEBAS

Como se detallo anteriormente se tiene una base de datos, que tiene imágenes de dibujos a mano alzada de diferentes estilos de aprendizajes de estudiantes de High School, donde se tuvo dificultades al momento de realizar las pruebas, porque las imágenes no se dibujaron de forma adecuada; además de haber dibujos detallistas y simples en su diseño cuya resolución dependía mucho del tipo de instrumento utilizado como lápiz, lapicero y colores. En la Figura 5.2 se puede ver el resultado de la inferencia del modelo entrenado el grupo de estilo DETALLISTA, donde se puede observar claramente que las imágenes tienen mayores características en sus diseños. La evidencia de la deducción es lo esperado, la cual corresponde al grupo de estilo DETALLISTA (Divergente y Asimilador)



Figura 5.2: Evidencia de la deducción del modelo entrenado, dando da como resultado grupo de imágenes de estilo Detallista. (Elaboración propia)

De igual forma en la Figura 5.3 se observa la evidencia de la deducción para el grupo de estilos SIMPLISTA (Convergente y Acomodador), donde se puede observar claramente que las imágenes tienen muy pocas características en sus diseños.



Figura 5.3: Evidencia de la deducción del modelo entrenado, dando da como resultado grupo de imágenes de estilo Simplista. (Elaboración propia)

5.3 RESULTADOS

El modelo propuesto de CNN, fue implementado en la plataforma Pytorch de un entorno de máquinas virtuales basado en Jupyter Notebooks de Google Colaboratory "Colab", que será la plataforma para la creación del prototipo de modelos de aprendizaje automático que esta implementado con potentes opciones de hardware como GPU y TPU. Proporciona un entorno portátil Jupyter sin servidor para el desarrollo interactivo, también proporciona tiempos de ejecución de Python 2 y 3 preconfigurados con las bibliotecas esenciales de aprendizaje automático e inteligencia artificial, como TensorFlow, Matplotlib y Keras. (Bisong, 2019). Arquitectura implementada en Colab: <https://colab.research.google.com/drive/1cMjYXSSjaV1vsxh48WY0ukhXtlyLp-U?usp=sharing>.

En la fase de entrenamiento se utilizó el aumento de datos para ampliar el conjunto de entrenamiento y definió las transformaciones para los datos de entrenamiento y los datos de prueba, se realizaron las siguientes transformaciones:

- RandomRotation(30).
- Resize(256),
- CenterCrop(224),
- ColorJitter(hue=.05, saturation=.05),
- RandomResizedCrop(224),
- RandomHorizontalFlip()
- ToTensor()
- Normalize ([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]))

Luego se cargo en el modelo “DenseNet”, ver Tabla 5.2 o Tabla 5.2 que están compuestos por dos partes principales, las características y el clasificador. La parte de características es una pila de capas convolucionales que funciona como un detector de características que se puede alimentar a un clasificador; la parte del clasificador es una sola capa completamente conectada: Linear (in_features = 1024, out_features = 1000), para reducir el sobre ajuste del modelo, se realizaron las siguientes variaciones a la imagen: variación aleatoria de forma horizontal, recorte de escala de forma aleatoria y un corte temporal de forma aleatoria.

Tabla 5.2 Configuración de los modelos

| DenseNet-121 | | | | Architecture: |
|----------------------|------------------|--|--|---|
| Layers | Output Size | DenseNet-121($k = 32$) | | Model.classifier = nn.Sequential(nn.Linear(1024, 512), nn.ReLU(), nn.Dropout(0.2), nn.Linear(512, 256), nn.ReLU(), nn.Dropout(0.1), nn.Linear(256, 2), nn.LogSoftmax(dim=1)) Optimizer = Adam Learning rate: lr=0.001 Batch size: 32 Target size: 150x150.... |
| Convolution | 112×112 | 7×7 conv, stride 2 | | |
| Pooling | 56×56 | 3×3 max pool, stride 2 | | |
| Dense Block (1) | 56×56 | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | | |
| Transition Layer (1) | 56×56 | 1×1 conv | | |
| | 28×28 | 2×2 average pool, stride 2 | | |
| Dense Block (2) | 28×28 | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | | |
| Transition Layer (2) | 28×28 | 1×1 conv | | |
| | 14×14 | 2×2 average pool, stride 2 | | |
| Dense Block (3) | 14×14 | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$ | | |
| Transition Layer (3) | 14×14 | 1×1 conv | | |
| | 7×7 | 2×2 average pool, stride 2 | | |
| Dense Block (4) | 7×7 | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$ | | |
| Classification Layer | 1×1 | 7×7 global average pool | | |
| | | 1000D fully-connected, softmax | | |

Tabla 5.3 Configuración de los modelos

| DenseNet-169 | | | Architecture: |
|----------------------|--|--|--|
| Layers | Output Size | DenseNet-169 | Model = densenet169 Model.classifier = nn.Sequential(nn.Linear(1664, 512), nn.ReLU(), nn.Dropout(0.2), nn.Linear(512, 256), nn.ReLU(), nn.Dropout(0.1), nn.Linear(256, 2), nn.LogSoftmax(dim=1)) Optimizer = Adam Learning rate: lr=0.003 Batch size: 32 Target size: 150x150 |
| Convolution | 112 × 112 | 7 × 7 conv, stride 2 | |
| Pooling | 56 × 56 | 3 × 3 max pool, stride 2 | |
| Dense Block (1) | 56 × 56 | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$ | |
| Transition Layer (1) | $\begin{matrix} 56 \times 56 \\ 28 \times 28 \end{matrix}$ | $\begin{matrix} 1 \times 1 \text{ conv} \\ 2 \times 2 \text{ average pool, stride 2} \end{matrix}$ | |
| Dense Block (2) | 28 × 28 | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$ | |
| Transition Layer (2) | $\begin{matrix} 28 \times 28 \\ 14 \times 14 \end{matrix}$ | $\begin{matrix} 1 \times 1 \text{ conv} \\ 2 \times 2 \text{ average pool, stride 2} \end{matrix}$ | |
| Dense Block (3) | 14 × 14 | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | |
| Transition Layer (3) | $\begin{matrix} 14 \times 14 \\ 7 \times 7 \end{matrix}$ | $\begin{matrix} 1 \times 1 \text{ conv} \\ 2 \times 2 \text{ average pool, stride 2} \end{matrix}$ | |
| Dense Block (4) | 7 × 7 | $\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$ | |
| Classification Layer | 1 × 1 | $\begin{matrix} 7 \times 7 \text{ global average pool} \\ 1000D \text{ fully-connected, softmax} \end{matrix}$ | |

La tasa de aprendizaje y el tamaño del lote son 10^{-3} y 16 para los diferentes conjuntos de datos. Se utilizó el gradiente descendiente estocástico (SGD) en mini lotes con un impulso de 0,5 y un descenso de peso de 10^{-3} para la optimización del modelo. La función de entropía cruzada definida en 5.1 se utiliza para el cálculo de la pérdida.

$$\text{loss}(x, \text{label}) = -\log \left(\frac{\exp(x[\text{label}])}{\sum_j \exp(x[j])} \right)$$

Para el test del modelo las imágenes se recortan espacialmente alrededor de los píxeles centrales y también en el espacio temporal para usar fotogramas adyacentes al centro. Además, para los experimentos, se usó una validación cruzada de diez partes, para así evaluar el rendimiento de reconocimiento del modelo propuesto.

Para demostrar la efectividad del sistema y considerando que no existe antecedente de la propuesta, es que se comparó con los modelos de *red ConvNet*, ver Tabla 5.4 y Tabla 5.5 teniendo en cuenta la característica de ingreso de las imágenes de 256x256x3, 256x256 píxeles por imagen y considerando tres colores (R,G y B), por que las imágenes de la base de datos son a colores y a blanco y negro realizados artesanalmente en hojas A4.

Al ejecutar la red, en la primera convolución extrae 16 filtros que son 3x3, seguida de la capa Max-Pooling con ventanas de 2x2; así sucesivamente para las siguientes convoluciones extraerá 32, 64 y 128 filtros. También se considera características de acoplar (más plano) a un tensor de 1-dim para que agregue capas completamente conectadas con activación ReLU y 512 unidades ocultas y cree una capa de salida con un solo nodo y activación sigmoidea.

Para ello todas las imágenes fueron reescalados por 1./255, las de entrenamiento de flujo en lotes de 20, usando el DataGenerator para producir los lotes de datos, como se muestra en las siguientes características:

Tabla 5.4 Configuración de los modelos

| ConvNet_01 | Architecture: |
|--|---|
| <pre> Model: "model_9" ----- Layer (type) Output Shape Param # ----- input_23 (InputLayer) [(None, 150, 150, 3)] 0 conv2d_96 (Conv2D) (None, 148, 148, 16) 448 max_pooling2d_81 (MaxPooling (None, 74, 74, 16) 0 conv2d_97 (Conv2D) (None, 72, 72, 32) 4640 max_pooling2d_82 (MaxPooling (None, 36, 36, 32) 0 conv2d_98 (Conv2D) (None, 34, 34, 64) 18496 max_pooling2d_83 (MaxPooling (None, 17, 17, 64) 0 conv2d_99 (Conv2D) (None, 15, 15, 64) 36928 max_pooling2d_84 (MaxPooling (None, 7, 7, 64) 0 conv2d_100 (Conv2D) (None, 5, 5, 128) 73856 max_pooling2d_85 (MaxPooling (None, 2, 2, 128) 0 flatten_8 (Flatten) (None, 512) 0 dense_16 (Dense) (None, 512) 262656 flatten_9 (Flatten) (None, 512) 0 dense_18 (Dense) (None, 512) 262656 dense_19 (Dense) (None, 1) 513 ----- Total params: 660,193 Trainable params: 660,193 Non-trainable params: 0 </pre> | <p>-16 Convolutional filters 3x3, MaxPooling, relu function</p> <p>-32 Convolutional filters 3x3, MaxPooling, relu function</p> <p>-64 Convolutional filters 3x3, MaxPooling, relu function</p> <p>-128 Convolutional filters 3x3, MaxPooling, relu function</p> <ul style="list-style-type: none"> ● Optimization: RMSprop ● Learning rate: lr=0.001 ● Batch size: 20 ● target size: 150x150 |

Tabla 5.5 Configuración de los modelos

| ConvNet_02 | Architecture: |
|------------|---------------|
| | |

| Layer (type) | Output Shape | Param # |
|-------------------------------|-----------------------|---------|
| input_1 (InputLayer) | [(None, 256, 256, 3)] | 0 |
| conv2d_2 (Conv2D) | (None, 254, 254, 32) | 896 |
| max_pooling2d_2 (MaxPooling2) | (None, 127, 127, 32) | 0 |
| conv2d_3 (Conv2D) | (None, 125, 125, 64) | 18496 |
| max_pooling2d_3 (MaxPooling2) | (None, 62, 62, 64) | 0 |
| conv2d_4 (Conv2D) | (None, 60, 60, 64) | 36928 |
| max_pooling2d_4 (MaxPooling2) | (None, 30, 30, 64) | 0 |
| conv2d_5 (Conv2D) | (None, 28, 28, 128) | 73856 |
| max_pooling2d_5 (MaxPooling2) | (None, 14, 14, 128) | 0 |
| conv2d_6 (Conv2D) | (None, 12, 12, 128) | 147584 |
| max_pooling2d_6 (MaxPooling2) | (None, 6, 6, 128) | 0 |
| flatten (Flatten) | (None, 4608) | 0 |
| dense (Dense) | (None, 512) | 2359808 |
| flatten_1 (Flatten) | (None, 512) | 0 |
| dense_2 (Dense) | (None, 512) | 262656 |
| dense_3 (Dense) | (None, 1) | 512 |
| Total params: 2,900,737 | | |
| Trainable params: 2,900,737 | | |
| Non-trainable params: 0 | | |

- 32 Convolutional filters 3x3, MaxPooling, relu function

- 64 Convolutional filters 3x3, MaxPooling, relu function

- 128 Convolutional filters 3x3, MaxPooling, relu function

- Optimization: RMSprop
- Learning rate: lr=0.001
- Batch size: 20
- target size: 256x256

Según los resultados de la tabla 5.6, proporciona la precisión de clasificación de los métodos más avanzados en cuatro conjuntos de datos, 3 de ellos estándar y el recolectado por el proyecto. Podemos ver que hay técnicas tradicionales de procesamiento de imágenes y modelos con aprendizaje profundo. De esta tabla, se puede concluir que los métodos de aprendizaje profundo generalmente funcionan mejor que los métodos tradicionales. Y el modelo propuesto logra el mejor

Tabla 5.6 Resultados del proceso de entrenamiento de las redes

| Red | Accuracy |
|--------------|----------|
| ConveNet-2 | 72.46% |
| DenseNet 121 | 87.77% |

Luego del proceso de entrenamiento de las redes; DenseNet obtuvo 15.31% de precisión a comparación de la ConveNet. Es importante señalar que no existe ningún tipo de preprocesamiento de las imágenes, eso dice de la simplicidad de la propuesta y su potencial uso para aplicaciones reales en el sector educación.

5.4 EVALUACIÓN DE LA EFICIENCIA

Para el entrenamiento de *DenseNet*, se utilizó el conjunto de datos, que se separó en 3 subconjuntos, 60% para entrenamiento, 20% para validación y 20% para prueba. Se repitió cinco veces para que la selección de los elementos no influyera en el resultado, ya que los conjuntos de datos fueron elegidos al azar para que los resultados a obtener sean confiables.

DenseNet es una red con las siguientes arquitecturas que se pueden ver a continuación tabla 5.2 y tabla 5.3, podemos ver una convolución de una máscara de 7x7 con 2 pasos, agrupación máxima de 3x3, podemos apreciar las arquitecturas 121, 169, 201 y 264, para el caso de la presente investigación se utilizó en este trabajo la configuración de DenseNet-121 ver tabla 5.2, la cual es de [6, 12, 24, 16] bloques densos; la misma presentó resultados aceptables para la solución del problema, además, se utilizaron estos parámetros de red:

- Algoritmo de optimización: Adam.
- Tasa de aprendizaje: lr 0.03.
- Tamaño de lote = 32

5.5 DECRECIMIENTO DE LOS MODELOS

Luego de entrenar los modelos, presentaremos el decrecimiento de los modelos de las gráficas de Loss (Pérdida) y el crecimiento del modelo Accuracy (precisión).

En el modelo presentado en la tabla 5.7 y tabla 5.8, se muestra como las pérdidas logran gradualmente decrecer dentro del rango de 0 a 50 épocas aproximadamente de entrenamiento el Train Loos y el Validation Loos y de la misma forma en otros modelos de Deep Learning, realizada para la base de datos seleccionada.

De igual forma se puede apreciar que la base de datos seleccionada en la tabla 5.8 tiene la mayor pérdida y decrece rápidamente en Train Loos; mientras el Validation Loos es más divergente. La misma se puede atribuir a la menor cantidad de imágenes en la base de datos que son menos de mil y las imágenes no han sido entrenadas previamente.

También tenemos en la tabla 5.8 y tabla 5.9, la Accuracy (precisión) donde se puede apreciar el rápido crecimiento en el Train Accuracy y la tendencia a permanecer constante del Validation Accuracy de la base de datos seleccionada; por otro lado en la tabla 5.8 y

la Figura 5.04, el Validation Accuracy tiende a permanecer constante en un rango de precisión de 0.8 a 0.9, aunque hay pequeñas variaciones en algunas épocas

Tabla 5.7 Decrecimiento de ConvNet

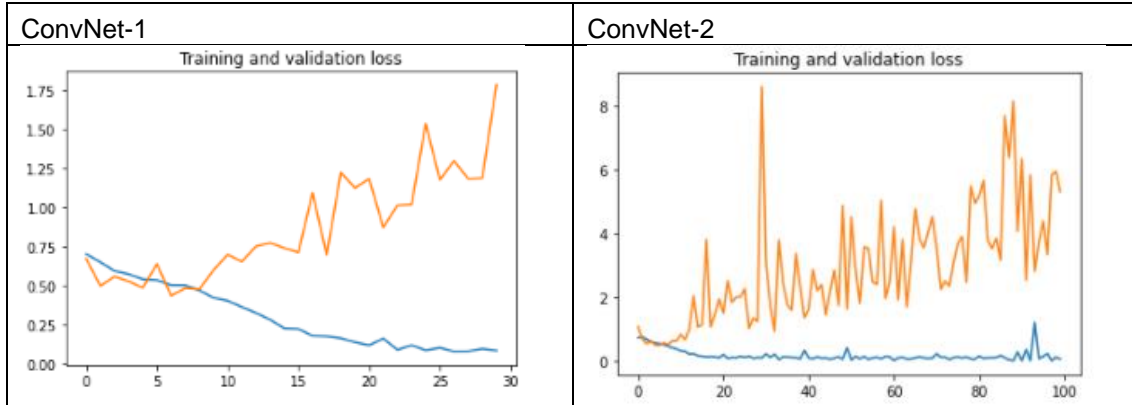
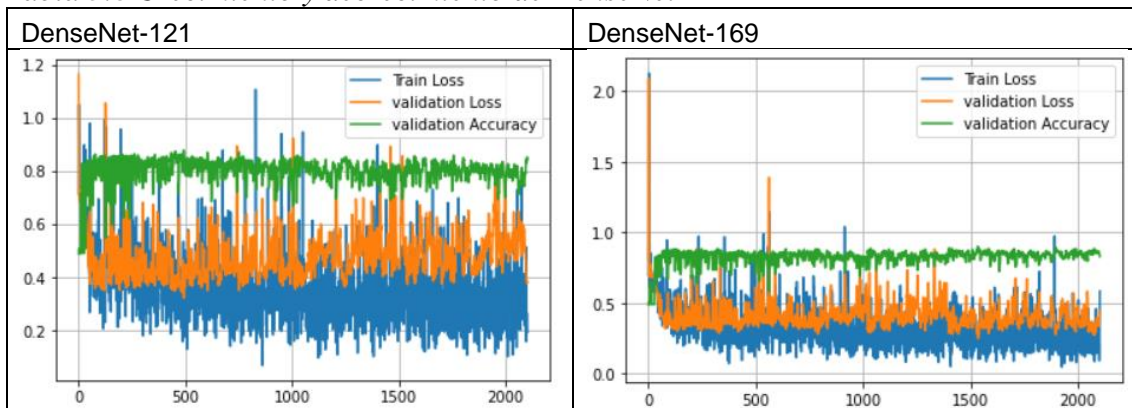


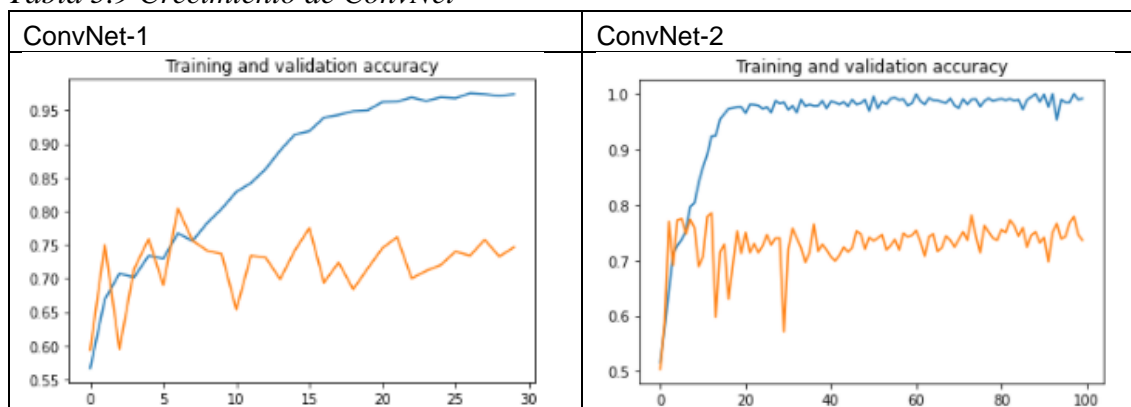
Tabla 5.8 Crecimiento y decrecimiento de DenseNet



5.6 ANÁLISIS DE LOS RESULTADOS

El creciente avance de la tecnología esta permitiendo mejorar las aplicaciones reales como el reconocimiento de diversas imágenes, los resultados muestran que el modelo propuesto es superior y único para el tipo de aplicación; también a otros del estado de arte y que la complejidad en las base de datos son determinantes al momento de clasificar los estilos de aprendizaje, teniendo en cuenta que las imágenes se realizaron en una hoja bond A4, luego se digitalizaron con un escáner y se presenta a la red capacitada (entrenada) para que sea la red quien decida qué estilo de aprendizaje es, y así dar la respuesta de la propuesta, Fig. 4.12., tal como se muestra al comparar el resultado de iteraciones de la red ConvNet que se puede ver en la tabla 5.9, para 30 épocas a más iteraciones y en la propuesta del modelo de la Figura 5.04.

Tabla 5.9 Crecimiento de ConvNet



Cabe precisar que DenseNet tiene varias arquitecturas donde la complejidad es la que cambia al incrementar mayor cantidad de capas densas principalmente, la misma mejora la capacidad de reconocimiento, tal como se muestra en el resultado de iteraciones de la propuesta del modelo de *DenseNet* en la Fig. 5.04, para 230 épocas de iteraciones.

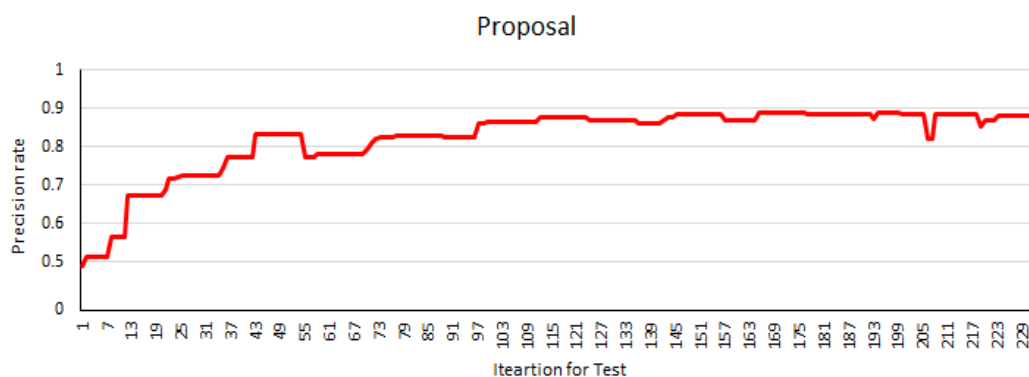


Figura 5.04: Resultado de precisión de DenseNet de la evolución del equipo de prueba desde el proceso de entrenamiento. (Elaboración propia)

La implementación del modelo CNN profundo y avanzado para la detección de los estilos de aprendizaje, mediante la aplicación de DenseNet, a partir de bocetos o dibujos a mano alzada de escena o paisaje libre, de estudiantes de *high school*; se podrá implementar en los sistemas recomendadores de objetos de aprendizaje, que han tomado gran importancia en los procesos de enseñanza aprendizaje; siendo una herramienta que aportan a la actividad docente y el aprendizaje de los estudiantes; porque este sistema es capaz de predecir estilos de aprendizaje, alcanzando una precisión del 87,77% que es buena para orientar a los estudiantes de *high school* en el proceso de mejora de su aprendizaje.

CONCLUSIONES Y RECOMENDACIONES

Los inconvenientes que se presentan para la validación de los sistemas recomendadores es la falta de información en muchos de los metadatos que describan una caracterización adecuada. Estos metadatos son necesarios para hacer una selección de los objetos de aprendizaje no sólo por su temática sino por las características y preferencias del usuario según los estilos de aprendizaje del mismo.

CONCLUSIONES

1. Es por ello, se propone un modelo de arquitectura Deep Learning que permite la detección de los estilos de aprendizaje en imágenes realizadas a mano alzada con la aplicación de la red convolucional densamente conectada DenseNet, con la finalidad de adecuarse a las reglas que se utilizan en los sistemas recomendadores, como la elección y selección de los recursos por parte de usuarios enmarcados en el perfil adecuado.
2. A través del inventario de estilos de aprendizaje de Kolb adaptado en una Ficha “Test de Estilos de Aprendizaje”, se llegó a predecir los estilos de aprendizaje asimilador (EA), divergente (ED), convergente (EC) y acomodador (EA) de los estudiantes de High School a partir de imágenes a mano alzada realizadas artesanalmente en hoja A4, la misma se pueda implementar en los modelos que se han llevado a la práctica y que se han utilizado para crear OA, con el propósito de hacerlos reutilizables y generativos.
3. Se propone el desarrollo de un sistema automático de detección de estilos de aprendizaje en imágenes realizadas artesanalmente a través de la aplicación de técnicas Deep Learning basado en redes Neuronales Convolucionales entrenadas con una base de datos de 500 imágenes agrupadas en dos grupos “RETAILER” (asimilador y divergente) y “SIMPLISTIC” (convergente y acomodador), según el estado de arte, se precisa que las CNN realizan automáticamente el proceso de extracción de características necesarias para obtener de las imágenes patrones relevantes, en este proyecto se evidencio algunas dificultades para que la red pueda

identificar los patrones y aprender; es por ello que se ensayó varias redes como ConvNet y DenseNet, esta última tuvo mejor desempeño en la identificación de los estilos de aprendizaje.

4. En modelo propuesto de *Deep Learning* puede aprender de manera eficiente las características y patrones de las imágenes, mediante la implementación de la arquitectura DenseNet a partir del dibujo de paisajes, alcanzando un 87,77% de precisión, es importante señalar que los dibujos son paisajes libres y al tener datos limitados lo que dificultaba la tarea de reconocimiento; es por ello se considero en la arquitectura las técnicas de Data Augmentation. Logrando la propuesta demostrar, la viabilidad de reconocer los estilos de aprendizaje con buena precisión, que se podrá implementar en el sistema recomendador de objetos de aprendizaje, que mejore el desempeño de los estudiantes de *high school*.
5. Esta metodología Deep Learning DenseNet se valida por ser capaz de predecir estilos de aprendizaje, alcanzando una precisión del 87,77% a comparación de la red CNN ConvNet, esto es un gran avance; pudiendo ser aplicado como una propuesta universal para implementar en los sistemas recomendadores de objetos de aprendizaje para orientar a los estudiantes de *high school* en el proceso de mejora de su aprendizaje.

RECOMENDACIONES

Creemos que el modelo sería mucho más eficaz si en la base de datos se considera más de 10 mil imágenes a mano alzada realizadas artesanalmente en hojas A4, los resultados aumentarían en porcentaje adecuado, de igual forma para este problema se requiere la alta precisión de CNN de Deep Learning; como hardware de alto rendimiento para tener el tiempo de respuesta de reconocimiento de estilos de aprendizajes, por lo tanto, las GPU son pertinentes para tal tarea. Teniendo presente las consideraciones señaladas se puede implementar en los sistemas recomendadores de objetos de aprendizaje para orientar a los estudiantes de *high school* en el proceso de mejora de su aprendizaje.

Bibliografía

- [Adan, Y. \(05 de 08 de 2018\). *What is the triplet loss in the context of Machine Learning*. Obtenido de https://www.quora.com/What-is-the-triplet-loss-in-the-context-of-Machine-Learning](https://www.quora.com/What-is-the-triplet-loss-in-the-context-of-Machine-Learning)
- [Alvarado, I. \(05 de 2018\). *Redes Neuronales*. Obtenido de https://ml4a.github.io/ml4a/es/neural_networks/](https://ml4a.github.io/ml4a/es/neural_networks/)
- [Álvarez Triviño, M. C. \(2017\). Deep learning aplicado a imágenes de retina como herramienta de tamizaje en un proceso de diagnóstico médico. *Bachelor's thesis, Universidad Autónoma de Occidente*.](#)
- [Antona Cortés, C. \(2017\). Herramientas modernas en redes neuronales: la librería Keras. *Bachelor's thesis, 9-10*.](#)
- [AprenderPython.net. \(2018\). *Aprender Python*. Obtenido de Descenso de Gradientes Estocástico “SGD”: https://www.aprenderpython.net/descenso-gradientes-estocastico-sgd/](https://www.aprenderpython.net/descenso-gradientes-estocastico-sgd/)
- [Artola Moreno, Á. \(2019\). Clasificación de imágenes usando redes neuronales convolucionales en Python.](#)
- [Ávila, R. H. \(2018\). Deep learning. Una revisión. *ResearchGate, 6*. doi:10.13140/RG.2.2.26893.84961](https://doi.org/10.13140/RG.2.2.26893.84961)
- [Bautista, H. M. \(2015\). Clasificación de recursos educativos a través de un sistema basado en el estilo de aprendizaje y las competencias de los alumnos de educación media superior. *Anais temporários do LACLO, 10\(3\), 219*.](#)
- [Betancur, D. M. \(2009\). Modelo para la recomendación y recuperación de objetos de aprendizaje en entornos virtuales de enseñanza/aprendizaje. *Revista Avances en Sistemas e Informática*.](#)
- [Bisong, E. \(2019\). Google Colaboratory. *En Creación de modelos de aprendizaje automático y aprendizaje profundo en Google Cloud Platform*, págs. 59-64.](#)
- [Bottou, L. C. \(2018\). Optimization methods for large-scale machine learning. *SIAM Review, 21-22*.](#)
- [Braga, J. &. \(2016\). Objetos de Aprendizaje. . *Objetos de Aprendizaje, 107*.](#)
- [Brownlee, J. \(20 de 12 de 2017\). *A Gentle Introduction to Transfer Learning for Deep Learning*. Obtenido de https://machinelearningmastery.com/transfer-learning-for-deep-learning/](https://machinelearningmastery.com/transfer-learning-for-deep-learning/)

- Burrueco, D. (2019). *INTERACTIVE CHAOS*. Obtenido de <https://www.interactivechaos.com/manual/tutorial-de-machine-learning/mini-batch-gradient-descent>
- Cadena Díaz, X. V. (2018). *Planteamiento de un Sistema Recomendador Basado en Contenido y Colaborativo para la Elección de Libros*. Bogota: UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS.
- Calvo, D. (20 de 07 de 2017). *Red Neuronal Convolucional CNN*. Obtenido de <http://www.diegocalvo.es/red-neuronal-convolucional/>
- Calvo, D. (07 de 12 de 2018). Obtenido de Función de activación – Redes neuronales: <http://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/>
- Canales Velis, E. R. (2016). *Sistema Informático de apoyo a los docentes para gestionar, impartir y optimizar la enseñanza de forma interactiva a los estudiantes de tercer ciclo del Centro Escolar San Ramón del municipio de Mejicanos del departamento de San Salvador*. San Salvador: UNIVERSIDAD DE EL SALVADOR FACULTAD DE INGENIERIA Y ARQUITECTURA ESCUELA DE INGENIERIA EN SISTEMAS INFORMÁTICOS.
- Canales, E. R., Gonzáles, L. A., Medina, A. F., & Palacios, M. A. (2016). *Sistema Informático de apoyo a los docentes para gestionar, impartir y optimizar la enseñanza de forma interactiva a los estudiantes de tercer ciclo del C.E.S.R. San Salvador, honduras: Universidad de el Salvador, facultad de Ingeniería y Arquitectura, escuela de Ingeniería en Sistemas Informáticos.*
- Carneiro, T. D. (2018). Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677-61685.
- Castañeda, P. N. (2012). Los estilos de aprendizaje de los estudiantes de la Universidad de La Salle según el modelo de Kolb y sus implicaciones para la didáctica universitaria. . *Actualidades Pedagógicas*, 123-147.
- Cazau, P. (2005). Estilos de aprendizaje: Generalidades. 5.
- Contreras, S. &. (2016). Aplicación de deep learning en robótica móvil para exploración y reconocimiento de objetos basados en imágenes. *In Computing Conference (CCC)*, IEEE 11th Colombian (pp. 1-8). IEEE.
- Cué, J. L. (2017). Sistema administrador de Objetos de Aprendizaje que contienen Estilos de Aprendizaje (SIGOAEA). *Journal of Learning Styles*, 10(17).

- Curry, B. (26 de 07 de 2018). *An Introduction to Transfer Learning in Machine Learning*. Obtenido de <https://medium.com/kansas-city-machine-learning-artificial-intelligen/an-introduction-to-transfer-learning-in-machine-learning-7efd104b6026>
- de Mola Garay, J. E. (2011). Estilos y estrategias de aprendizaje en el rendimiento académico de los estudiantes de la universidad peruana “Los Andes” de Huancayo–Perú. *Journal of Learning Styles*, 4(8).
- Deco, C. B. (2010). Sistema recomendador de recursos educativos para la enseñanza de las ciencias.
- Deco, C. C. (2014). Herramientas para potenciar el uso de Repositorios de Objetos de Aprendizaje. *In XVI Workshop de Investigadores en Ciencias de la Computación.*, 971 de 1158.
- Díaz-Barriga Arceo, F. (2010). Los profesores ante las innovaciones curriculares. *Revista Iberoamericana de Educación Superior*. Vol. I, N° 1. México. (Pp. 37-57).
- Durán Suárez, J. (2017). Redes neuronales convolucionales en R: Reconocimiento de caracteres escritos a mano.
- Durán, J. (4 de setiembre de 2019). *Todo lo que Necesitas Saber sobre el Descenso del Gradiente Aplicado a Redes Neuronales*. Obtenido de <https://medium.com/metadatos/todo-lo-que-necesitas-saber-sobre-el-descenso-del-gradiente-aplicado-a-redes-neuronales-19bdbb706a78>
- Frossard, D. (17 de 06 de 2016). *VGG en TensorFlow*. Obtenido de <https://www.cs.toronto.edu/~frossard/post/vgg16/>
- gage, J. (06 de 08 de 2018). *Introduction to Dataset Augmentation and Expansion*. Obtenido de <https://blog.algorithmia.com/introduction-to-dataset-augmentation-and-expansion/>
- Gandhi, R. (16 de 05 de 2018). *Siamese Network & Triplet Loss*. Obtenido de <https://towardsdatascience.com/siamese-network-triplet-loss-b4ca82c1aec8>
- González Clavero, M. V. (2011). Estilos de aprendizaje: su influencia para aprender a aprender. *Journal of Learning Styles*, 4(7).
- Google. (21 de junio de 2018). *Colaboratory: Frequently Asked Questions*. Obtenido de <https://research.google.com/colaboratory/faq.html>

- He, K. Z. (2016). Identity mappings in deep residual networks. *In European conference on computer vision* , 1-4.
- Hernández, L. (22 de 10 de 2018). *Metriplika S.L.* Obtenido de data augmentation: <http://www.doctormetrics.com/2018/10/22/datos-aumentados/#.XBj-MWhKi1v>
- Huang, G. L. (2017). Densely connected convolutional networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4700-4708).
- Larranaga, P. I. (1997). redes neuronales. *Redes Neuronales*, 12-17.
- LeCun, Y. B. (2015). Deep Learning. *nature*, 512(7553), 436.
- León, O. A. (2016). Estilos de aprendizaje y objetos virtuales para enseñanza. *Journal of Learning Styles*, 113.
- León, O. A. (2016). Estilos de aprendizaje y objetos virtuales para enseñanza. *Revista de Estilos de Aprendizaje*, 99.
- Liu, A. (18 de 01 de 2018). *¿Qué es un minibatch en una red neuronal?* Obtenido de <https://www.quora.com/What-is-a-minibatch-in-a-neural-network>
- Loncomilla, P. (2016). Deep learning: Redes convolucionales. 19-27.
- Longoni, M. G. (2010). odelos de Redes Neuronales Perceptrón Multicapa y de Base Radial para la predicción del rendimiento académico de alumnos universitarios. *In XVI Congreso Argentino de Ciencias de la Computación.*, 5.
- López Guzmán, C. . (2014). *Los Repositorios de Objetos de Aprendizaje como soporte a un entorno e-learning* , 40.
- Maldonado M, J. (2015). Desarrollo de un marco de análisis para la selección de metodologías de diseño de objetos de aprendizaje (OA) basado en criterios de calidad para contextos educativos específicos. *Doctoral dissertation, Universidad Nacional de La Plata*, 21.
- Maldonado M., J. (2015). *Desarrollo de un marco de análisis para la selección de metodologías de diseño de objetos de aprendizaje (OA) basado en criterios de calidad para contextos educativos específicos*. La Plata, Buenos Aires, Argentina: Doctoral dissertation, Universidad Nacional de La Plata.
- Maldonado Mahauad, J. (2015). *Desarrollo de un marco de análisis para la selección de metodologías de diseño de objetos de aprendizaje (OA) basado en criterios de calidad para contextos educativos específicos*. La Plata, Buenos Aires, Argentina : Doctoral dissertation, Universidad Nacional de La Plata.

- Maldonado, J. &. (2014). Los Objetos de Aprendizaje: Un estado del arte en Iberoamérica. In VI Conferencia Iberoamericana sobre Tecnologías y Aprendizaje, 8.
- Maldonado, J. C. (2015). Metodologías y propuestas metodológicas para el diseño de objetos de aprendizaje: Un estado del arte en iberoamérica. . Anais temporários do LACLO , 10(1), 36.
- Martínez Llamas, J. (2018). Reconocimiento de imágenes mediante redes neuronales convolucionales. Madrid: Reconocimiento de imágenes mediante redes neuronales convolucionales.
- Massiris, M. D. (2018). Detección de equipos de protección personal mediante red neuronal convolucional YOLO. In XXXIX Jornadas de Automática. Área de Ingeniería de Sistemas y Automática, Universidad de Extremadura., pp. 1023.
- Matich, D. J. (2001). Redes Neuronales: Conceptos básicos y aplicaciones. Universidad Tecnológica Nacional, 13-14.
- Matlab Engineers. (2018). Entrenando a la red - Mini lotes. Obtenido de <https://matlab4engineers.com/lesson/training-network-mini-batches/?lang=en&v=3acf83834396>
- Mayaute, L. M. (1992). Adaptación del inventario de estilos de aprendizaje de Kolb. Revista de Psicología, 10(1), 125-142.
- Mendoza Sánchez, J. L. (2020). Herramienta para segmentación de líneas de texto en imágenes basada en Deep Learning.
- MINEDU. (2017). Currículo Nacional de la Educación Básica. Definiciones Claves que sustentan el Perfil del Estudiante(III), 29 al 35.
- MINEDU. (2017). Currículo Nacional de la Educación Básica. Definiciones Claves que sustentan el Perfil del Estudiante, Cap. III. pág. 29 al 35.
- Moindrot, O. (19 de 03 de 2018). Triplet Loss and Online Triplet Mining in TensorFlow. Obtenido de <https://omindrot.github.io/triplet-loss>
- Morales, E. M., Garcia, F., Campos, R. A., & Astorga, C. (2013). Desarrollo de competencias a través de objetos de aprendizaje. RED. Revista de Educación a Distancia. Número 36, 2.
- Morgado, E. M. (2010). Gestión del conocimiento en sistemas «e-learning», basado en objetos de aprendizaje, cualitativa y pedagógicamente definidos. Salamanca: Ediciones Universidad de Salamanca - Edición Victor.

- Navarro, B. G. (2015). *Implementación de técnicas de Deep learning*. Escuela Superior de Ingeniería y Tecnología. Laguna: Universidad de la Laguna.
- Ocampo, R. (2014). Red de Base Radial: Nueva Aproximación con Menos Parámetros. *Escuela de Graduados en Ingeniería y Arquitectura (EGIA)*, 21-23.
- Ocampo-Botello, F. &.-D.-C. (2014). Sistema de recomendación para el comercio electrónico aplicado a una tienda de libros. *Científica*, 18 (2), 55-62., 56-58.
- Ortega, E. R. (2019). Estilos de aprendizaje: estrategias de enseñanza en LUZ. *Telos: Revista de Estudios Interdisciplinarios en Ciencias Sociales*, 21(3), 710-730.
- ORTUÑO, R. A. (2016). Diseño de Objetos de Aprendizaje adaptados para cuatro estilos de aprender: un estudio de caso. *Revista de Educação Pública*, 25(59/2), 548-572.
- Perez, G. (02 de 10 de 2018). *¿Cómo funcionan las redes neuronales convolucionales?* Obtenido de <https://es.quora.com/C%C3%B3mo-funcionan-las-redes-neuronales-convolucionales>
- Ponti, M. A., & da Costa, G. B. (2017). Como funciona o Deep Learning. *Tópicos em Gerenciamento de Dados e Informacoes*, 66.
- Ponti, M. A., & Paranhos da Costa, G. B. (2017). Como funciona o Deep Learning. *Tópicos em Gerenciamento de Dados e Informacoes*, 65.
- Ponti, M. y. (2018). Como funciona o Aprendizaje profundo. *Tópicos em Gerenciamento de Dados e Informacoes*, 85.
- Qi, Q. H. (2019). Recuperación de imágenes personalizada basada en bocetos mediante red neuronal convolucional y aprendizaje de transferencia profunda. *Acceso IEEE*, 7, 16537-16549.
- Raj, B. (11 de 04 de 2018). *Data Augmentation / How to use Deep Learning when you have Limited Data*. Obtenido de <https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced>
- Restrepo Arteaga, G. J. (2015). Aplicación del aprendizaje profundo (deep learning) al procesamiento de señales digitales. (*Bachelor's thesis, Universidad Autónoma de Occidente*), 58 y 59.
- Rodríguez, P. A. (2015). Análisis de Características del Perfil de Usuario para un Sistema de Recomendación de Objetos de Aprendizaje. *Conferencias LACLO*, 5(1).

- [Romero, L. U. \(2010\). Estilos de aprendizaje basados en el modelo de Kolb en la educación virtual. *Apertura: Revista de innovación educativa*, 72-85.](#)
- [Rosebrock, A. \(20 de 03 de 2017\). *ImageNet: VGGNet, ResNet, Inception y Xception con Keras*. Obtenido de <https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>](#)
- [Salumaa, S. O. \(2018\). Convolutional Neural Networks for Cellular Segmentation. *Tesis de máster \(30 ECTS\)*, 17-18.](#)
- [Sánchez Alba, R. \(2017\). Implementación de un algoritmo de aprendizaje automático en Apache Spark. *Doctoral dissertation*, 21 y 22.](#)
- [Sánchez Bocanegra, C. L. \(2017\). *HealthRecSys: Sistema Recomendador para la salud*. Sevilla: HealthRecSys: Sistema Recomendador para la salud.](#)
- [Schilardi, A., León, O. A., Segura, S., & Polenta, C. \(2016\). Estilos de Aprendizaje y Objetos Virtuales para la Enseñanza. *Revista de estilos de Aprendizaje*, 99.](#)
- [Shankar, S. \(02 de 09 de 2017\). *Investigador en Ciencias de la Computación en Cambridge*. Obtenido de <https://www.quora.com/profile/Sukrit-Shankar>](#)
- [Sprock, A. S. \(2015\). Evaluación del Método para el Desarrollo de Objetos de Aprendizaje, basado en Estilos de Aprendizaje: MeLOTS. . *Revista Tecnológica-ESPOL*, 40 y 42.](#)
- [Szegedy, C. L. \(2015\). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* , 4-6.](#)
- [Szegedy, C. L. \(2015\). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* , 4-6.](#)
- [Szegedy, S. I. \(2015\). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In ICML*.](#)
- [Thomas, A. \(2018\). *AVENTURAS EN EL APRENDIZAJE AUTOMÁTICO*. Obtenido de <http://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-tensorflow/>](#)
- [Tsang, S. \(22 de 08 de 2018\). *Review: VGGNet—1st Runner-Up \(Image Classification\). Winner \(Localization\) in ILSVRC 2014*. Obtenido de <https://medium.com/coinmonks/paper-review-of-vggnet-1st-runner-up-of-ilsvrc-2014-image-classification-d02355543a11>](#)

- Vega Arias, J. M. (2019). Modelo de aprendizaje profundo/red neuronal convolucional (CNN) para clasificación de calidad de ácidos grasos por imágenes de semillas de *Helianthus annuus*. 39.
- Virdee, A. (06 de 06 de 2018). *Data Augmentation Experimentation*. Obtenido de <https://towardsdatascience.com/data-augmentation-experimentation-3e274504f04b>
- Vo, Q. K. (2017). Segmentación de líneas de texto utilizando una red totalmente convolucional en imágenes de documentos escritos a mano. *Procesamiento de imágenes IET*, 12 (3), 438-446.
- Wiley, D. A. (2000). *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. The instructional use of learning objects*. Bloomington, Indiana: Agency for Instructional Technology Association for Educational Communications & Technology.
- X. Glorot, A. B. (2011). Deep sparse rectifier neural networks. *In AISTATS*.
- Ye, Y. (2016). Aplicación del aprendizaje profundo al reconocimiento de bocetos de escenas y la recuperación de modelos 3D basados en bocetos.
- Yeung, F.-F. L. (02 de 05 de 2017). *CNN Architectures*. Obtenido de http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture9.pdf
- Yungan D., L. (2015). *Análisis del nivel de influencia de las TIC en los procesos educativos de los estudiantes del octavo al décimo año del colegio Galápagos*. Guayaquil - Ecuador: Universidad de Huayaquil, facultad de comunicación social, carrera de comunicación social.