

## Urkund Analysis Result

**Analysed Document:** PROPUESTA ANDRADE-CAÑAR.docx (D44239881)  
**Submitted:** 11/19/2018 9:26:00 PM  
**Submitted By:** jortizm2@unemi.edu.ec  
**Significance:** 1 %

Sources included in the report:

CAPITULO II.docx (D10607583)

Instances where selected sources appear:

2

## INTRODUCCIÓN

El aprendizaje automático es una rama de la inteligencia artificial que permite a los sistemas informáticos aprender directamente de ejemplos, datos y experiencias. Al permitir que las computadoras realicen tareas específicas de manera inteligente, los sistemas de aprendizaje automático pueden llevar a cabo procesos complejos al aprender de los datos, en lugar de seguir reglas pre programadas. Los últimos años han visto avances emocionantes en el aprendizaje automático, que han aumentado sus capacidades en un conjunto de aplicaciones. El aumento de la disponibilidad de datos ha permitido que los sistemas de aprendizaje automático se entrenen en una gran cantidad de ejemplos, mientras que el aumento de la capacidad de procesamiento de la computadora ha respaldado las capacidades analíticas de estos sistemas. Dentro del propio campo también se han producido avances algorítmicos, que han dado un mayor poder al aprendizaje automático. Como resultado de estos avances, los sistemas que solo hace unos años funcionaron en niveles notablemente inferiores a los humanos ahora pueden superar a los humanos en algunas tareas específicas. Muchas personas ahora interactúan con sistemas basados en el aprendizaje automático todos los días, por ejemplo, en sistemas de reconocimiento de imágenes, como los que se usan en las redes sociales; sistemas de reconocimiento de voz, utilizados por asistentes personales virtuales. A medida que el campo se desarrolla, el aprendizaje automático promete apoyar avances potencialmente transformadores en una variedad de áreas, y las oportunidades sociales y económicas. A medida que los sistemas de aprendizaje automático se vuelven más omnipresentes, o significativos en ciertos campos debido al aumento de datos que han proporcionado una gran cantidad de ejemplos, que los sistemas de aprendizaje automático pueden utilizar para mejorar su rendimiento. A su vez, el aprendizaje automático puede ayudar a abordar los beneficios sociales y económicos que se esperan de los llamados 'big data', y extraer información valiosa a través de análisis de datos avanzados. Apoyar el desarrollo de esta función para el aprendizaje automático requiere un entorno de datos adaptable, basado en estándares y marcos o comportamientos abiertos para garantizar la disponibilidad de datos en todos los sectores. El presente trabajo está conformado por cinco capítulos los cuales son: Capítulo 1: En el primer capítulo podremos observar el planteamiento del problema, así como su respectiva línea de investigación, objetivo general y específico y justificación del problema. Capítulo 2: El segundo capítulo está formado por los conceptos básicos en cuanto al tema como sus elementos, métodos de aprendizaje y software que se emplearan. Capítulo 3: En el tercer capítulo podremos apreciar la metodología descriptiva y documental que se empleó para el desarrollo del tema, esta incluye ejemplos, clasificación y reconocimiento de datos. Capítulo 4: En este capítulo se mostrara un ejemplo de reconocimiento de imágenes conjuntamente con el algoritmo y procedimientos que se deben seguir.

## CAPÍTULO 1

PROBLEMA DE INVESTIGACIÓN . 1.1 Planteamiento del problema Los avances en el aprendizaje automático y el uso de servicios de internet están impulsando el crecimiento de esta tecnología, algo que muchas personas no conocen acerca de la inteligencia artificial es el proceso que se dedica a crear lo que se define un conjunto de datos. Así es como se entrena el

modelo de aprendizaje: practica hacer predicciones a partir de la información en un conjunto de datos y utiliza esa experiencia en situaciones del mundo real. Parte de por qué el reconocimiento de imágenes es una forma de inteligencia artificial tan desarrollada y ampliamente utilizada es la forma en que se desarrollan los conjuntos de datos. La importancia y la necesidad del procesamiento digital de imágenes se deriva de dos áreas de aplicación principales: la primera es la mejora de la información gráfica para la interpretación humana y la segunda el procesamiento de los datos de una escena para una percepción autónoma de la máquina. El procesamiento digital de imágenes tiene una amplia gama de aplicaciones, como la detección remota, el almacenamiento de imágenes y datos para la transmisión en aplicaciones comerciales, imágenes médicas como procesamiento de rayos X, escaneo ultrasónico, micrografías electrónicas, imágenes de resonancia magnética, imágenes de resonancia magnética nuclear, etc., imágenes acústicas, ciencias forenses y automatización industrial. Además existen las aplicaciones de imágenes espaciales, estas incluyen el reconocimiento y el análisis de objetos contenidos en imágenes las cuales han sido obtenidas de misiones de sondas de espacio profundo. El mercado de reconocimiento de imágenes se divide en hardware, software y servicios. El segmento de hardware dominado por teléfonos inteligentes y escáneres puede jugar un papel muy importante en el crecimiento del mercado de reconocimiento de imágenes. Según datos de la revista Proméxico diariamente se generan 2.5 exabytes de datos o lo que es lo mismo, 2.5 billones de gigabytes en todo el mundo, este fenómeno se conoce como Big Data. El MIT la define como procesamiento y análisis de grandes volúmenes de información, con variedad de ubicaciones de datos, los cuales requieren de aplicar Como podemos observar actualmente el procesamiento de datos es mayor, esto es debido a que el coste de almacenamiento de los mismos se ha vuelto accesible y al mismo tiempo sus algoritmos más sofisticados CITATION Nev17 \p 15 \l 12298 (Nevela, 2017, pág. 15). Es aquí donde vemos plasmado el funcionamiento de esta disciplina denominada machine learning, la cual consiste en convertir una enorme cantidad de datos en información y conocimiento, como su definición más simple lo menciona, es una área de las ciencias computacionales donde la computadora obtiene la destreza de educarse sin la necesidad de ser explícitamente programadas, es decir que esta sea capaz por si sola de detectar un reconocimiento de patrones y vínculos que permiten clasificar un conjunto de datos para de este modo emplear algoritmos que pueden generar descubrimientos importantes sin la necesidad de que una persona tenga que escribir instrucciones o códigos de manera tradicional. Por lo tanto la herramienta de código abierto de MATLAB es un gran recurso para implementar el aprendizaje automático de última generación.

1.2 Línea de investigación Procesamiento y análisis de datos. 1.3 Objetivos 1.2.1 Objetivo general Estudiar todas las funciones necesarias para el procesamiento de imágenes usando técnicas de Machine Learning de Matlab. 1.2.2 Objetivos específicos 1. Identificar los elementos y modelos del aprendizaje automático y su funcionamiento junto con la inteligencia artificial. 2. Enlistar los ejemplos de aplicaciones comunes con Machine Learning en MATLAB. 3. Describir los procesos para el reconocimiento de imágenes usando técnicas de Machine Learning de Matlab

1.3 Justificación El aprendizaje automático ofrece un valor potencial a las compañías que intentan aprovechar Big Data y les ayuda a comprender mejor los cambios en el comportamiento, las preferencias o la satisfacción del cliente. Los líderes empresariales están comenzando a darse cuenta de que suceden muchas cosas dentro de sus organizaciones y sus industrias que no se pueden entender a través de una consulta

Por lo tanto es posible aprovechar los algoritmos y modelos para predecir los resultados en donde se requieren técnicas de aprendizaje automático para mejorar la precisión de los modelos dependiendo de la naturaleza del problema sea este de ámbito comercial, industrial, medicina entre otros que se esté abordando, existen diferentes enfoques basados en el tipo y volumen de los datos.

## CAPÍTULO 2

### MARCO TEÓRICO CONCEPTUAL

2.1 Conceptos de Machine Learning El aprendizaje automático es una tecnología que permite a las computadoras aprender directamente de ejemplos y experiencias en forma de datos. Los enfoques tradicionales de programación se basan en reglas codificadas, que establecen cómo resolver un problema, paso a paso. A diferencia de los sistemas de aprendizaje automático se configuran como una tarea y se les da una gran cantidad de datos para usar como ejemplos de cómo se puede lograr esta tarea o para detectar patrones. Luego, el sistema aprende la mejor manera de lograr la salida deseada. Puede pensarse como una IA: el aprendizaje automático admite sistemas inteligentes, que pueden aprender una función en particular, dado un conjunto específico de datos para aprender. El Machine Learning es un sub campo de las ciencias de la computación y una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a los computadores aprender, es decir, generalizar comportamientos y conocimientos a partir de una información suministrada en forma de ejemplos CITATION Lan15 \l 12298 (Lantz, 2015)

2.2 Elementos de máquina de aprendizaje Esta sección observaremos la nomenclatura clave para el campo del aprendizaje automático.

2.2.1 Datos Todos los métodos de aprendizaje son impulsados por datos. Se utilizan conjuntos de datos para entrenar el sistema. Estos conjuntos pueden ser recolectados por humanos y usados para entrenamiento. Los conjuntos pueden ser muy grandes. Los sistemas de control pueden recopilar datos de los sensores a medida que los sistemas operan y usarlos para identificar parámetros, o capacitar al sistema. Al recopilar datos del entrenamiento, se debe tener cuidado para garantizar que se comprenda la variación temporal del sistema. Si la estructura de un sistema cambia con el tiempo, puede ser necesario descartar los datos antiguos antes de entrenar el sistema.

2.2.2 Modelos Los modelos se utilizan a menudo en los sistemas de aprendizaje. Un modelo proporciona un marco matemático para la instrucción. Son de origen humano basados en observaciones y experiencias humanas. Por ejemplo, un modelo de automóvil, visto desde arriba, puede ser que tiene forma rectangular con dimensiones que se ajustan a un lugar de estacionamiento estándar. Los modelos generalmente se consideran basados en observaciones y experiencias y proporcionan un marco para el aprendizaje automático. Sin embargo, algunas formas de aprendizaje automático desarrollan sus propios modelos sin una estructura derivada del ser humano. El

concepto adaptación hace referencia al hecho de que la aplicación cambia (adapta) su comportamiento para cada usuario. La adaptación se puede basar en varios rasgos del usuario almacenados en el modelo de usuario, o en características del entorno como la localización del usuario, la hora, el dispositivo usado, etc CITATION Par08 \l 12298 (Paredes, 2008).

### 2.2.3 Entrenamiento

Un sistema que asigna una entrada a una salida necesita capacitación para hacer esto de una manera útil. Al igual que las personas deben estar capacitadas para realizar tareas, los sistemas de aprendizaje automático deben capacitarse. La capacitación se realiza al proporcionar al sistema una entrada y la salida correspondiente y modificar la estructura (modelos o datos) en la máquina de aprendizaje para que aprenda el mapeo. En cierto modo esto es como ajuste de curva o regresión. Si tenemos suficientes pares de entrenamiento, entonces el sistema debería ser capaz de producir salidas correctas cuando se introducen nuevas entradas. Por ejemplo, si le damos a un sistema de reconocimiento facial miles de imágenes de gatos y le decimos que son gatos, esperamos que cuando se le den nuevas imágenes de gatos, también las reconozca como gatos. Pueden surgir problemas cuando no le das suficientes series de entrenamiento o los datos de entrenamiento no son lo suficientemente diversos, es decir, no representan la gama completa de gatos en este ejemplo.

- **Aprendizaje supervisado** La computadora se presenta con entradas de ejemplo y sus salidas deseadas, dadas por un maestro, y el objetivo es aprender una regla general que asigne entradas a salidas CITATION Smi17 \p 12 \l 12298 (Smith, 2017, pág. 12). Cuando hablamos de aprendizaje supervisado estamos haciendo énfasis a "conjuntos de datos que se encuentran clasificados y etiquetados" los cuales se los conoce como "derivados humanos", es decir son basados en observaciones y experiencias humanas, por ello requieren supervisión de un experto el mismo que se encargara de agrupar y categorizar al conjunto de muestra para el entrenamiento, dicho esto cabe recalcar que no necesariamente una persona debe estar verificando constantemente los resultados obtenidos. Para este tipo de supervisión existen dos pasos: 1. El proceso de clasificación de las salidas del sistema: Este proceso es empleado cuando existe un conjunto dado de entradas, denominado también etiquetado. En pocas palabras es donde el experto dice explícitamente que resultados son correctos o a su vez los resultados que espera para cada conjunto de entrada existente. 2. El proceso de entrenamiento de los conjuntos: Este es un aspecto importante dado que en este transcurso el experto debe asegurarse de proporcionar los suficientes datos del mundo real a los conjuntos para que el sistema suministre los resultados correctos, dado esto la información es sometida a un conjunto de pruebas para validar los resultados, si estos no son los esperados los conjuntos son enviados nuevamente al entrenamiento y se repite el proceso.

0: CAPITULO II.docx

100%

Aprendizaje no supervisado Las redes con aprendizaje no supervisado (también conocido como auto supervisado)

0: CAPITULO II.docx

100%

no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta.

CITATION Mat01 \p 19 \l 12298 (Matich, 2001, pág. 19) El aprendizaje no supervisado no utiliza conjuntos de entrenamiento. A menudo se usa para descubrir patrones en datos para los que no hay una respuesta "correcta". Por ejemplo, si usó un aprendizaje no supervisado para entrenar un sistema de identificación de rostros, el sistema podría agrupar los datos en conjuntos, algunos de los cuales podrían ser rostros. Los algoritmos de agrupación en general son ejemplos de aprendizaje no supervisado. La ventaja del aprendizaje no supervisado es que puede aprender cosas sobre los datos que tal vez no sepa de antemano. Es una forma de encontrar estructuras ocultas en los datos. • Aprendizaje semi supervisado Con el enfoque semi supervisado, algunos de los datos están en forma de conjuntos de entrenamiento etiquetados y otros no lo están. De hecho, normalmente solo se etiqueta una pequeña cantidad de los datos de entrada, mientras que la mayoría no lo está, ya que el etiquetado puede ser un proceso intensivo que requiere un humano experto. El pequeño conjunto de datos etiquetados se aprovecha para interpretar los datos sin etiquetar. • Aprendizaje en línea El sistema se actualiza continuamente con nuevos datos. Esto se denomina "en línea" porque muchos de los sistemas de aprendizaje utilizan datos recopilados en línea. También podría denominarse "aprendizaje recursivo". Puede ser beneficioso para "procesar" periódicamente los datos de procesos utilizados hasta un momento determinado y luego volver al modo de aprendizaje en línea. Los sistemas de filtrado de spam de la introducción utilizan el aprendizaje en línea.

2.2.4 La máquina de aprendizaje La máquina absorbe información del entorno y se adapta. Tenga en cuenta que las entradas se pueden separar en aquellas que producen una respuesta inmediata y aquellas que conducen al aprendizaje. En algunos casos están completamente separados. Por ejemplo, en una aeronave, una medida de altitud no se suele usar directamente para el control. En su lugar, se utiliza para ayudar a seleccionar parámetros para las leyes de control reales. Los datos requeridos para el aprendizaje y la operación regular pueden ser los mismos, pero en algunos casos se necesitarán mediciones o datos separados para que el aprendizaje tenga lugar. Las mediciones no necesariamente significan datos recopilados por un sensor como un radar o una cámara. Pueden ser datos recopilados por sondeos, precios del mercado de valores, datos en libros contables o datos recopilados por cualquier otro medio. El aprendizaje automático es entonces el proceso mediante el cual las mediciones se transforman en parámetros para la operación futura (Ver Figura 1)

IMAGEN Tenga en cuenta que la máquina produce salida en forma de acciones. Se puede pasar una copia de las acciones al sistema de aprendizaje para que pueda separar los efectos de las acciones de la máquina de los del entorno. Esto es similar a un sistema de control de avance, que puede resultar en un mejor rendimiento. 2.3 Métodos de aprendizaje

2.3.1 Regresión Este método se utiliza para predecir el valor de un atributo continuo. Consiste en encontrar la mejor ecuación que atraviese de forma óptima un conjunto de puntos ( $n$ -dimensiones). Se utiliza cuando la precisión no es crítica y el número de variables es pequeño.

Ej.: Predecir el precio de una vivienda, dado su tamaño (Ver Figura 2). IMAGEN La regresión es una forma de ajustar los datos a un modelo. Un modelo puede ser una curva en múltiples dimensiones. El proceso de regresión ajusta los datos a la curva, produciendo un modelo que puede usarse para predecir datos futuros. Algunos métodos, como la regresión lineal o los mínimos cuadrados, son paramétricos porque se conoce el número de parámetros que deben ajustarse. Un ejemplo de regresión lineal se muestra en la lista a continuación (ver Figura 3). Este modelo se creó comenzando con la línea  $y=x$  y agregando ruido a  $y$ . La línea fue creada usando un ajuste de mínimos cuadrados a través de la función `pinv Pseudoinverse` de MATLAB. IMAGEN

2.3.2 Redes Neuronales Un grupo muy importante de algoritmos para el aprendizaje automático supervisado y no supervisado son las redes neuronales. Estos se basan en gran parte del aprendizaje automático, y mientras que los modelos simples como la regresión lineal utilizados pueden usarse para hacer predicciones basadas en un pequeño número de características de datos, como en el ejemplo de Google con cerveza y vino, las redes neuronales son útiles cuando se trata de grandes conjuntos de Datos con muchas características CITATION Hea18 \l 12298 (Heath, 2018). Una red neuronal es una red diseñada para emular las neuronas en un cerebro humano. Cada "neurona" tiene un modelo matemático para determinar su salida a partir de su entrada; por ejemplo, si la salida es una función escalonada con un valor de 0 o 1, se puede decir que la neurona está "disparando" si el estímulo de entrada da como resultado una salida de 1. Las redes se forman entonces con múltiples capas de neuronas interconectadas. Las redes neuronales son una forma de reconocimiento de patrones. La red debe ser entrenada utilizando datos de muestra, pero no se requiere un modelo a priori. Las redes pueden entrenarse para estimar la salida de procesos no lineales y la red se convierte en el modelo. La Figura 4 muestra una red neuronal simple que fluye de izquierda a derecha, con dos nodos de entrada y un nodo de salida. Hay una capa "oculta" de neuronas en el medio. Cada nodo tiene un conjunto de pesos numéricos que se ajusta durante el entrenamiento. Una red neuronal "profunda" es una red neuronal con múltiples capas intermedias entre la entrada y la salida. Las redes neuronales son un área activa de investigación. IMAGEN 2.3.3 Árboles de decisión Un árbol de decisiones es un gráfico similar a un árbol que se usa para tomar decisiones. Tiene tres tipos de nodos: 1. Nodos de decisión 2. Nodos de azar 3. Nodos finales Se debe seguir la ruta desde el principio hasta el nodo final. Los árboles de decisión son fáciles de entender e interpretar. El proceso de decisión es completamente transparente, aunque los árboles de decisión muy grandes pueden ser difíciles de seguir visualmente. La dificultad es encontrar un árbol de decisiones óptimo para un conjunto de datos de entrenamiento. Dos tipos de árboles de decisión son los árboles de clasificación, que producen resultados categóricos, y los árboles de regresión, que producen resultados numéricos. En la Figura 5 se muestra un ejemplo de un árbol de clasificación. Esto ayuda a un empleado a decidir dónde ir para el almuerzo. Este árbol solo tiene nodos de decisión. Esto podría ser usado por la administración para predecir dónde podrían encontrar un empleado a la hora del almuerzo. La decisión es Hambriento, Ocupado y Tener una Tarjeta de Crédito. A partir de eso se pudo sintetizar el árbol. Sin embargo, si hubiera otros factores en la decisión de los empleados, por ejemplo, es el cumpleaños de alguien, lo que daría lugar a que el empleado fuera a un restaurante, entonces el árbol no sería exacto. IMAGEN 2.3.4 Sistemas Expertos Un sistema utiliza una base de conocimientos

para razonar y presentar al usuario un resultado y una explicación de cómo llegó a ese resultado. Los sistemas expertos también se conocen como sistemas basados en el conocimiento. El proceso de creación de un sistema experto se denomina "ingeniería del conocimiento". Esto involucra a un ingeniero del conocimiento, alguien que sabe cómo construir el sistema experto, entrevistando a expertos para obtener el conocimiento necesario para construir el sistema. Algunos sistemas pueden inducir reglas a partir de datos, acelerando el proceso de adquisición de datos. Una ventaja de los sistemas expertos sobre los expertos humanos es que el conocimiento de múltiples expertos puede incorporarse a la base de datos. Otra ventaja es que el sistema puede explicar el proceso en detalle para que el usuario sepa exactamente cómo se generó el resultado. Incluso un experto en un dominio puede olvidarse de verificar ciertas cosas. Un sistema experto siempre verificará metódicamente su base de datos completa. Tampoco se ve afectado por la fatiga o las emociones. La adquisición de conocimiento es un importante cuello de botella en la construcción de sistemas expertos. Otro problema es que el sistema no puede extrapolar más allá de lo programado en la base de datos. Se debe tener cuidado con el uso de un sistema experto, ya que generará respuestas definitivas para los problemas en los que existe incertidumbre. La facilidad de explicación es importante porque alguien con conocimiento de dominio puede juzgar los resultados de la explicación. En los casos en los que se debe considerar la incertidumbre, se recomienda un sistema experto probabilístico. Una red bayesiana puede ser utilizada como un sistema experto. Una red bayesiana también se conoce como una red de creencias. Es un modelo gráfico probabilístico que representa un conjunto de variables aleatorias y sus dependencias. En los casos más simples, una red Bayesiana puede ser construida por un experto. En casos más complejos, debe generarse a partir de datos del aprendizaje automático.

2.4 Software para Machine Learning MATLAB es un software matemático el cual permite realizar gráficas en base a los cálculos y análisis de datos. MATLAB tiene muchas herramientas específicamente destinadas para Machine Learning que serán alistadas a continuación: • Herramientas de Estadísticas y Aprendizaje Automático • Herramientas para red neuronal • Herramientas para sistema de visión artificial • Herramientas para sistemas de identificación

Herramientas de estadísticas y aprendizaje automático Facilitan métodos de análisis de datos para patrones a partir de enormes cantidades de datos. Las funciones de las herramientas se pueden dividir ampliamente en herramientas de clasificación, herramientas de regresión y herramientas de agrupación en clústeres CITATION The18 \l 12298 (The MathWorks,Inc, 2018). Herramientas de red neuronal Proveen funciones para crear, entrenar y simular redes neuronales. Además incluyen redes neuronales convolucionales y redes de aprendizaje profundo CITATION Pal17 \l 12298 (Paluszek & Thomas, 2017). Herramientas del sistema de visión artificial Tiene un amplio soporte para el procesamiento de video, pero también incluye funciones para la detección, extracción de características, es compatible con la visión tridimensional (3D), además permite procesar información desde cámaras estéreo CITATION The18 \l 12298 (The MathWorks,Inc, 2018).

## CAPÍTULO 3

## METODOLOGÍA

Metodología de la Investigación, es una herramienta en el campo de la investigación, que por su estructura y contenidos abordan los elementos básicos que guían un proyecto de manera metodológica y didáctica. Desde siempre, el hombre ha tenido por naturaleza la curiosidad y necesidad de conocer, entender y explicar el pasado, presente de su mundo vida, es decir, de los fenómenos naturales y sociales que lo rodean y producen. El primer contacto con el mundo se da por la sensación, percepción e intuición, para después pasar al juicio y la razón producto de la experiencia cotidiana o conocimiento teórico con el fin de entenderlas CITATION SEM05 \l 12298 (SEMAR, 2005). Es por ello que en nuestra investigación se emplearan las siguientes metodologías.

3.1 Metodología descriptiva Los estudios descriptivos buscan especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis. Es decir, únicamente pretenden medir o recoger información de manera independiente o conjunta sobre los conceptos o las variables a las que se refieren, esto es, su objetivo no es indicar cómo se relacionan éstas CITATION Her10 \l 12298 (Hernandez, Fernandez, & Baptista, 2010). Tienen como objetivo indagar la incidencia y los valores en que se manifiesta una o más variables (dentro del enfoque cuantitativo) o proporciona una visión de una comunidad, un fenómeno o una situación (describir como su nombre lo indica dentro del enfoque cuantitativo) CITATION Góm \p 103 \l 12298 (Gómez M. , 2006, pág. 103).

3.2 Metodología documental Es el método investigativo basado en la revisión de textos, artículos, bibliografías, videos, películas entre otros ya existentes sobre un tema y que pueden ser utilizadas para dar inicio o traer a flote un tema ya tratado. Allí se puede encontrar una investigación histórica hecha ya sobre el tema de interés CITATION Ressf \l 12298 ( Restrepo García, s.f). Desde el punto de vista de su evolución histórica, la investigación documental, tal como había sido definida, constituyó una serie de métodos y técnicas que los trabajadores en información descubrieron y perfeccionaron a lo largo de la historia con el propósito de ofrecer información a la sociedad CITATION Tansf \p 92 \l 12298 (Tancara Q. , s.f, pág. 92).

4.1 Ejemplos de aplicaciones comunes con Machine Learning en MATLAB Presentamos cinco tipos de algoritmos de aprendizaje automático. En cada caso mostramos un algoritmo simple para lograr los resultados deseados. 1. Redes neuronales Este ejemplo utilizará una red neuronal para clasificar los dígitos. La clasificación de dígitos es uno de los usos más antiguos del aprendizaje automático. La Oficina de Correos de EE. UU. Introdujo la lectura de códigos postales años antes de que el aprendizaje automático comenzara a aparecer en las portadas de todos los periódicos. Los lectores de dígitos anteriores requerían letras mayúsculas escritas en puntos bien definidos en un formulario. La lectura de dígitos en cualquier sobre es un ejemplo de aprendizaje en un entorno no estructurado. 2. Reconocimiento facial El reconocimiento facial está disponible en casi todas las aplicaciones de fotos. Muchos sitios de redes sociales, como Facebook y Google Plus, también utilizan el reconocimiento facial. Las cámaras tienen un reconocimiento facial incorporado, aunque no una identificación, para ayudar a enfocar cuando se toman retratos. Nuestro objetivo es conseguir que el algoritmo

coincida con las caras, no clasificarlas. El reconocimiento facial es un subconjunto del reconocimiento general de imágenes. Nuestro ejemplo de reconocimiento facial funciona dentro de un entorno estructurado. Todas las imágenes se toman desde el frente y la imagen solo muestra la cabeza. Esto hace que el problema sea mucho más fácil de resolver.

3. Clasificación de datos Este ejemplo utiliza un árbol de decisión para clasificar los datos. La clasificación de datos es una de las áreas más utilizadas del aprendizaje automático. En este ejemplo, asumimos que dos puntos de datos son suficientes para clasificar una muestra y determinar a qué grupo pertenece. Tenemos un conjunto de capacitación de puntos de datos conocidos con membresía en uno de tres grupos. Luego, utilizamos un árbol de decisiones para clasificar los datos. Presentaremos una pantalla gráfica para facilitar la comprensión del proceso. Con cualquier algoritmo de aprendizaje, es importante saber por qué el algoritmo tomó su decisión. Los gráficos pueden ayudarlo a explorar grandes conjuntos de datos cuando las columnas de números no son muy útiles.

4. Previsión anticipada de los tiempos de parada de la máquina y de la planta Las máquinas modernas y las instalaciones de producción a menudo tienen una vida útil de 20 años o más. Durante este tiempo, normalmente operan las 24 horas del día y los siete días de la semana. Cualquier interrupción no programada para trabajos de mantenimiento o reparación cuesta dinero y reduce el rendimiento del sistema o de la máquina.

IMAGEN El uso de MATLAB para el desarrollo de funciones para el mantenimiento predictivo garantiza que el usuario pueda recopilar y evaluar los datos adquiridos desde la máquina.

5. Detección de objetos en la visión por ordenador. La detección de objetos es el proceso de encontrar instancias de objetos del mundo real, como caras, bicicletas y edificios en imágenes o videos. Los algoritmos de detección de objetos suelen utilizar características extraídas y algoritmos de aprendizaje para reconocer instancias de una categoría de objetos. Se usa comúnmente en aplicaciones como los sistemas de recuperación de imágenes, seguridad, vigilancia y asistencia avanzada para el conductor (ADAS). Puedes detectar objetos usando una variedad de modelos, incluyendo:

- Detección profunda de objetos de aprendizaje.
- Detección de objetos basada en características

## IMAGEN

En este capítulo hemos observado los cinco tipos de aprendizaje automático referente a machine learning, recordemos que cada uno de ellos nos brindara resultados esperados en cuanto a la clasificación de los conjuntos. El tipo de aprendizaje automático en el cual nos vamos a centrar es el quinto: Detección de objetos en la visión por ordenador, a continuación se verá más al detalle conjuntamente con ejemplos que se encontraran en el cuarto capítulo.

Proceso para el reconocimiento de objetos Gómez (2015) En base al enfoque estándar de Machine Learning, al abordar un problema de reconocimiento de objetos se sigue un proceso básico de cinco pasos como se muestra en la Figura 8 (p. 2).

IMAGEN Adquisición de la imagen: Para la recopilación de imágenes se puede utilizar la base de datos organizada de ImageNet o directamente del Smartphone. Pre-procesamiento: Existen una gran variedad de procedimientos que permiten, a partir de una imagen, obtener otra modificada como se muestra en la figura 9 a esto se lo conoce como técnicas de filtrado. IMAGEN Algunas de estas técnicas están incluidas en el paquete de Image Processing Toolbox de Matlab que consta de un conjunto completo de algoritmos, funciones y aplicaciones dedicados el procesamiento, el

análisis y la visualización de imágenes, lo cual permite mejorar imágenes, reducción de ruido, transformaciones geométricas y registro de imágenes CITATION Gim16 \p 2 \l 12298 (Gimenez , Alemany M, & Monsoriu, 2016, pág. 2).

**Segmentación:** Una vez cambiada la imagen a “escala de grises” y posteriormente de ser filtrada se procede hacer la binarización. Esto representa al proceso que transforma una imagen que está en escala de grises a una imagen en dos colores: blanco y negro por medio de las funciones `im2bw` y `graythresh`, contenidas en Image Processing Toolbox de MATLAB. Por último, en la tarea de segmentación, conocida también como segmentación semántica, la salida de la red debe ser una máscara del mismo tamaño que la imagen original, donde el valor de cada pixel representa la clase a la que pertenece esa porción de la imagen CITATION Gar17 \p 5 \l 12298 (Garcia , 2017, pág. 5).

**Extracción de rasgos:** Finalmente se obtienen las características más significativas que permiten diferenciar los objetos a clasificar, los cuales describen la forma en que se distribuyen los píxeles de un objeto sobre el plano de la imagen. Esto se lo realiza con el algoritmo de extracción de características HOG con un modelo SVM.

**Clasificación de Objetos:** En la extracción de rasgos, un objeto es caracterizado mediante un vector que contiene características, denominado patrón, el cual está asociado a su etiqueta de clase, que indica la pertenencia a un tipo de objeto en específico entre diferentes clases de objetos CITATION Gim16 \l 12298 (Gimenez , Alemany M, & Monsoriu, 2016). Esta clasificación de objetos se lo realiza mediante el algoritmo KNN (k vecinos más cercanos) o Modelos bag-of-words. Algunas de estas técnicas están incluidas en el paquete de Image Processing ToolBox de Matlab que consta de un conjunto completo de algoritmos, funciones y aplicaciones dedicados el procesamiento, el análisis y la visualización de imágenes, lo cual permite mejorar imágenes, reducción de ruido, transformaciones geométricas y registro de imágenes CITATION Gim16 \l 12298 (Gimenez , Alemany M, & Monsoriu, 2016).

## CAPÍTULO 4

### DESARROLLO DEL TEMA

El reconocimiento de objetos es una técnica de visión artificial para encontrar e identificar objetos sean estos en una imagen o video CITATION The18 \l 12298 (The MathWorks,Inc, 2018). Para las personas, reconocer objetos en imágenes se lo realiza con absoluta facilidad, inclusive si los objetos sufren variaciones de forma, tamaño, localización, color, textura, brillo o están parcialmente obstruidos. Por ejemplo, en la Figura 10 se muestran varias imágenes en las cuales identificamos ‘gatos’. Entonces, ¿Cómo un computador puede identificar que son gatos y no otro objeto? ¿Qué características las identifican y las hacen únicas? IMAGEN En consecuencia para que un computador pueda reconocer una serie de fotografías, se pueden adoptar diversos enfoques en cuanto al reconocimiento de objetos. En consecuencia para que un computador pueda reconocer una serie de fotografías, se pueden adoptar diversos enfoques en cuanto al reconocimiento de objetos, por lo cual la posibilidad de probar distintos algoritmos para llevarlas a cabo se las describirá en los siguientes ejemplos.

**EJEMPLO 1: DETECCIÓN DE OBJETOS EN ESCENARIOS DESORDENADOS MEDIANTE LA FUNCIÓN DE PUNTOS DE COINCIDENCIA.** Este ejemplo muestra un algoritmo para la detección de un objeto específico basado en la búsqueda de correspondencias de puntos

entre la referencia y la imagen de destino. Puede detectar objetos a pesar de un cambio de escala o rotación en el plano. Para la detección de objetos específico basado en la búsqueda de correspondencias de puntos considerar los siguientes cinco pasos: 1) Leer imágenes Como se observa en la Figura, para leer y mostrar la imagen de referencia que contiene el objeto se utiliza el siguiente fragmento de código. `boxImage = imread('imagenMuestra.jpg'); figure; imshow(boxImage); title('Imagen de una Caja');`

IMAGEN Repetimos el fragmento de código con una variable y nombre diferente para leer la imagen de destino que contiene una escena de desorden como se observa en la figura 12 `sceneImage = imread('escritorioDesordenado.jpg'); figure; imshow(sceneImage); title('Imagen de una escena Desordenada');`

IMAGEN 2) Detectar puntos de características Con la lectura previamente de la imagen ahora se debe detectar los puntos característicos en ambas imágenes. `boxPoints = detectSURFFeatures(boxImage); scenePoints = detectSURFFeatures(sceneImage);`

Ahora se necesita visualizar los puntos más relevantes encontrados en la imagen como se muestra en la Figura 14. `figure;`

`imshow(boxImage); title('100 Strongest Feature Points from Box Image'); hold on; plot(selectStrongest(boxPoints, 100));`

IMAGEN Se procede hacer lo mismo para poder visualizar los puntos más relevantes de la característica encontrados en la imagen destino. `figure; imshow(sceneImage); title('300 Strongest Feature Points from Scene Image'); hold on; plot(selectStrongest(scenePoints, 300));`

IMAGEN 3) Extraer descriptores de la función Para extraer los puntos de las características detectados en ambas imágenes se realiza lo siguiente: `[boxFeatures, boxPoints] = extractFeatures(boxImage, boxPoints); [sceneFeatures, scenePoints] = extractFeatures(sceneImage, scenePoints);`

4) Encontrar supuestos puntos En cuanto a la coincidencia de las características se debe utilizar sus descriptores. `boxPairs = matchFeatures(boxFeatures, sceneFeatures);`

En la figura 15 se puede observar los puntos supuestamente coincidentes.

`matchedBoxPoints = boxPoints(boxPairs(:, 1), :); matchedScenePoints = scenePoints(boxPairs(:, 2), :); figure; showMatchedFeatures(boxImage, sceneImage, matchedBoxPoints, matchedScenePoints, 'montage'); title('Putatively Matched Points (Including Outliers)');`

IMAGEN

5) Localizar el objeto en la escena usando las supuestas coincidencias Con la función `estimateGeometricTransform` se calcula la transformación sobre los puntos coincidentes, eliminando valores atípicos como se observa en la figura 16. Esta transformación nos permite localizar el objeto en la escena. `[tform, inlierBoxPoints, inlierScenePoints] = ... estimateGeometricTransform(matchedBoxPoints, matchedScenePoints, 'affine');` `figure;`

```
showMatchedFeatures(boxImage, sceneImage, inlierBoxPoints, ... inlierScenePoints,  
'montage'); title('Matched Points (Inliers Only)');
```

## IMAGEN

Ahora se requiere obtener el polígono final de la imagen de referencia para después convertirlo en el sistema de coordenadas de la imagen de destino. Esto nos indica la ubicación del objeto en la escena. `boxPolygon = [1, 1;...`

```
size(boxImage, 2), 1;... size(boxImage, 2), size(boxImage, 1);... 1, size(boxImage, 1);... 1, 1];  
newBoxPolygon = transformPointsForward(tform, boxPolygon); .
```

Finalmente podemos observar los resultados obtenidos en la figura 17 en donde se muestra el objeto detectado. `figure; imshow(sceneImage); hold on; line(newBoxPolygon(:, 1), newBoxPolygon(:, 2), 'Color', 'y'); title('Detected Box');`

## IMAGEN

**EJEMPLO 2: CLASIFICACIÓN DE DÍGITOS USANDO CARACTERÍSTICAS HOG** En este ejemplo mostraremos cómo clasificar los dígitos mediante las funciones HOG y un clasificador SVM multiclase. La clasificación de objetos es una tarea importante en muchas aplicaciones de visión artificial, incluida la vigilancia, la seguridad automotriz y la recuperación de imágenes. Por ejemplo, en una aplicación de seguridad automotriz, es posible que deba clasificar los objetos cercanos como peatones o vehículos. Independientemente del tipo de objeto que se clasifique, el procedimiento básico para crear un clasificador de objetos es:

- Adquiera un conjunto de datos etiquetados con imágenes del objeto deseado.
- Partición del conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba.
- Entrena al clasificador utilizando las características extraídas del conjunto de entrenamiento.
- Pruebe el clasificador utilizando características extraídas del conjunto de pruebas.

Para ilustrar, este ejemplo muestra cómo clasificar los dígitos numéricos usando las características HOG (Histograma de Gradiente Orientado) y un clasificador SVL (Máquina de vectores de soporte) multiclase. Este tipo de clasificación se usa a menudo en muchas aplicaciones de reconocimiento óptico de caracteres (OCR).

El ejemplo utiliza la función `fitcecoc` de las herramientas de estadística y aprendizaje automático y la función `extractHOGFeatures` que pertenecen a las herramientas de visión artificial.

## Conjunto de datos de dígitos

Las imágenes de dígitos sintéticos se utilizan para el entrenamiento. Cada una de las imágenes de entrenamiento contiene un dígito rodeado por otros dígitos, que imita la forma en que los dígitos se ven juntos. El uso de imágenes sintéticas es conveniente y permite la

creación de una variedad de ejemplos de entrenamiento sin tener que recopilarlos manualmente. Para las pruebas, los escaneos de dígitos escritos a mano se usan para validar qué tan bien se desempeña el clasificador en datos que son diferentes a los datos de entrenamiento. Aunque este no es el conjunto de datos más representativo, hay suficientes datos para entrenar y probar un clasificador y mostrar la viabilidad del enfoque.

```
% Cargar datos de entrenamiento y prueba usando | imageDatastore |. syntheticDir = fullfile
(toolboxdir('vision'), 'visiondata','digits','synthetic'); handwrittenDir = fullfile(toolboxdir('vision'),
'visiondata','digits','handwritten');
```

```
% | imageDatastore | escanea recursivamente el árbol de directorios que contiene el % de
imágenes. Los nombres de las carpetas se utilizan automáticamente como etiquetas para
cada imagen. trainingSet = imageDatastore(syntheticDir, 'IncludeSubfolders', true,
'LabelSource', 'foldernames'); testSet = imageDatastore(handwrittenDir, 'IncludeSubfolders',
true, 'LabelSource', 'foldernames');
```

Utilizamos countEachLabel para tabular el número de imágenes asociadas con cada etiqueta. En este ejemplo, el conjunto de entrenamiento consta de 101 imágenes para cada uno de los 10 dígitos. El conjunto de prueba consta de 12 imágenes por dígito.

```
countEachLabel(trainingSet) countEachLabel(testSet) ans=10x2 table ans=10x2 table
```

La siguiente imagen Muestra algunas de las imágenes de entrenamiento y prueba. figure;

```
subplot(2,3,1); imshow(trainingSet.Files{102});
```

```
subplot(2,3,2); imshow(trainingSet.Files{304});
```

```
subplot(2,3,3); imshow(trainingSet.Files{809});
```

```
subplot(2,3,4); imshow(testSet.Files{13});
```

```
subplot(2,3,5); imshow(testSet.Files{37});
```

```
subplot(2,3,6); imshow(testSet.Files{97});
```

## IMAGEN

Antes de entrenar y probar un clasificador, se aplica un paso de pre procesamiento para eliminar los artefactos de ruido introducidos al recopilar las muestras de imágenes. Esto proporciona mejores vectores de características para entrenar al clasificador. % Mostrar resultados de preprocesamiento exTestImage = readimage (testSet, 37); processingImage = imbinarize (rgb2gray (exTestImage)); figura;

```
subparcela (1,2,1) imshow (exTestImage)
```

```
subparcela (1,2,2) imshow (processingImage)
```

IMAGEN Usando las características de HOG Los datos utilizados para entrenar al clasificador son vectores de características HOG extraídos de las imágenes de entrenamiento. Por lo tanto, es importante asegurarse de que el vector de características HOG codifique la cantidad correcta de información sobre el objeto. La `extractHOGFeatures` función devuelve una salida de visualización que puede ayudar a formar cierta intuición sobre qué significa exactamente la "cantidad correcta de información". Al variar el parámetro de tamaño de celda HOG y visualizar el resultado, puede ver el efecto que tiene el parámetro de tamaño de celda en la cantidad de información de forma codificada en el vector de características: `img = readimage (trainingSet, 206);`

```
% Extract características HOG y visualización HOG [hog_2x2, vis2x2] = extractHOGFeatures
(img, 'CellSize' , [2 2]); [hog_4x4, vis4x4] = extractHOGFeatures (img, 'CellSize' , [4 4]); [hog_8x8,
vis8x8] = extractHOGFeatures (img, 'CellSize' , [8 8]);
```

```
% Muestra la imagen original figura; subparcela (2,3,1: 3); imshow (img);
```

```
% Visualiza las características de HOG subparcela (2,3,4); parcela (vis2x2); title ({ 'CellSize = [2
2]' ; [ 'Length =' num2str (length (hog_2x2))]);
```

```
subtrama (2,3,5); parcela (vis4x4); title ({ 'CellSize = [4 4]' ; [ 'Length =' num2str (length
(hog_4x4))]);
```

```
subparcela (2,3,6); parcela (vis8x8); title ({ 'CellSize = [8 8]' ; [ 'Length =' num2str (length
(hog_8x8))]);
```

IMAGEN La visualización muestra que un tamaño de celda de [8 8] no codifica mucha información de forma, mientras que un tamaño de celda de [2 2] codifica una gran cantidad de información de forma, pero aumenta significativamente la dimensionalidad del vector de características HOG. Un buen compromiso es un tamaño de celda de 4 por 4. Esta configuración de tamaño codifica suficiente información espacial para identificar visualmente la forma de un dígito y limitar el número de dimensiones en el vector de características HOG, lo que ayuda a acelerar el entrenamiento. En la práctica, los parámetros HOG deben variarse con el entrenamiento y las pruebas repetidas del clasificador para identificar la configuración de parámetros óptima. tamaño de celda = [4 4]; hogFeatureSize = longitud (hog\_4x4);

Entrena a un clasificador de dígitos La clasificación de dígitos es un problema de clasificación multiclase, en el que debe clasificar una imagen en una de las diez posibles clases de dígitos. En este ejemplo, la `fitcecoc` función de Statistics and Machine Learning Toolbox se utiliza para crear un clasificador multiclase utilizando SVM binarios. Comenzamos por extraer las características HOG del conjunto de entrenamiento. Estas características serán utilizadas para entrenar al clasificador.

```
% Recorra el entrenamiento Ajuste y extraiga las características HOG de cada imagen. Se
utilizará un % de procedimiento similar para extraer características del testSet.
```

```
numImages = numel (trainingSet.Files); trainingFeatures = zeros (numImages, hogFeatureSize,
'single' );
```

```
para i = 1: numImages img = readimage (trainingSet, i);  
img = rgb2gray (img);  
% De aplicación de pasos de pre-procesamiento. img = imbinarize (img);  
trainingFeatures (i, :) = extractHOGFeatures (img, 'CellSize' , cellSize); fin  
% Obtener etiquetas para cada imagen. trainingLabels = trainingSet.Labels;
```

A continuación, entrena un clasificador usando las características extraídas. % fitcecoc usa estudiantes de SVM y un esquema de codificación 'Uno contra uno'.

```
clasificador = fitcecoc (trainingFeatures, trainingLabels);
```

Evaluación del clasificador de dígitos Evaluamos el clasificador de dígitos utilizando imágenes del conjunto de prueba y genere una matriz de confusión para cuantificar la precisión del clasificador. Como en el paso de entrenamiento, primero extraiga las características HOG de las imágenes de prueba. Estas características se utilizarán para hacer predicciones utilizando el clasificador entrenado.

```
% Extraer características HOG del conjunto de prueba. El procedimiento es similar a lo que  
% se mostró anteriormente y se encapsula como una función auxiliar para la brevedad.  
[testFeatures, testLabels] = helperExtractHOGFeaturesFromImageSet (testSet, hogFeatureSize,  
cellSize);
```

```
% Haz predicciones de clase usando las características de prueba. predictedLabels = predict  
(clasificador, testFeatures);
```

```
% Tabular los resultados utilizando una matriz de confusión. confMat = confusionmat  
(testLabels, predictedLabels);
```

```
helperDisplayConfusionMatrix (confMat)
```

La matriz de confusión en forma de porcentaje. Las columnas de la matriz representan las etiquetas predichas, mientras que las filas representan las etiquetas conocidas. Para este conjunto de pruebas, el dígito 0 a menudo se clasifica erróneamente como 6, probablemente debido a sus formas similares. Se ven errores similares para 9 y 3. La capacitación con un conjunto de datos más representativos como MNIST o SVHN, que contiene miles de caracteres escritos a mano, es probable que produzca un mejor clasificador en comparación con el creado con estos datos sintéticos conjunto.

## CONCLUSIONES

En base a las bibliografías analizadas se concluye que las máquinas tienen la capacidad de aprender de los datos por medio de algoritmos de Machine Learning, además pueden adoptar nuevos conocimientos para generar predicciones. Por lo cual los enfoques del

aprendizaje automático han otorgado a los sistemas informáticos habilidades totalmente nuevas, esta innovadora tecnología de vanguardia es ahora la fuerza impulsora de innumerables industrias por ello:

- El aprendizaje automático se utiliza para manejar datos multidimensionales y de múltiples variedades en entornos dinámicos.
- Procesamiento rápido y predicciones en tiempo real
- Proporciona una calidad continua con entornos de procesos grandes y complejos. Sin embargo Machine Learning no sólo es aplicable para el campo industrial su enfoque va mucho más allá por tal razón posee aplicaciones en casi todos los demás campos de estudio. Los criterios de validación para un algoritmo de aprendizaje automático son simples, pero se deben tener en cuenta numerosos factores como:

- ¿Qué algoritmo se adaptan mejor al problema o los datos?
- ¿Qué características deben incluirse?
- ¿Se pueden limpiar, transformar o refinar los datos para exponer mejor los elementos clave al modelo?
- ¿Cómo deben configurarse o ajustarse los parámetros del algoritmo para un rendimiento óptimo?

Haciendo estos cuestionamientos a nuestro enfoque seleccionado que fue el Reconocimiento de Objetos, esto se debe a que muchas de las últimas innovaciones tecnológicas dependen del reconocimiento de imágenes debido a que los algoritmos se vuelven más inteligentes con el tiempo. En este método se debe considerar que las imágenes son sometidas a un proceso de preparación a través de diferentes algoritmos en donde a partir de una imagen, obtener otra modificada por consiguiente se pueden eliminar las partes dañadas o quitar el exceso de iluminación, cambiar su tamaño para que su peso sea menor, transformar a escalas grises conocido todo este proceso permite el reconocimiento de los objetos en la imagen.

## Hit and source - focused comparison, Side by Side:

Left side: As student entered the text in the submitted document.

Right side: As the text appears in the source.

---

Instances from: CAPITULO II.docx

1 100%

Aprendizaje no supervisado Las redes con aprendizaje no supervisado (también conocido como

1: CAPITULO II.docx 100%

Aprendizaje no supervisado.- Las redes con aprendizaje no supervisado (también conocido como

2 100%

no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta.

2: CAPITULO II.docx 100%

no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta.

---