Revisión de algoritmos de *machine learning* y *deep learning* apropiados para la implementación de visión artificial y movimientos autónomos en un brazo robótico.

Pedro Daniel Ospina Arias

C.C: 1007415324

Asesor de Proyecto

Jairo Luis Gutiérrez Torres

Ingeniero Electrónico

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería

Tecnología en Desarrollo de Software

Octubre 2021

PAGINA DE ACEPTACION

Nota de aceptación

_____

_____

_____

_____

_____

Jurado

_____

Jurado

Dedicatoria

El presente trabajo está dedicado principalmente a Dios, en seguida a todas las personas que han sido participes de este proceso y que han hecho parte del desarrollo que se realizó en la institución asignada para diligenciar el proyecto.

Agradecimientos

Agradezco a la Universidad Nacional Abierta y a Distancia – UNAD, por haber generado la puerta para lograr mi meta de estudiar Desarrollo de Software.

A cada uno de los tutores que me acompañaron en las diferentes etapas de mi proceso y formación académica, mil gracias.

Resumen

Este informe presenta el trabajo de grado en la modalidad de proyecto de investigación, y como aporte al macro proyecto de investigación denominado “Diseño e implementación de herramienta robótica inteligente para robots de servicios e industria 4.0”. Las actividades fueron encaminadas a la búsqueda, prueba y análisis de algoritmos deInteligencia artificial, que permitieran, por medio de visión artificial, la detección de objetos en un espacio de trabajo determinado. Se inicia con una vigilancia tecnológica de los diferentes desarrollos e implementaciones que ha tenido la inteligencia artificial en la visión artificial de robots manipuladores. Luego, se procede a realizar pruebas, en el software MATLAB, de los algoritmos de *machine learning y deep learning* más destacadosen aplicaciones de visión artificial, específicamente, en la detección de objetos. Finalmente,se entrega un análisis de los datos obtenidos en cada uno de los algoritmos probados en la detección de imágenes y una propuesta para su implementación en el macro proyecto de investigación.

Abstract

This report presents the degree work in the form of a research project, and as a contribution to the research macro project called "Design and implementation of an intelligent robotic tool for service robots and industry 4.0". The activities were aimed at thesearch, test and analysis of artificial intelligence algorithms, which would allow, through artificial vision, the detection of objects in a given workspace. It begins with a technological surveillance of the different developments and implementations that artificialintelligence has had in the artificial vision of manipulative robots. Then, we proceed to carry out tests, in MATLAB software, of the most prominent machine learning and deep learning algorithms in artificial vision applications, specifically, in the detection of objects.Finally, an analysis of the data obtained in each of the algorithms tested in the detection of images and a proposal for its implementation in the main research project is delivered.

Palabras clave: algoritmos, inteligencia artificial, *machine learning*, *redes neuronales*, *big data.*

Tabla de Contenido

Tabla de Ilustraciones

Introducción

El presente documento se enmarca en el desarrollo del proyecto de investigación denominado “Diseño e implementación de una pinza robótica inteligente para robot de servicios e industria 4.0”, desarrollado por el semillero de investigación InvZing, en alianza con el centro de investigación del Servicio Nacional de Aprendizaje SENA - SENNOVA.

Se analizan algoritmos de *machine learning y deep learning* en el entorno de programación de Matlab, junto a un modelo 3D del efector final de un robot manipulador (Tipo mano de tres dedos). El robot contará con cámaras 3D y sensores de contacto, situados en la parte superior del robot y en cada una falange del efector, respectivamente. Estos sensores permiten obtener información relacionada con la localización de objetos dentro de un espacio de trabajo cercano al robot, así como sobre la cercanía con estos elementos.

Inicialmente se realiza una revisión del estado del arte, enfocada principalmente a desarrollos e investigaciones en donde se incluyen efectores finales de robots manipuladores y algoritmos de *machine learning y deep learning* utilizados en aplicacionesrobóticas.

La revisión del estado del arte permite identificar los principales adelantos en efectores finales y elegir las características principales para aplicar al diseño de la mano de tres dedos para el presente proyecto. Así mismo, se logran reconocer los algoritmos de inteligencia artificial que mejor comportamiento presentan en proyectos de robótica, visión artificial y movimientos en un plano tridimensional.

Planteamiento del problema

Nos encontramos en la era informática, en la que las empresas están en busca de robots autónomos que realicen muchas de las labores que ejecutan normalmente los operarios.

El aumento en la demanda de productos y el mejoramiento de la calidad de estos hallevado a muchas de las empresas a adquirir maquinaria de alto costo y con mayor rendimiento y precisión; sin embargo, estos equipos y maquinaria no son producidos o fabricados en Colombia lo que incrementa su costo y pone al país en un atraso en cuanto a la generación de tecnología a nivel industrial y de servicios.

Las últimas tecnologías están enfocadas en el uso de sistemas de control avanzados, sistemas robóticos y la programación y aplicación de algoritmos de inteligencia artificial, y en esta última, se le brinda la posibilidad a un robot de que aprenda, de manera automática, a realizar una acción dentro de un proceso o sistema. El proyecto busca motivar a los estudiantes para que se orienten hacia la investigación de aplicaciones con inteligencia artificial, aplicadas a procesos industriales y académicos.

Los robots manipuladores convencionales cuentan con efectores finales diseñados para operaciones específicas y el agarre de elementos de una contextura específica, lo que limita su aplicación en diferentes tareas. El objetivo de este proyecto es buscar que el efector final o en este caso la pinza con tres dedos pueda tomar una serie de objetos en una zona específica, y de diferentes materiales, sin mayor dificultad.

Pregunta de Investigación

¿De qué forma es posible seleccionar el algoritmo de Inteligencia Artificial más apropiado para la detección de objetos en el espacio de trabajo de robot manipulador?

Alcance

El proyecto de investigación se enfoca en buscar algoritmos de *machine learning y deep* learning aplicados a técnicas de visión artificial para robots manipuladores.

Una vez identificados los diferentes algoritmos de inteligencia artificial, se realiza una evaluación y clasificación de estos, mediante pruebas de detección de imágenes. Estas pruebas permitirán identificar los algoritmos más apropiados para ser implementados en el proyecto macro denominado “Diseño e implementación de herramienta robótica inteligente para robots de servicios e industria 4.0”.

Finalmente, el proyecto brindará recomendaciones sobre los algoritmos a implementar en el proyecto macro.

Justificación

Con la inteligencia artificial, aplicada a robots manipuladores, se busca recolectar información que pueda ser de ayuda para que un robot realice una o varias acciones definidas. Las tareas que realizan operarios en diferentes áreas a nivel industrial requieren de un margen de precisión alto, así como, acciones que son repetitivas para una persona. La inteligencia artificial aplicada a robots manipuladores autónomos pretende mejorar los niveles de calidad en los procesos industriales y reducir las enfermedades laborales asociadas a tareas repetitivas. Los robots manipuladores pueden llegar a realizar tareas específicas como la limpieza del hogar o tareas un poco más complejas como el ensamblaje en las industrias, repartición y empaque de productos en fábricas, actividades que generan un gran número de incapacidades por enfermedades laborales. En el año 2020, se presentaron un total de 50.947 personas con diagnóstico de enfermedad laboral, en Colombia. (Fuente: Minsalud. (2020). Indicadores de riesgos laborales. Colombia)

Objetivo general

Evaluar algoritmos de *machine learning* aplicados a visión artificial en un robot manipulador, como parte del proceso que busca obtener una herramienta robótica inteligente.

Objetivos específicos

- Realizar una revisión tecnológica sobre algoritmos de inteligencia artificial empleados en visión artificial para robots manipuladores industriales.
- Evaluar los algoritmos de *machine learning y deep learning*, que apliquen a la detección de objetos mediante visión artificial.
- Recomendar los algoritmos de *machine learning y deep learning* más apropiados para visión artificial en un robot manipulador.

Marco Teórico

Inteligencia artificial

La IA es la parte de las Ciencias de la Computación que tiene por objetivo diseñar sistemas informáticos inteligentes, es decir sistemas que exhiban las características que asociamos con la inteligencia humana: comprensión del lenguaje, aprendizaje, razonamiento solución de problemas etc... (E. A. Feigenbamm 1981)

Redes Neuronales

Son redes interconectadas masivamente contienen elementos simples (usualmente adaptativos) y presentan una organización jerárquica, las cuales interactúan con objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico. Las redes neuronales artificiales presentan un gran número de características semejantes a las del cerebro. (Matich, 2001)

"Las redes neuronales artificiales (ANN) posiblemente sean una de las herramientas computacionales más potentes actualmente disponibles para resolver problemas complejos. Se basan en el funcionamiento de las neuronas cerebrales y en la estructura del cerebro humano.

Estas redes neuronales están formadas por una colección de unidades de procesamiento altamente interconectadas llamadas neuronas, que trabajan conjuntamente alcanzando una gran capacidad de paralización.

Lo que hace a las ANN tan interesantes es su capacidad de replicar muchas de las características de la mente humana. Gran paralelismo, computación distribuida, capacidad de aprendizaje, generalización y adaptación son algunas de las características de esta herramienta.

Cada una de las neuronas que forman la red se conectan por lo que se denomina sinapsis (que son en este caso valores con un cierto peso). Cada neurona realiza algún tipo de cálculo (por ejemplo, una función sigmoide) y el resultado de este cálculo se multiplica por el peso de cada sinapsis según se propaga a lo largo de toda la red neuronal".(Tejada-Begazo, 2018)

Big Data

Hace referencia a aquellos sistemas de información que manejan conjuntos de datos de gran volumen, a una alta velocidad, de veracidad, de valor y de gran variedad de recursos, que demandan formas rentables e innovadoras de procesamiento de datos para mejorar la comprensión y la toma de decisiones. Optimiza el cálculo y precisión algorítmica al reunir, analizar, enlazar y comparar grandes conjuntos de datos permitiendo analizar dichas características en lapsos de tiempo mucho más cortas. (Meneses & Marcelo, 2021)

Visión artificial

Permite procesar y comprender imágenes del mundo real a través de una cámara a través del análisis de cualquier imagen para su comprensión. Tiene como objetivo generar descripciones inteligentes y útiles de escenas y secuencias visuales, así como de los objetos que aparecen en ellas, mediante la realización de operaciones sobre imágenes y vídeos. (neosentec, 2019)

Machine Learning

Los algoritmos han evolucionado a través del tiempo para analizar y mejorar como lo son: árboles de decisión, programación lógica inductiva (ILP), clustering para almacenar

y leer grandes volúmenes de datos, redes Bayesianas y un numeroso abanico de técnicas que los programadores de *data science* pueden aprovechar. Estos algoritmos interpretan los datos, para aprender de ellos y luego ser capaces de hacer una predicción o sugerencia sobre algo. (Rodríguez, 2018)

Aprendizaje Supervisado

Este tipo de aprendizaje consta de un entrenamiento a partir de una serie de datos establecidos, este proceso se realiza con el fin de validar las características del modelo entrenado, posteriormente se realiza un seguimiento a los datos algunos de los diagramas trabajados son árboles de decisión, *Support Vector*. (Grado, 2019)

Aprendizaje no supervisado

Para esta técnica no se toman en cuentan ningún conocimiento previo, a medida que se analizan los datos el clasificador asocia una serie de características comunes algunos de los métodos más conocidos *KMeans, KNN y Gaussian*. (Grado, 2019)

Deep Learning

Ofrece un modelo que permite evaluar ejemplos y una pequeña colección de instrucciones para modificar el modelo cuando se produzcan errores. Simula un sistema de redes artificiales de neuronas dentro del software de análisis de datos.

Cada neurona asigna un peso a la entrada, como un resultado correcto o incorrecto de forma relativa a su cometido. La salida estará determinada por la suma de esos pesos. (Rodríguez, 2018).

Algoritmo Q- Learning

Este tipo de algoritmo utiliza la experiencia para aprender la acción mas adecuada, a diferencia de otro tipo de modelos que intentan modelar el entorno y establecer una serie de parametros en un modelo previo.(Learning, 2017)

Algoritmo Deep Q Networks

Este algoritmo permite trabajar un nivel muy alto de caracteristicas, consta de una red neuronal convulucional para aproximar la función (Permite agilizar el proceso de extracción de imágenes). (Learning, 2017)

K nearest neighbors (KNN, vecinos más cercanos K)

Clasifica nuevas instancias como la clase mayoritaria de entre los k vecinos más cercanos de entre los datos de entrenamiento, durante el entrenamiento, sólo guarda las instancias, no construye ningún modelo. La clasificación se hace cuando llega la instancia de test; Es no paramétrico (no hace suposiciones sobre la distribución que siguen los datos, a diferencia de, por ejemplo, un modelo lineal). (Aler Mur, 2015)

Máquinas de soporte de vectores

Una Máquina de Soporte Vectorial (SVM) aprende la superficie decisión de dos clases distintas de los puntos de entrada. Como un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera. (Betancourt, 2005)

Metodología

Para el cumplimiento de los objetivos específicos de este proyecto se contemplan las siguientes fases metodológicas.

Fase 1: Búsqueda de algoritmos y revisión tecnológica

Se indaga sobre algoritmos de machine learning y Deep learning aplicados a visión artificial y para ello se utilizan bases de datos especializadas en investigación como Google *Academy, Scopus* e IEEE, obteniendo un conjunto de los principales algoritmos de IA.

Fase 2: Evaluación de algoritmos

Con los algoritmos seleccionados en la fase 1, aplicados a la detección de imágenes, se procede a realizar pruebas de eficiencia para la clasificación, detección y localización de objetos.

Fase 3: Recomendación de algoritmos de machine learning y Deep learning

Con base a los algoritmos encontrados y evaluados en las fases anteriores, se recomiendan los algoritmos más apropiados de inteligencia artificial para la detección y localización de objetos dentro del espacio de trabajo de un robot manipulador.

Desarrollo

Fase 1: Búsqueda de algoritmos y revisión tecnológica

Principales Algoritmos de Machine Learning

Árboles de decisiones y bosques aleatorios

El uso de árboles de decisión tuvo su origen en las ciencias sociales con los trabajos de Sonquist y Morgan (1964), y Morgan y Messenger (1979) realizado en el *Survey Research Center del Institute for Social Research* de la Universidad de Michigan. Pero, en el campo del aprendizaje automático, no fue hasta 1984 cuando Breiman, Friedman, Olshen y Stone introdujeron el algoritmo conocido como CART (*Classification And Regression Trees*) para la construcción de árboles y se aplicó a problemas de regresión y clasificación. Casi al mismo tiempo, el proceso de inducción mediante árboles de decisión comenzó a ser usado por la comunidad de aprendizaje automático (Michalski, (1973), Quinlan (1983)). Uno de los árboles de decisión más conocidos, el C4.5, fue introducido por Quinlan en 1993 y desciende del ID3, también creado por Quinlan en 1983. Los arboles de decisión se emplean con frecuencia en operaciones de búsquedas, especialmente en análisis de decisión, para ayudar a identificar la estrategia con mayor probabilidad de alcanzar un objetivo. Otro uso de los árboles de decisión es el de un medio descriptivo para calcular probabilidades condicionales. (V, 2009)

Agrupamiento (Clustering): Medios k, análisis de agrupamiento jerárquico

El agrupamiento se puede considerar como la aproximación más utilizada en aprendizaje no supervisado. Su objetivo general es encontrar algún tipo de estructura en una colección de datos sin etiquetar o sin clasificar, ya que en la mayoría de los casos no se dispone de este tipo de información. Los algoritmos de agrupamiento buscan organizar objetos en distintos grupos cuyos miembros tienen características similares. Un clúster o grupo es por tanto una colección de objetos que ´ son similares entre ellos y diferentes respecto a los miembros de otros grupos.(V, 2009).

Método del vecino más cercano (*K Nearest neighbors- k-NN*)

Una forma práctica y de fácil aplicación para predecir o clasificar un nuevo dato, basado en observaciones conocidas o pasadas, es la técnica del vecino más cercano. A manera de ejemplo, el caso de un médico que está tratando de predecir el resultado de un procedimiento quirúrgico puede predecir que el resultado de la cirugía del paciente será aquel del paciente más parecido que conoce, que haya sido sometido al mismo procedimiento. Esto puede resultar un tanto extremo, ya que un solo caso similar en el cual la cirugía falló puede influir de manera excesiva sobre otros casos, ligeramente menos similares, en los cuales la cirugía fue un éxito. Por esta razón el método del vecino más cercano se generaliza a uso de los *k* vecinos más cercanos.

Esta técnica se basa, simplemente, en recordar todos los ejemplos que se vieron en la etapa de entrenamiento. Cuando un nuevo dato se presenta al sistema de aprendizaje, este se clasifica según el comportamiento del dato más cercano. (Mora-florez & Barrera-cárdenas, 2008)

Visualización y Reducción de Dimensionalidad: Núcleo PCA, distribuido de t PCA.

T-SNE

Es un método cuyo objetivo es la visualización de grandes conjuntos de datos en alta dimensión y dar una localización coordenada a coordenada en un plano bidimensional o tridimensional. Esta técnica es una variante de la "*Stochastic Neighbor Embedding*", aunque mucho más fácil de optimizar y elimina la tendencia de agrupar los puntos en el centro del mapa. La función de coste de t-SNE se centra en mantener la estructura local de los datos en el mapa. La minimización de la función de coste se consigue haciendo la primera derivada. En regiones densas un pequeño valor de sigma es normalmente más apropiado que en regiones dispersas. Esta distribución tiene una entropía que aumenta si sigma aumenta. La fuerza ejercida por el resorte entre X e Y es proporcional a su longitud y también proporcional a su rigidez, que es la falta de coincidencia entre las similitudes de las parejas de puntos de datos y los puntos del mapa. En las primeras etapas de optimización, el ruido gaussiano es añadido al mapa de puntos después de cada interacción.(Mecánica et al., 2012)

Máquinas de Vectores Soporte (SVM)

Las máquinas de vectores soporte (SVM, del inglés *Support Vector Machine*) tienen su origen en los trabajos sobre la teoría del aprendizaje estadístico y fueron introducidas en los años 90 por Vapnik y sus colaboradores Boser et al. (1992); Cortes & Vapnik (1995). Aunque originariamente las SVMs fueron pensadas para resolver problemas de clasificación binaria, actualmente se utilizan para resolver otros tipos de problemas (regresión, agrupamiento, multi-clasificación). También son diversos los campos en los que han sido utilizadas con éxito, tales como visión artificial, reconocimiento de caracteres,

categorización de texto e hipertexto, clasificación de proteínas, procesamiento de lenguaje natural, análisis de series temporales. Dentro de la tarea de clasificación, las SVMs pertenecen a la categoría de los clasificadores lineales, puesto que inducen separadores lineales, también llamados hiperplanos, ya sea en el espacio original de los ejemplos de entrada, si éstos son linealmente separables o cuasiseparables (ruido), o en un espacio transformado (espacio de características), si los ejemplos no son linealmente separables en el espacio original. Como se verá más adelante, la búsqueda del hiperplano de separación en estos espacios transformados, normalmente de muy alta dimensión, se hará de forma implícita utilizando las denominadas funciones kernel. Mientras la mayoría de los métodos de aprendizaje se centran en minimizar los errores cometidos por el modelo generado a partir de los ejemplos de entrenamiento (error empírico), el sesgo inductivo asociado a las SVMs radica en la minimización del denominado riesgo estructural. La idea es seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para, de esta forma, conseguir lo que se denomina un margen máximo a cada lado del hiperplano. Además, a la hora de definir el hiperplano, sólo se consideran los ejemplos de entrenamiento de cada clase que caen justo en la frontera de dichos márgenes. Estos ejemplos reciben el nombre de vectores soporte. Desde un punto de vista práctico, el hiperplano separador de margen máximo ha demostrado tener una buena capacidad de generalización, evitando en gran medida el problema del sobreajuste a los ejemplos de entrenamiento. Desde un punto de vista algorítmico, el problema de optimización del margen geométrico representa un problema de optimización cuadrático con restricciones lineales que puede ser resuelto mediante técnicas estándar de programación cuadrática. La propiedad de convexidad exigida para su resolución garantiza una solución única, en

contraste con la no unicidad de la solución producida por una red neuronalartificial. (Carmona, 2016).

Revisión Tecnológica

En el año 2018, Robinson Jiménez Moreno, realizó su tesis doctoral denominada "Estructuración de un Algoritmo Basado en Deep Learning para Entrenamiento de Robots Asistentes en Reconocimiento de Objetos para Plataformas Multiherramienta", en la cual, se desarrolló un algoritmo aplicado a robótica asistencial y múltiples procesos, dicho proceso hace referencia al trabajo realizado por un robot y que anteriormente era ejecutado por una persona. El reconocimiento que realiza a la hora de aprender distintas herramientas por medio de una cámara. Al reconocer y establecer la herramienta se ejecutan una serie de algoritmos de trayectorias, que permiten el movimiento de un efector final, éste realiza y traza el camino para desplazarse hasta la herramienta de trabajo. (Basado et al., 2018).

De acuerdo con el artículo " *Robotic Arm Control and Task Training through Deep Reinforcement Learning*" se realiza un paralelo sobre las posibles mejoras de políticas de Red profunda, respecto a otro tipo de algoritmos, este paralelo permite observar cuales son los mejores algoritmos para ser aplicados a un robot manipulador, al realizar la comparación de los algoritmos se establece que los experimentos simulados y en el mundo real deben realizarse para demostrar que se implementa el uso adecuado de las políticas de red. (Franceschetti et al., 2020)

El artículo "*End-to-End Robotic Reinforcement Learning without Reward Engineering*" menciona la unión de modelos de redes neuronales, algoritmos de aprendizaje que hacen posible aprender políticas para comportamientos de robots, también se hace para un mejor reconocimiento de imágenes; se debe tener en cuenta que las aplicaciones que

utilizan el aprendizaje por refuerzo se basan en recompensas programadas, se propone un enfoque que busca quitar parte del ciclo de recompensas con el fin de que el robot aprenda un numero de resultados más exactos, el método propuesto solicita una serie de requerimientos al iniciar el entrenamiento y que permite al enfoque ser más eficaz, utilizando este proceso se establece un orden en los resultados, sin necesidad de recurrir a la definición previa del algoritmo. (Singh et al., 2019)

En el año 2017, Francisco de la Vieja realizo su trabajo de grado denominado "Coordinación Entre Brazo Robótico Y Cámara Usando *Deep Reinforcement Learning*", la tecnología avanza exponencialmente y realiza un papel fundamental en nuestra vida, entre los desarrollos más recientes se encuentra la implementación de inteligencia artificial a robots, el proyecto muestra el desarrollo de un brazo robótico programado con algoritmos de Deep Q-learning, dichos algoritmos permiten al robot aprender por medio de redes neuronales estas analizan y clasifican una serie de datos con el fin de optimizar el proceso e indicar al robot que movimientos debe ejecutar. En base a los resultados obtenidos se concluye que él algoritmo cumple con los parámetros propuestos, gracias a las pruebas realizadas en los entornos propuestos y en otros tipos de entornos que no se consideraron probar al inicio, los algoritmos y la capacidad de entrenamiento de robots requieren de una mayor investigación para ampliar los resultados en las áreas de Deep learning y machine learning.(Learning, 2017)

De acuerdo con el artículo " *Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates*" expone el alcance del aprendizaje por refuerzo, el cual busca que robots autónomos aprendan una variedad de comportamientos y acciones sin la necesidad de participación humana, para este tipo de aprendizajes se busca la autonomía del proceso para obtener entrenamientos favorables, la

primera intervención debe ser realizada por una persona; Estos tipos de aprendizajes son realizados en ambientes simulados y ejecutan tareas sencillas, también se demostró que los algoritmo de Q-funciones permiten trabajar modelos 3D complejos junto a las redes neuronales de manera eficiente, los algoritmos de entrenamiento pueden reducirse distribuyendo los procesos entre diferentes robots que cumplan los requisitos establecidos.(Gu et al., 2018)

En el año 2018 María Fernanda Tejada Begazo realizo su trabajo de grado "Control Visual de un Brazo Manipulador con 7GDL, en Base Visión Monocular, para el Seguimiento de Objetivos" expone el aumento de producción de la revolución industrial el impacto tecnológico que influyo en el trabajo del hombre la aplicación de máquinas en el sector industrial permitió el aumento de trabajo, el uso de brazo manipuladores se observa en las diferentes industrias y fabricas para que realizarán trabajos continuos y repetitivos, en la nueva era se vieron más posibilidades para el uso de brazos manipuladores en el campo de la medicina, trabajo industrial entre otros. El objetivo del manipulador es alcanzar elementos en un área de trabajo, esto proceso se realiza por medio de procesos y funciones que permiten al robot realizar diferentes tipos de movimientos, también se implementaron sensores para detección de objetos; se trabaja en un eje tridimensional y bidimensional para tener diferentes ángulos de enfoque respecto a la cámara tomada.(Tejada-Begazo, 2018)

De acuerdo con el artículo "Visión para Robots en Tareas de Ensamble", se propone un sistema de visión que permita a un robot obtener una serie de datos para realizar una clasificación en tiempo real, el sistema y arquitectura presenta un método de recolección de información de la posición y el eje de orientación en tiempo real de imágenes recolectadas por técnicas de fragmentación y análisis de diagramas. El manejo de imágenes binarias

permite al sistema ser práctico y de fácil manejo, este proceso permite procesar datos y generar figuras de menor complejidad. (Peña et al., 2015)

El artículo “VISUAL: Herramienta para la Enseñanza Práctica de la Visión Artificial” menciona una temática practica para el aprendizaje de visión artificial, describe la herramienta Visual, la cual permite al usuario detallar algoritmos de recolección de características de imágenes en diagramas compuestos. (Gil et al., n.d.)

En el año 2021 Damian Marcelo Gutierrez Meneses realizo el proyecto de grado “Diseño y desarrollo de un sistema de video vigilancia basado en dispositivos emebidos técnicas de visión artificial y algoritmos inteligentes” realiza un sistema de reconocimiento facial, el cual trabaja por medio de dispositivos que aplica técnicas de visión artificial, utilizo una arquitectura libre para la identificación de personas en tiempo real, como primera fase define el reconocimiento y localización de rostros, posteriormente se investigaron y estudiaron las librerías de OpenCV y dlib las cuales le permitieron comprender y utilizar los datos recolectados, también se realizó seguimiento a diferentes artículos y estudios para tener una perspectiva mucho más clara. (Meneses & Marcelo, 2021)

En el año 2020 Kevin Arturo Ramirez Zavalza realizo su tesis “Robot inteligente recolector de basura asistido por redes neuronales artificiales”, expone el punto de partida de un robot recolector de desechos basado en algoritmos de inteligencia artificial, busca generar una alternativa eficiente ante la contaminación de desechos, el robot es capaz de moverse libremente en cualquier lugar y recoger los desechos que se encuentren allí; el entrenamiento de redes neuronales fue el eje principal del proyecto, en la parte hardware se

realizó una selección entre varios tipos con el fin de encontrar el que más cumpla con requerimientos propuestos, finalmente se trabajó un prototipo que cumplía con la gran mayoría de funciones definidas inicialmente. (Arturo & Zavalza, n.d.)

Fase 2 / Evaluación de algoritmos

Para el presente trabajo se utiliza un equipo de cómputo con una serie de características especiales para la aplicación de algoritmos de machine learning y Deep learning. Las cuales se relacionan a continuación (tabla 1):

Tabla 1.

Características del computador con el que se realizan las pruebas de desempeño de algoritmos de machine learning con los diferentes grupos de imágenes.

Procesador:	Intel(R) Core(TM) i5-8250U CPU @1.60GHz 1.80 GHz
RAM:	12.00 GB
Tipo de sistema Operativo:	Windows 10 – Sistema Operativo 64-bits procesador x64

Fuente: Autor

El proyecto inicia con la recopilación de imágenes de diferentes objetos, en diferentes ángulos y niveles de iluminación (Ilustración 1). Se crearon carpetas con 7, 100 y 167 imágenes lo que constituyo el insumo principal para la obtención de datos.

Ilustración 1.

Arandela, objeto seleccionado para realizar las pruebas con los algoritmos de machine learning.

Fuente: Autor

Para el desarrollo de la fase de revisión y evaluación de algoritmos se utilizó el programa de Matlab Versión 2021b, el cual permitió trabajar con una serie de librerías de *machine learning y deep learning*. A continuación, se encuentra el código realizado en Matlab para la búsqueda de características en los archivos de imágenes:

1. Imagenes=imageSet('Arandela', 'recursive');
2. bag=bagOfFeatures(Imagenes);
3. imageFeature=encode(bag, Imagenes);
4. Datos_Imagen=array2table(imageFeature);
5. Datos_Imagen_Tipo=getImageLabels(Imagenes);

La primera línea del código anterior, permite extraer las imágenes de la carpeta denominada arandela.

La línea dos, permite extraer las características de las imágenes.

La línea tres, codifica los datos de las imágenes, generando una cadena de bits con los datos encontrados en cada imagen.

La línea cuatro, genera una serie de vectores con los datos encontrados en las imágenes.

La línea cinco etiqueta las características extraídas en las imágenes.

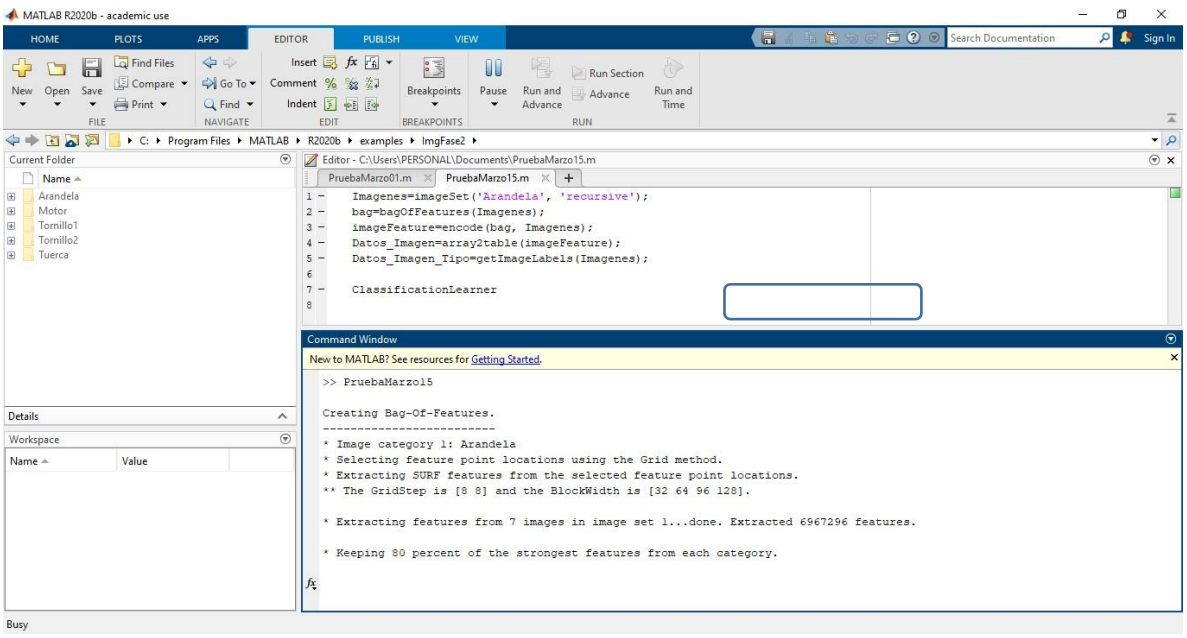
Prueba para un conjunto de 7 imágenes

Una vez ejecutado el algoritmo de machine learning en el software de Matlab, este creará los distintos grupos de datos, comenzará a extraer características en las imágenes recolectadas y generará información importante para la posterior aplicación de algoritmos de *machine learning y deep learning*.

El total de características extraídas para 7 imágenes es de 6’967.296.

Ilustración 2.

Extracción de características para un conjunto de 7 imágenes



Fuente: Autor

El algoritmo muestra el número de características y el porcentaje de extracción hasta el momento, así como, el número agrupaciones realizadas y total de características encontradas.

Tabla 2.

Datos generados para un conjunto de 7 imágenes

Datos imágenes de arandelas	
Total imágenes	7
Características extraídas	6967296
Características válidas	55373837
Numero de clusters	500
Tiempo de procesamiento	1h 56 min

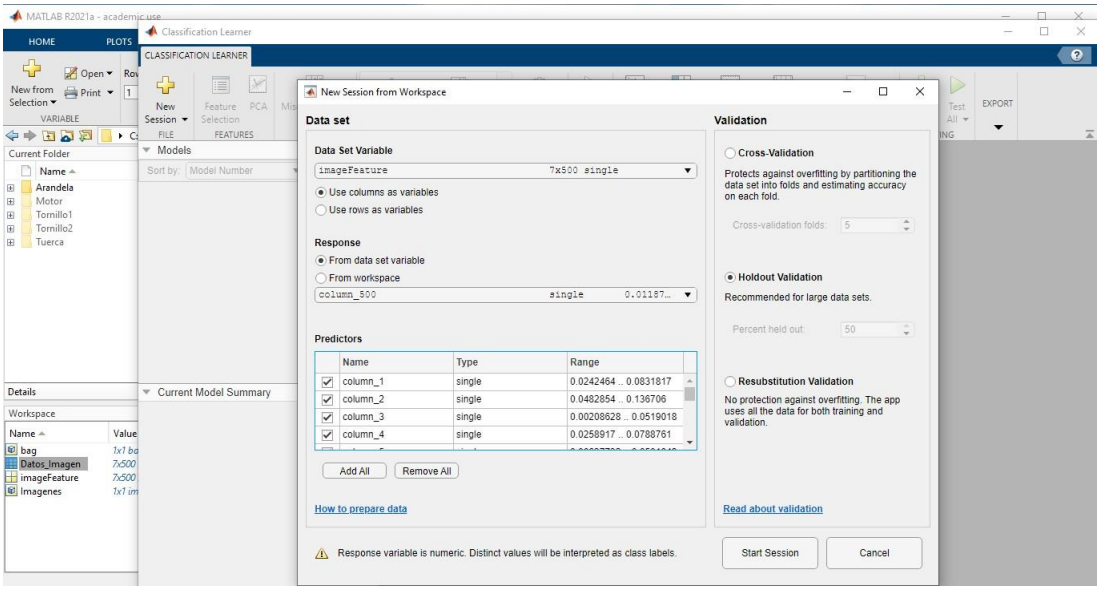
Fuente: Autor

Al ejecutarse el programa, se inicializa en segundo plano las agrupaciones y las características encontradas hasta el momento en cada una de las imágenes, una vez terminado una agrupación, continuará con la siguiente hasta que el proceso haya finalizado para cada una de las imágenes propuestas.

Una vez completado las interacciones, daremos clic en la aplicación *Classification Learner*, este modelo clasifica los datos por medio del aprendizaje automático supervisado, permite también seleccionar distintos modelos de clasificación como diagramas regresión logística, de árbol de decisión, clasificación de conjuntos redes neuronales entre otros; el aprendizaje supervisado utiliza un conjunto de datos de entrada que utiliza como ejemplo para entrenar datos y generar predicciones a respuestas de datos nuevos.

Ilustración 3.

Classification learner para el conjunto de 7 imágenes

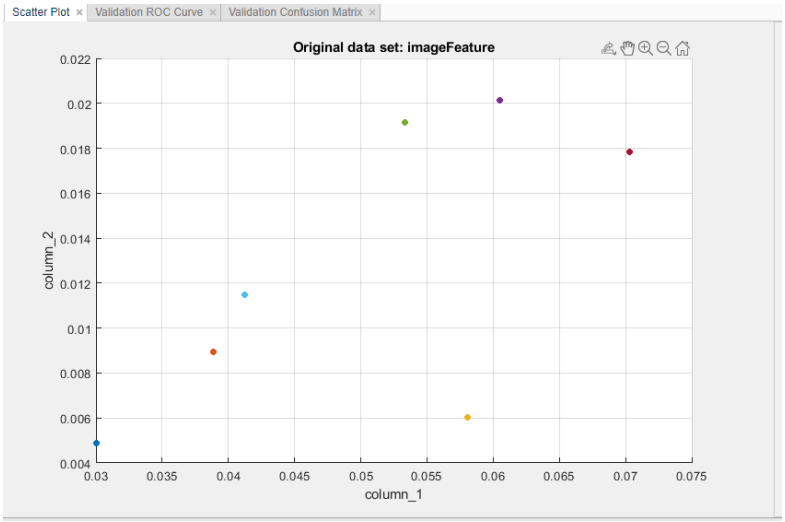


Fuente: Autor

Una vez se inicia la aplicación *classification learner*, se da clic en el diagrama de árbol de decisiones, posteriormente se cargará el diagrama y el porcentaje de efectividad, este tipo de algoritmo potencian los modelos con una alta precisión, estabilidad y facilidad de interpretación, son adaptables a cualquier tipo de problema, tienen la estructura de un diagrama de flujo, utilizan nodos y divisiones en su desarrollo.

Ilustración 4.

Modelo árbol de decisiones

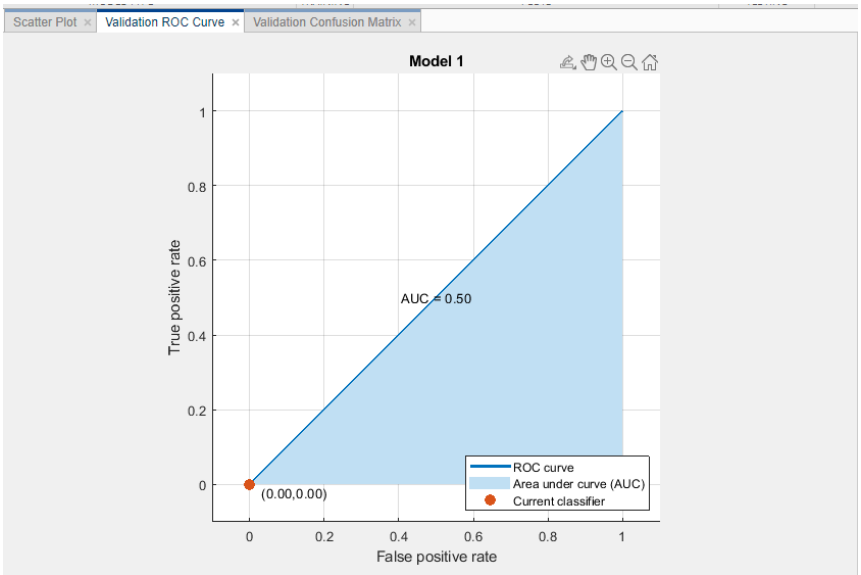


Fuente: Autor

La curva de ROC indica el porcentaje de rendimiento de la muestra seleccionada, dependiendo el nivel de aproximación será clasificado como (buena, regular o mala). El porcentaje de curva no es representativo para la medición de efectividad del algoritmo; este es determinado por la relación del porcentaje de valores positivos y negativos, estos permiten predecir si el diagrama se inclina al punto exacto, en donde observaríamos un valor de (1) para el caso positivo y un valor de (0) para la variable negativa, la curva de roc muestra la gráfica dependiendo del número de imágenes entrenadas en algunos casos se requiere adjuntar más imágenes pero en otros el modelo muestra si cuenta con suficientes imágenes en el entrenamiento si la curva tiende a (1).

Ilustración 5.

Curva ROC Modelo árboles de decisiones

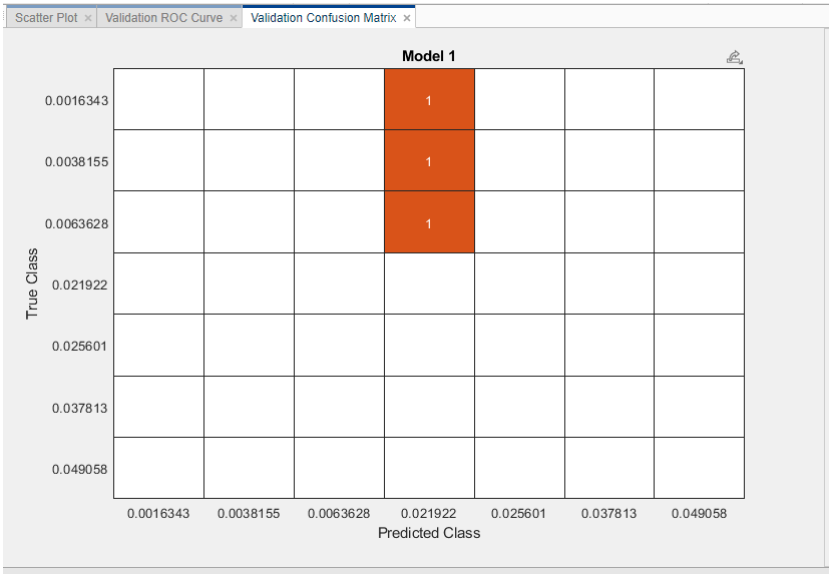


Fuente: Autor

La gráfica de matriz de confusión permite comprender cómo se desempeñó el clasificador seleccionado en cada clase. Cuando abre el gráfico, las filas muestran la clase verdadera y las columnas muestran la clase predicha. Si utiliza la validación retenida o cruzada, la matriz de confusión se calcula utilizando las predicciones de las observaciones retenidas (validación). Las celdas diagonales muestran dónde coinciden la clase verdadera y la clase predicha. Si estas celdas diagonales son azules, el clasificador ha clasificado correctamente las observaciones de esta clase verdadera.

Ilustración 6.

Matriz de confusión Modelo árboles de decisiones

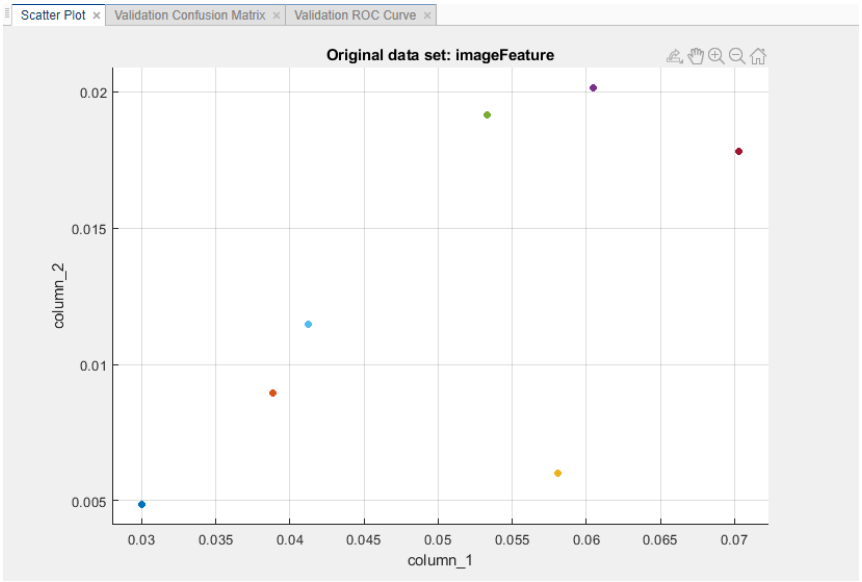


Fuente: Autor

En el modelo lineal SVM se observa que los puntos se encuentran muy dispersos al realizar el entrenamiento, por lo que el porcentaje de validez de características será muy bajo.

Ilustración 7.

Modelo Lineal SVM

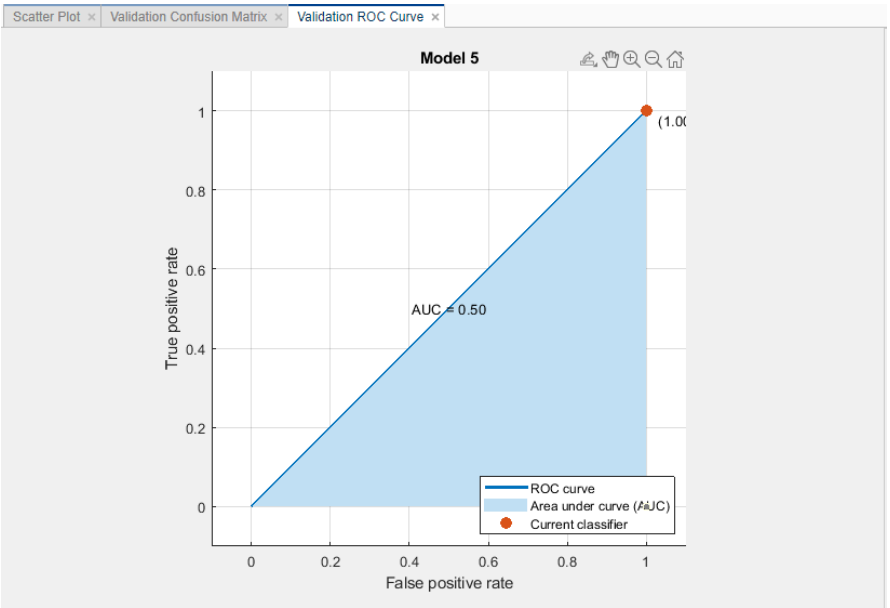


Fuente: Autor

La curva de ROC para el modelo Lineal SVM (máquina de vectores de soporte) decisiones no cuenta con el porcentaje de imágenes requerido por lo que el modelo no pasa de 0.50 y este debería de ser mayor para obtener mejores resultados.

Ilustración 8.

Curva de ROC Modelo Lineal SVM

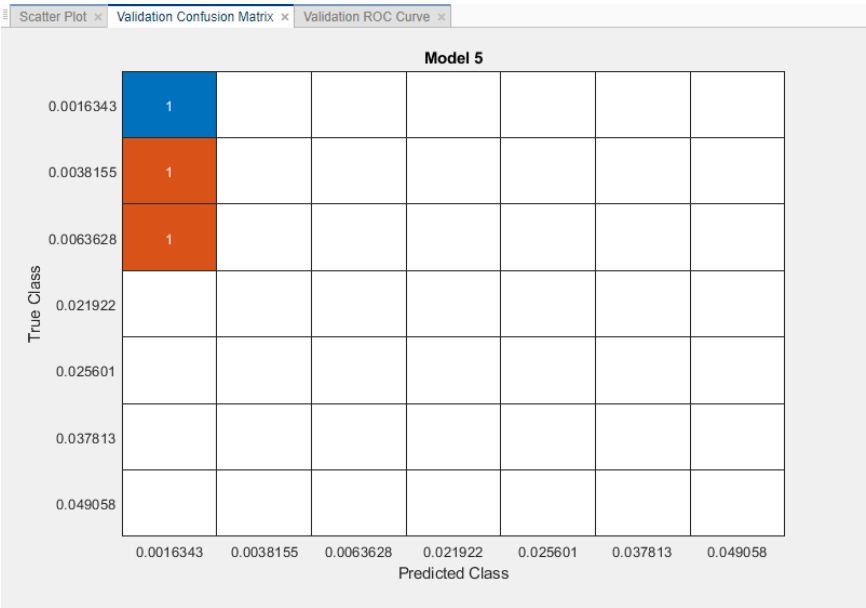


Fuente: Autor

Para la matriz de confusión del modelo lineal SVM, se observa que no cuenta con la diagonal en donde indique que todos los valores son verdaderos, cuentan con un valor que hace alusión a los falsos positivos, en el diagrama los reconoce como positivos, pero no deberían de aparecer ya que dichos valores no son verdaderos.

Ilustración 9.

Matriz de confusión Modelo Lineal SVM

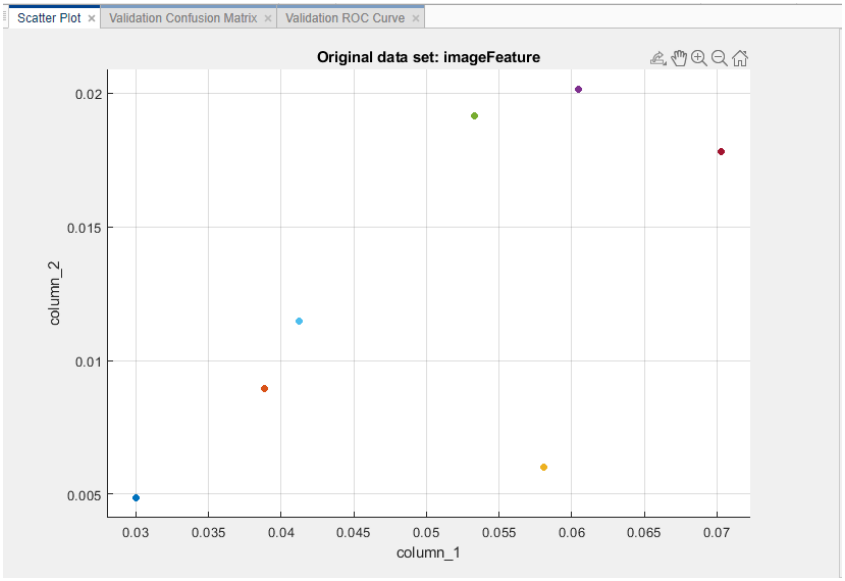


Fuente: Autor

Para el modelo de Teoría de Bayes se observa que las características encontradas se encuentran dispersas, en el modelo por lo que el diagrama no puede mostrar resultados favorables para el entrenamiento de imágenes.

Ilustración 10.

Modelo Teoría de Bayes

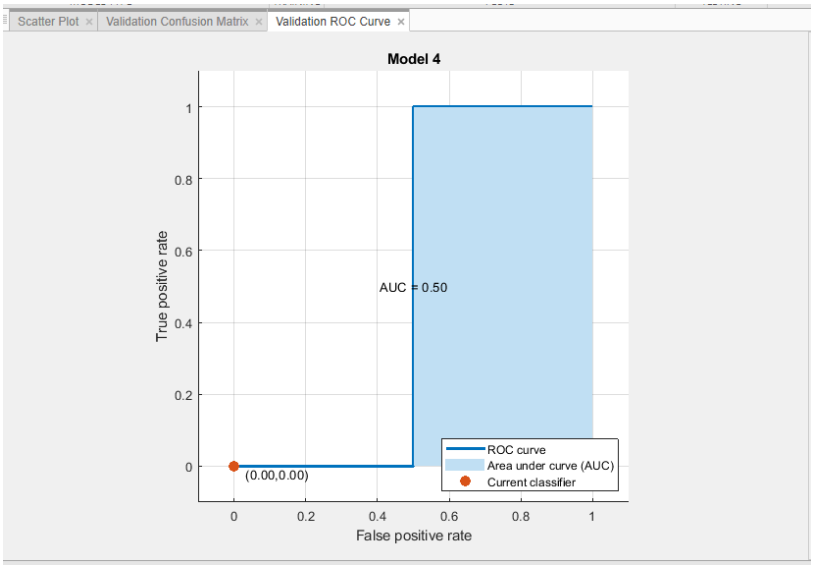


Fuente: Autor

La curva de ROC en el Modelo de teoría de Bayes muestra un porcentaje de 0.50 lo que nos indica que está por debajo de la mitad, este no muestra una aproximación al 1, ya que el número de imágenes utilizadas para el entrenamiento del algoritmo es muy pequeño.

Ilustración 11.

Curva ROC Modelo Teoría de Bayes

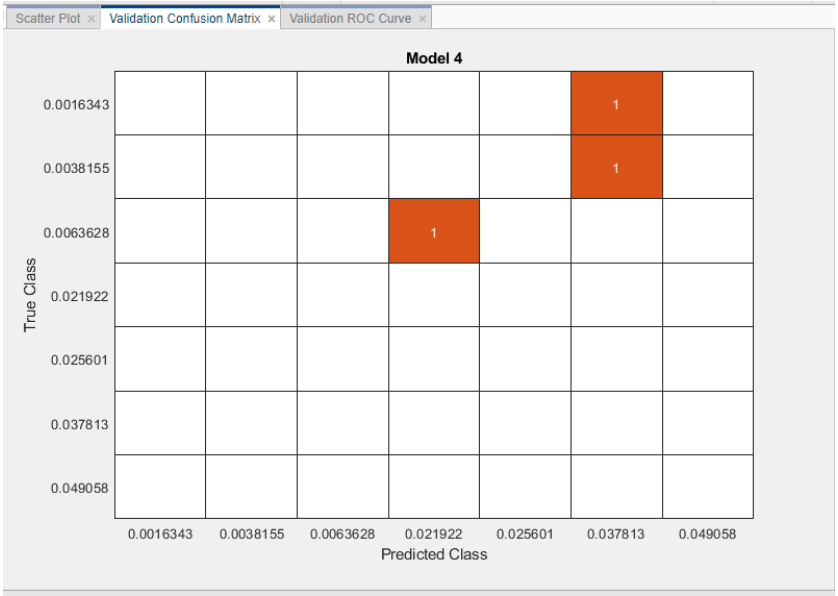


Fuente: Autor

La matriz de confusión para el modelo de teoría de Bayes no muestra la diagonal de los valores positivos, muestra otros valores que reconoce como verdaderos, pero para el diagrama no sabe dónde clasificarlos en la matriz.

Ilustración 12.

Matriz de Confusión Modelo Teoría de Bayes

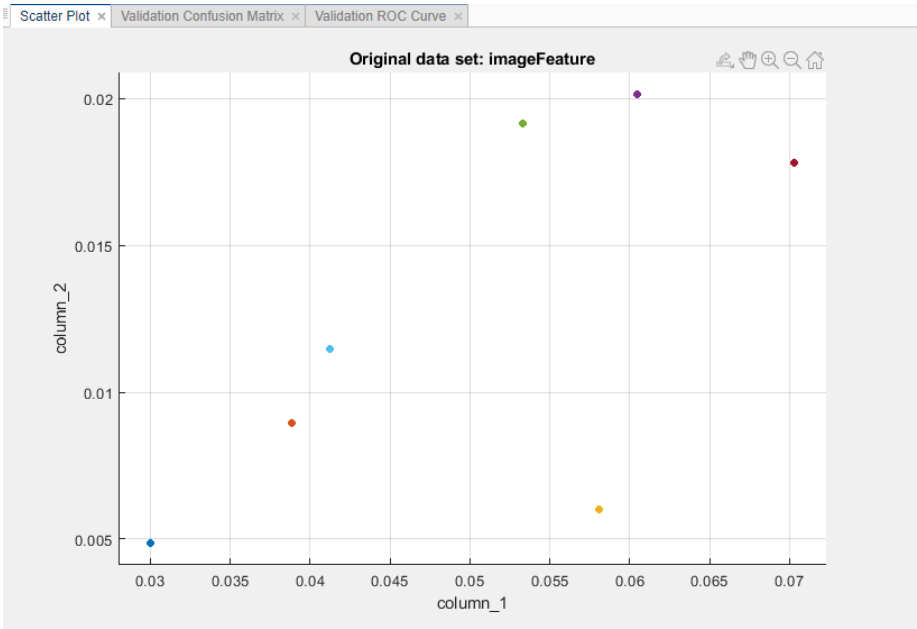


Fuente: Autor

En el modelo Métodos Gaussianos los datos entrenados se encuentran muy dispersos, ya que las características encontradas no fueron clasificadas de manera correcta, este modelo no cumple con el objetivo de los algoritmos de detección de objetos.

Ilustración 13.

Modelo Métodos Gaussianos

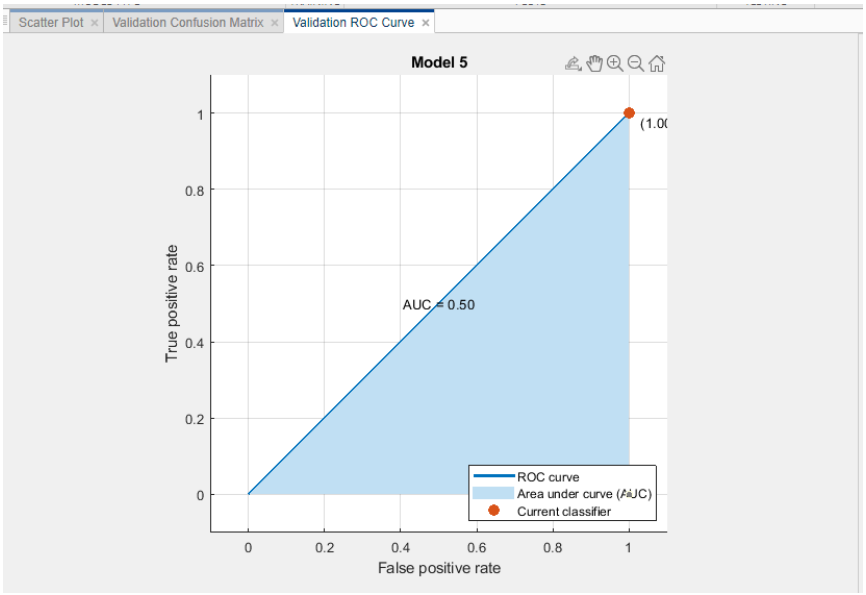


Fuente: Autor

En la curva de ROC del modelo Métodos Gaussianos, se observa que el porcentaje de imágenes requerido por lo que el modelo no pasa de 0.50, este diagrama no se aproxima al 1, lo que indica que requiere de una mayor cantidad de imágenes para una mejor predicción.

Ilustración 14.

Curva ROC Modelo Métodos Gaussianos



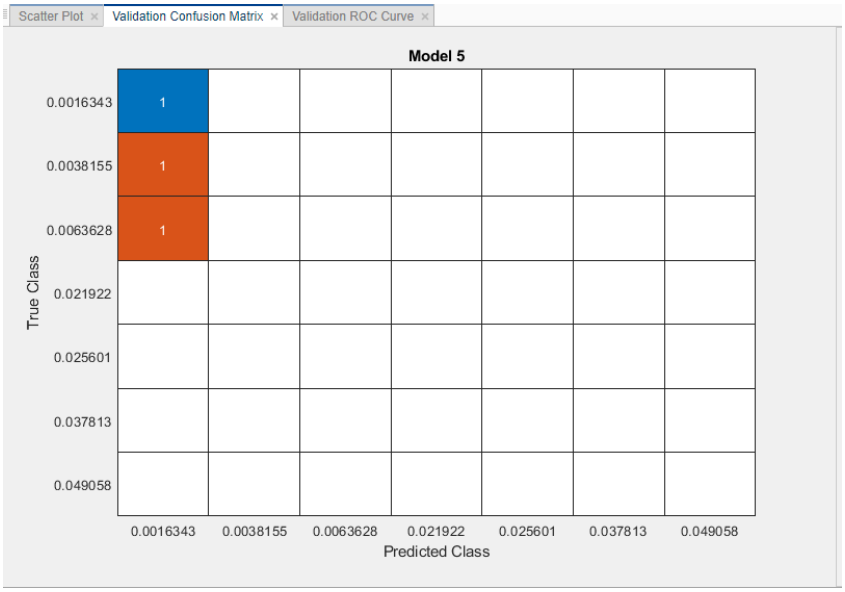
Fuente: Autor

En la matriz de confusión del modelo Métodos Gaussianos, se observa que solo un valor se encuentra en la diagonal de valores verdaderos y dos de los valores encontrados se

encuentran como falsos positivos, esto indica que los valores fueron clasificados de manera incorrecta en el diagrama.

Ilustración 15.

Matriz de confusión Modelo Métodos Gaussianos

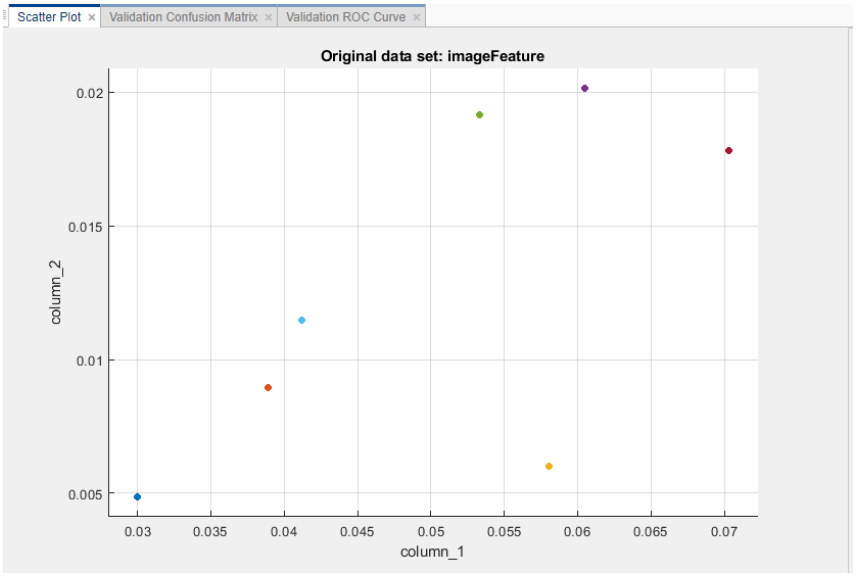


Fuente: Autor

El modelo de KNN (Vecinos más cercanos) muestra que algunos de los puntos se encuentran alineados, esto permitirá que algoritmo, entre las características de una mejor manera permitiendo observar un mayor porcentaje de validez.

Ilustración 16.

Modelo KNN

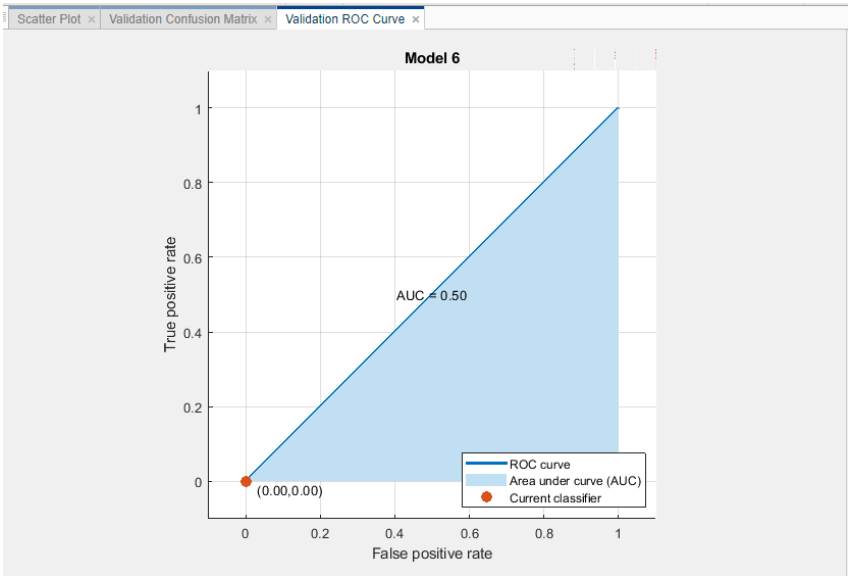


Fuente: Autor

En la curva de ROC de Modelo KNN, se observa que el porcentaje es de 0.50 lo que indica que el algoritmo requiere una mayor cantidad de imágenes para que el porcentaje de validación en el entrenamiento sea mejor.

Ilustración 17.

Curva ROC Modelo KNN

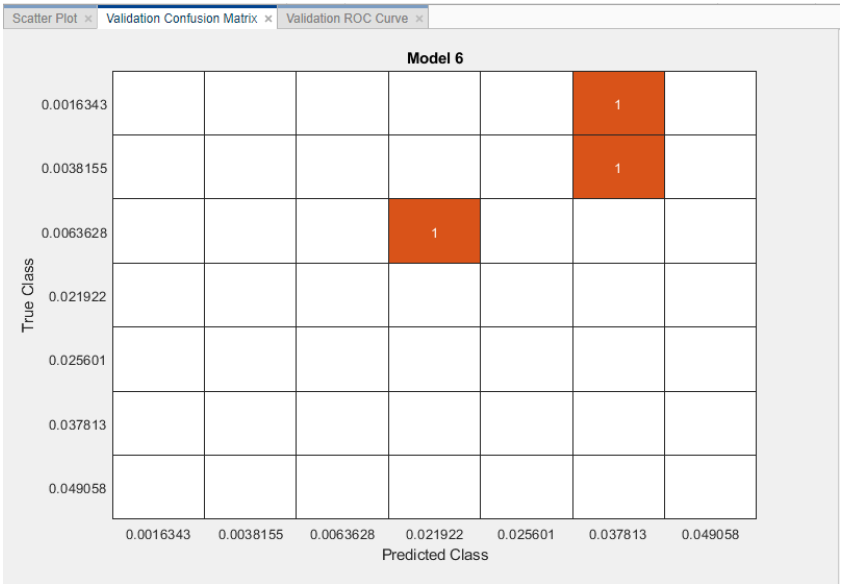


Fuente: Autor

En la matriz de confusión del Modelo KNN observamos que los valores son tomados como positivos, pero no se encuentran en la diagonal de verdaderos, lo que indicaque los valores fueron clasificados erróneamente por el algoritmo.

Ilustración 18.

Matriz de Confusión Modelo KNN



Fuente: Autor

Realizaremos un balance en los resultados de validación al entrenar los datos para cada uno de ellos (Tabla 3).

Tabla 3.

Porcentaje de validación de algoritmos para un conjunto de 7 imágenes

Tipo de Algoritmo	Porcentaje de Eficacia/Validación	Numero de características Imágenes
Árbol de decisiones	0.0%	500
Modelo Lineal SVM	33.3%	500
Teoría de Bayes	0.0%	500
Métodos Gaussianos	33.3%	500
Modelo KNN	0.0%	500

Fuente: Autor

Prueba para un conjunto de 100 imágenes

Realizaremos nuevamente la ejecución del programa, pero ahora con un conjunto de 100 imágenes, extraeremos los datos obtenidos (Tabla 4).

Tabla 4.

Datos generados para un conjunto de 100 imágenes

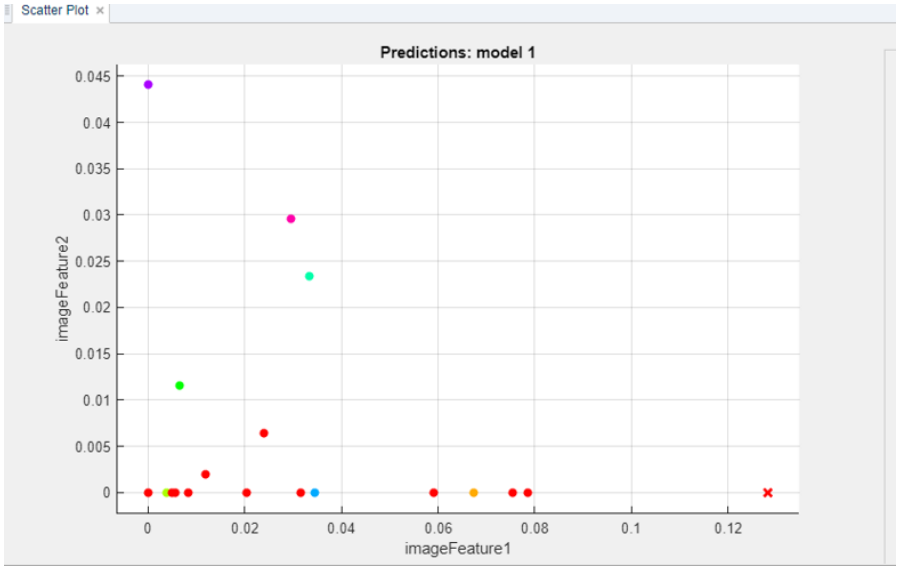
Datos imágenes de arandelas	
Total imágenes	100
Características extraídas	380560
Características válidas	4757500
Numero de clusters	500
Tiempo de procesamiento	2h

Fuente: Autor

Seleccionaremos el algoritmo de árbol de decisiones y entrenaremos el algoritmo observamos que, para el entrenamiento de 100 imágenes, se observan más características loque nos indicara un mayor porcentaje de efectividad.

Ilustración 19.

Modelo árbol de decisiones.

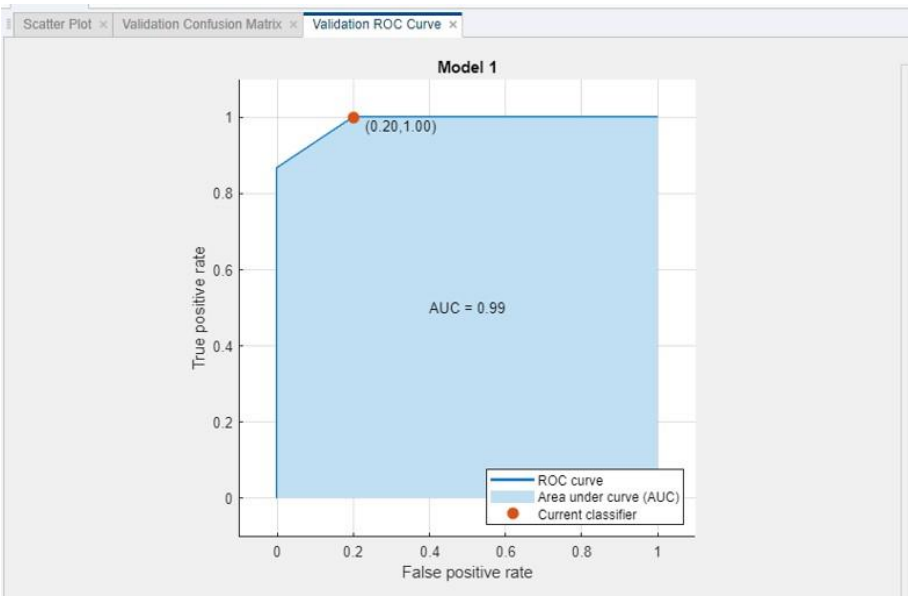


Fuente: Autor

En la curva de ROC del diagrama de árbol de decisiones se observa que el porcentaje es de 0.99, esto nos indica que la cantidad de datos suministrada para el algoritmo es alta y permite detectar una gran cantidad de características para el entrenamiento del algoritmo.

Ilustración 20.

Curva ROC árbol de decisiones

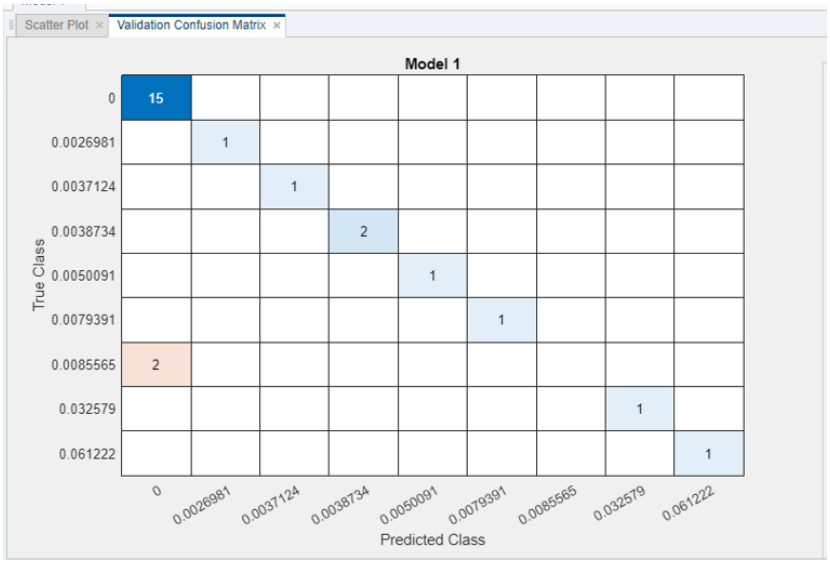


Fuente: Autor

En la matriz de confusión del diagrama de árboles decisiones se observa la diagonal de valores verdaderos, aunque uno de los valores es reconocido como falso positivo, el porcentaje de efectividad de la matriz es alta.

Ilustración 21.

Matriz de confusión árbol de decisiones

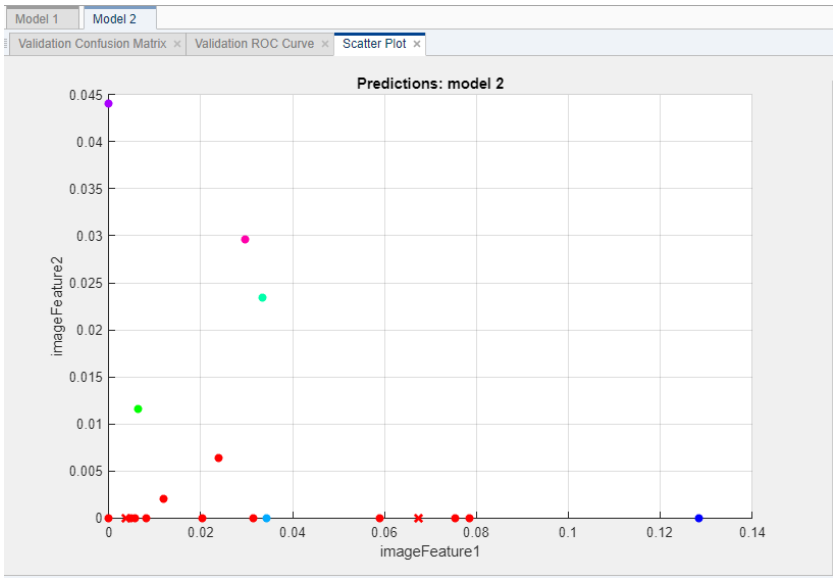


Fuente: Autor

En el modelo lineal SVM se observa que una gran cantidad de los puntos se encuentran en el eje x del diagrama, lo que indica que el entrenamiento realizará una aproximación muy exacta de las características encontradas.

Ilustración 22.

Modelo Lineal SVM

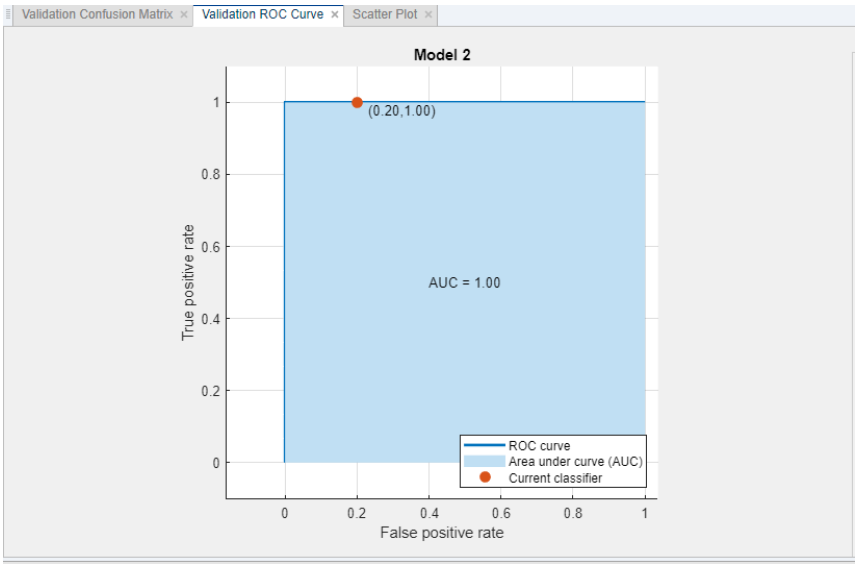


Fuente: Autor

La curva de ROC del Modelo Lineal SVM muestra el 1.00% de aproximación lo que indica que el algoritmo cumple con todos los parámetros extraídos de cada una de las características, este porcentaje debe ser menor para que se puedan analizar los datos entrenados en el algoritmo, no se tomara en cuenta este modelo.

Ilustración 23.

Curva ROC Modelo Lineal SVM

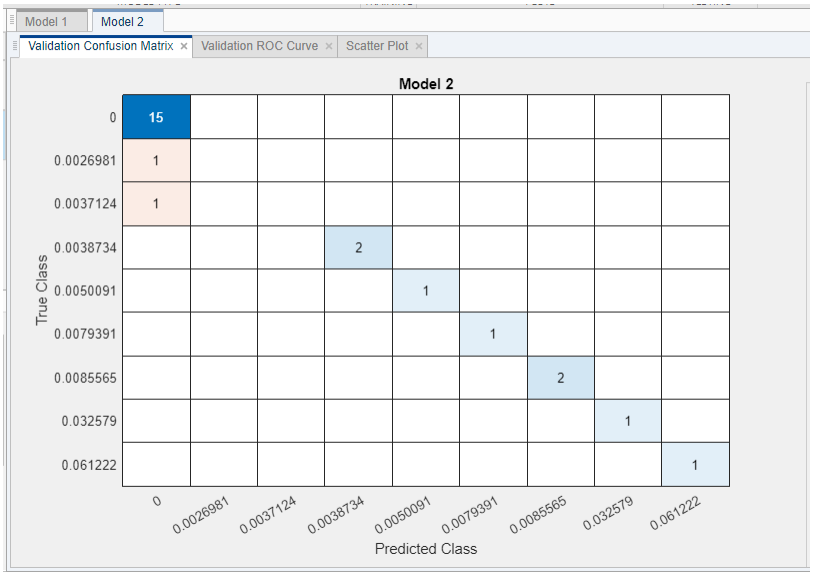


Fuente: Autor

La matriz de confusión del Modelo Lineal SVM, cuenta con la diagonal de valores verdaderos, aunque dos de los valores son tomados como falsos positivos dichos valores no fueron clasificados de manera correcta por el algoritmo.

Ilustración 24.

Matriz de confusión Modelo Lineal SVM

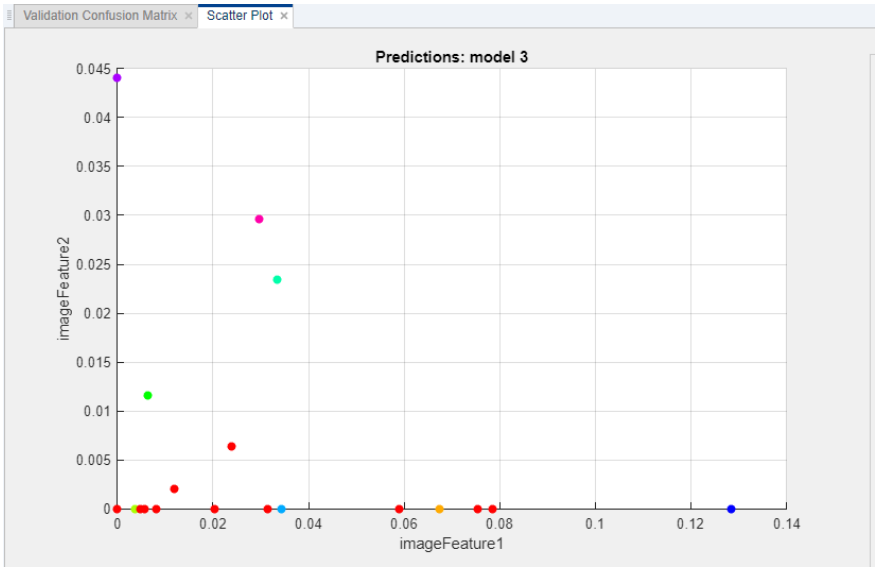


Fuente: Autor

En el Modelo Teoría de Bayes se observa que una gran cantidad de puntos tienden al eje x, para el entrenamiento de este algoritmo se espera que la validación sea muy aproximada, aunque algunos de los valores se encuentren dispersos.

Ilustración 25.

Modelo Teoría de Bayes

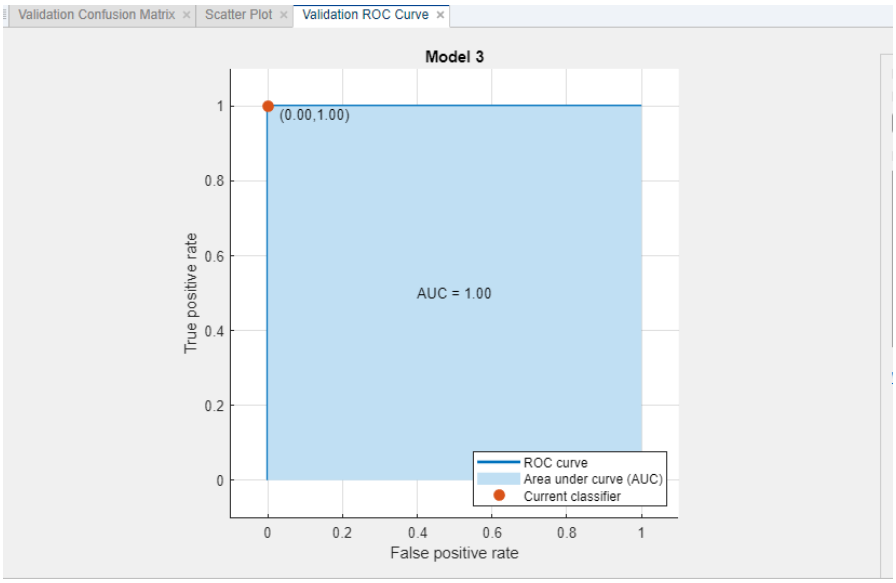


Fuente: Autor

En la curva de ROC del Modelo Teoría de Bayes se observa que la aproximación es del 1.00% lo que indica que la clasificación de características en el entrenamiento no tuvo ninguna variación, por lo que este modelo no será tomado en cuenta para la aplicación y clasificación de imágenes.

Ilustración 26.

Curva ROC Modelo Teoría de Bayes

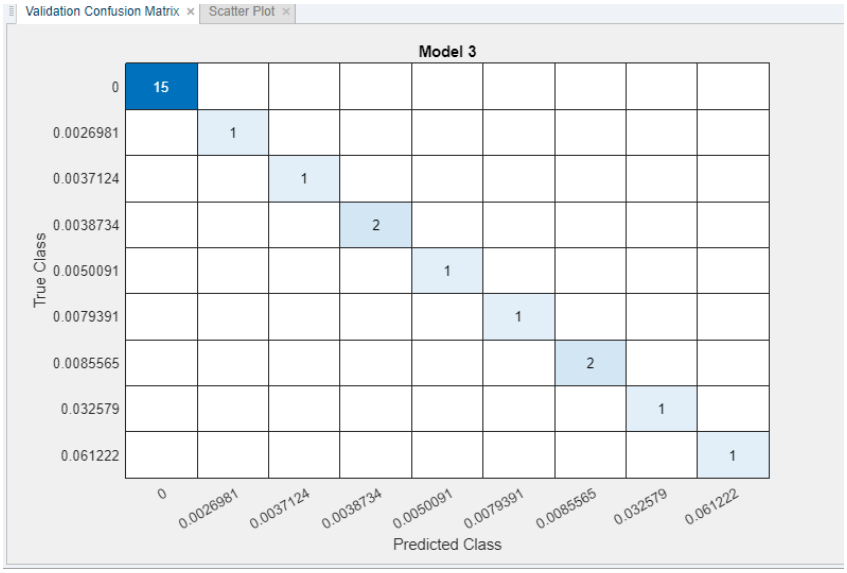


Fuente: Autor

En la matriz de confusión del Modelo de Teoría de Bayes se observa la diagonal de valores verdaderos, aunque este diagrama cuenta con todos los valores verdaderos a la hora de entrenar el algoritmo muestra una efectividad del 100%, este valor no puede ser utilizado porque se requiere una variación para los algoritmos de clasificación de objetos.

Ilustración 27.

Matriz de Confusión Modelo Teoría de Bayes

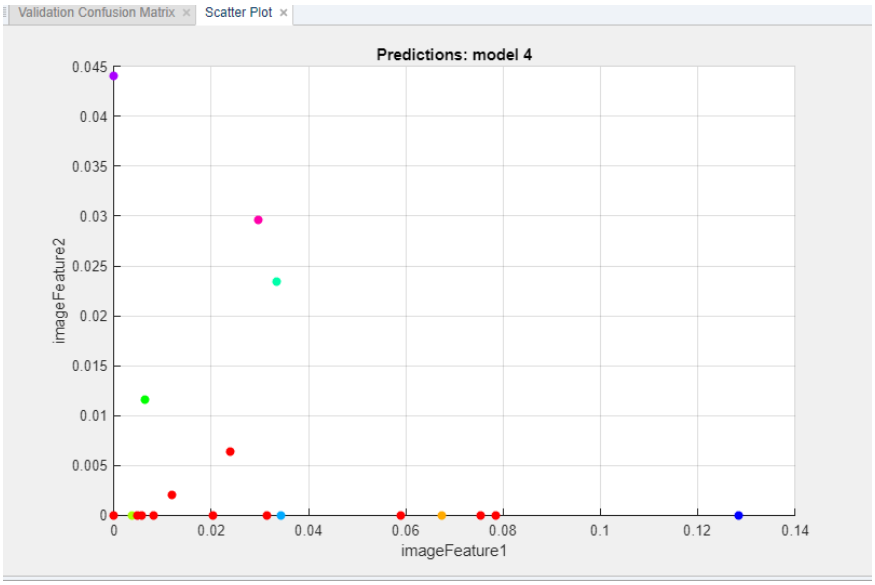


Fuente: Autor

En el Modelo Métodos Gaussianos se observa que gran cantidad de los datos se encuentran situados en el eje x, lo que permitirá tener una mayor aproximación y validez a la hora de entrenar el algoritmo.

Ilustración 28.

Modelo Métodos Gaussianos

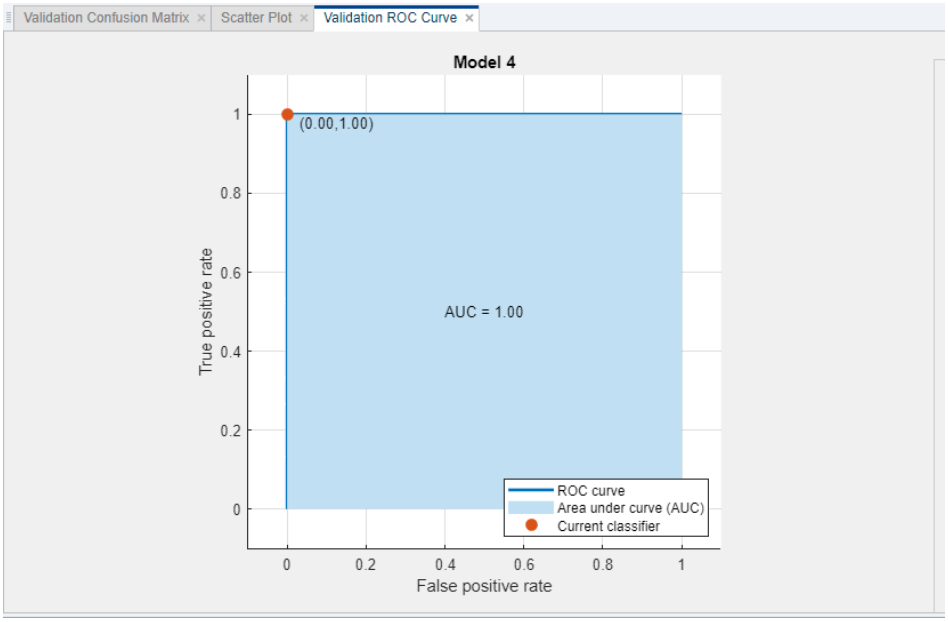


Fuente: Autor

En la curva de ROC del modelo Teoría de Bayes se observa que el porcentaje es 1.00%, lo cual nos indica que no ha encontrado ninguna inconsistencia al analizar los datos extraídos en las imágenes, dicho diagrama no muestra ninguna variación por lo que no será tomado en cuenta para el análisis del algoritmo de clasificación.

Ilustración 29.

Curva ROC Métodos Gaussianos

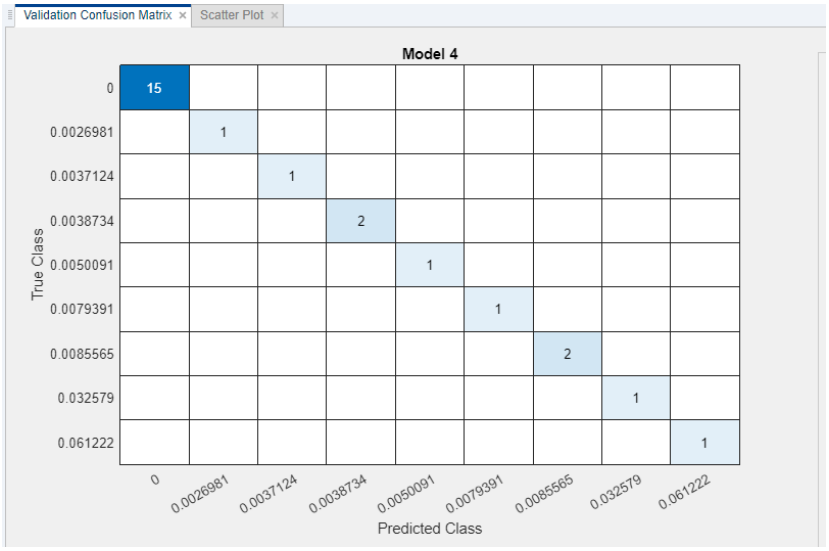


Fuente: Autor

La matriz de confusión del diagrama de Métodos Gaussianos muestra la diagonal de valores verdaderos, este diagrama cuenta con todos sus valores positivos pero la validez del entrenamiento equivale al 100%, lo que nos indica que no hay ninguna aproximación o margen de error por lo cual el diagrama no puede tomarse como en cuenta ya que se está realizando una aproximación.

Ilustración 30.

Matriz de confusión Métodos Gaussianos

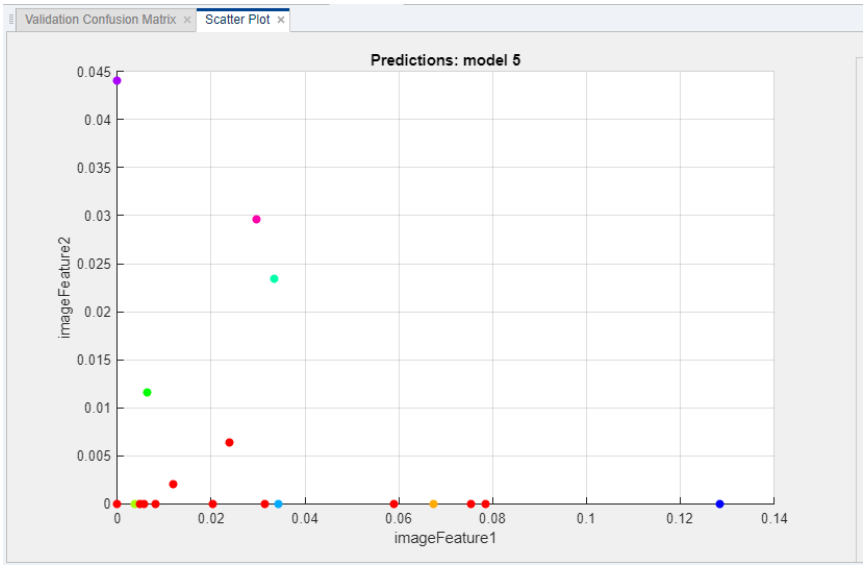


Fuente: Autor

En el modelo KNN (Vecinos más cercanos) se observa que una gran cantidad de valores se encuentran en el eje x, permitirá al algoritmo realizar una validación mucho más aproximada.

Ilustración 31.

Modelo KNN

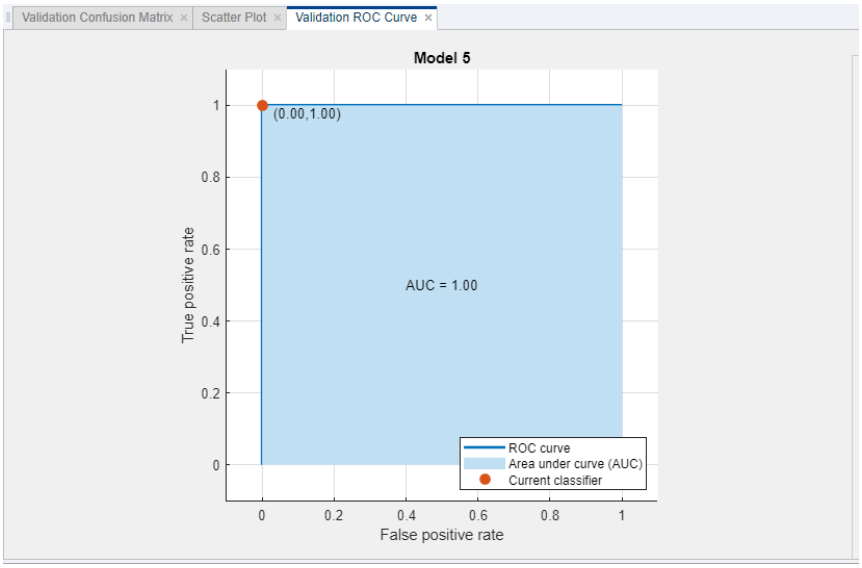


Fuente: Autor

En la curva de ROC del Modelo KNN observamos que el valor es igual 1.00%, lo cual nos indica que no se ha encontrado ninguna variación para el entrenamiento del algoritmo, por lo cual no se tomará el diagrama para el algoritmo de clasificación de imágenes.

Ilustración 32.

Curva ROC Modelo KNN

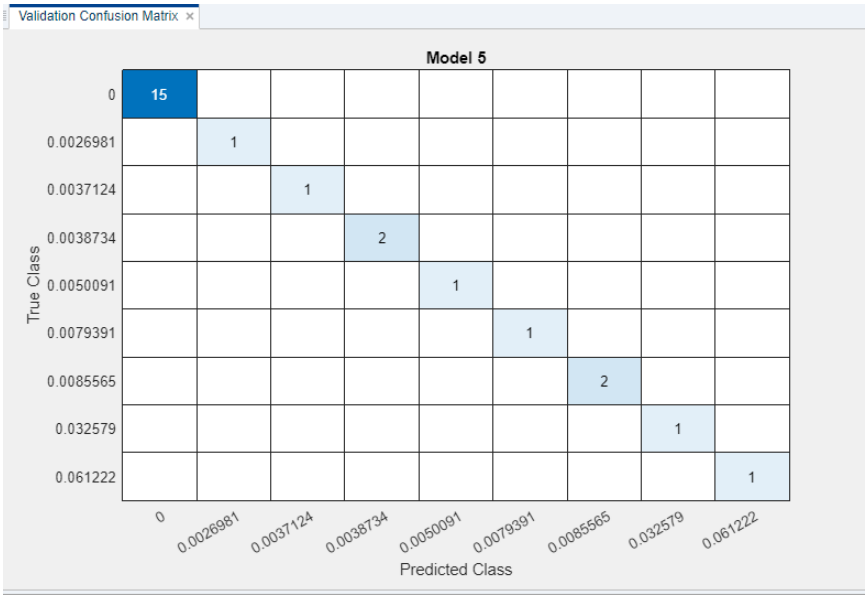


Fuente: Autor

En la matriz de confusión del Modelo KNN observamos la diagonal de valores verdaderos, al igual que los anteriores algoritmos se encuentra una validación del 100% a la hora de entrenarlo, por lo cual no puede realizarse una estadística de los valores que el no son correctos en el algoritmo.

Ilustración 33.

Matriz de confusión Modelo KNN



Fuente: Autor

Documentaremos los resultados de validación al entrenar los datos para cada uno de los algoritmos anteriores (Tabla 5).

Tabla 5.

Porcentaje de validación de algoritmos

Tipo de Algoritmo	Porcentaje de Eficacia/Validación	Numero de características Imágenes
Árbol de decisiones	92.2%	500
Modelo Lineal SVM	92.2%	500
Teoría de Bayes	100.0%	500
Métodos Gaussianos	100.0%	500
Modelo KNN	100.0%	500

Fuente: Autor

Para el entrenamiento de algoritmo de Deep learning se utilizó el siguiente código, este algoritmo permite analizar objetos en tiempo real por medio de una cámara y una base de datos previamente entrenada:

```
1. clear all

2. camera = webcam;
3. base = alexnet;

4. while true
5. picture = camera.snapshot;
6. picture = imresize(picture, [227, 227]);

7. label = classify (base, picture);

8. image(picture);
9. title (char(label));
10. drawnow;
11. end
```

La primera línea de código limpia el entorno de Matlab.

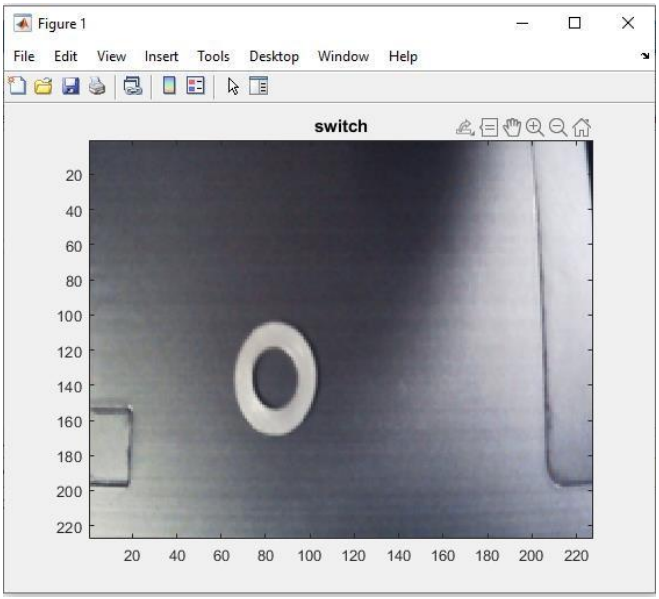
La segunda línea de código reconoce la cámara que esté conectada o integrada al equipo

La tercera línea de código se realiza un llamado a la base de datos previamente entrenada y que se encuentra en alexnet.

Desde la línea cuatro a la línea once se encuentra un ciclo while, el cual se ejecuta mientras este activa la cámara, posteriormente hace un reconocimiento a las imágenes que se encuentran en la base de datos de alexnet, finalmente clasifica el objeto detectado, el cual se observa en un título (label) en la parte superior de la imagen.

Ilustración 34.

Arandela detectada por medio de cámara



Fuente: Autor

Fase 3: Recomendación de algoritmos de machine learning y Deep learning

El algoritmo más apropiado para el entrenamiento de imágenes es de Deep learning ya que este trabaja con redes neuronales para el entrenamiento de imágenes.

Para trabajar con algoritmos de Machine learning y Deep learning se requiere de una computadora con una serie de características específicas a nivel de hardware y software. Es recomendable una computadora con al menos estas características de hardware: Procesador: Corei5; Memoria: 12 RAM; Disco de 500GB preferiblemente de estado sólido, tarjeta gráfica de 8GB RTX2070.

Para el entrenamiento se sugiere que sea realizado con una base de datos pública, tal como alexnet, *kaggle, ImageNetDogs, Caltech* 256, entre otras. Lo anterior, debido a que se requiere de una gran cantidad de imágenes para realizar el entrenamiento de algoritmo de machine learning.

En caso de realizar el entrenamiento de forma independiente, con imágenes personalizadas se recomienda agrupar las imágenes en carpetas para que el algoritmo pueda acceder a la ruta de manera mucho más rápida.

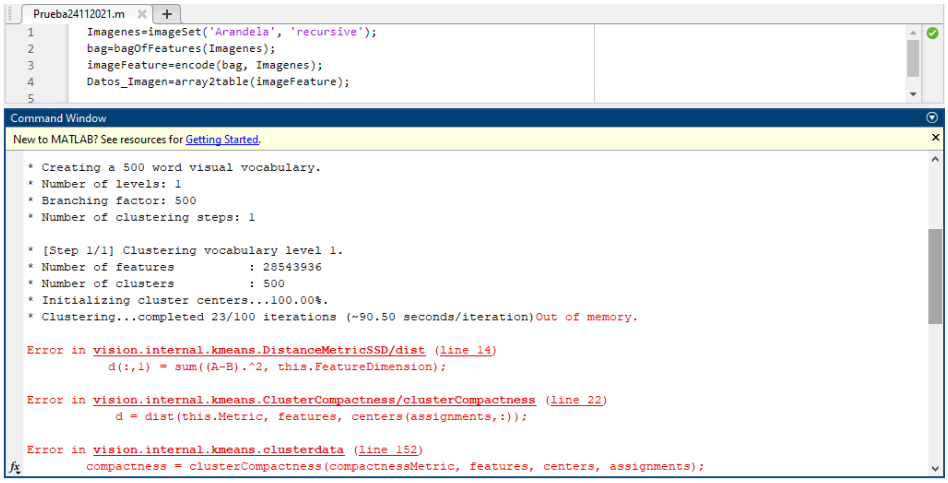
El algoritmo que se recomienda para el entrenamiento de imágenes es el de redes neuronales de *deep learning*, gracias a que cuenta con un tiempo menor a diferencia de los algoritmos de *machine learning*; el tiempo de ejecución es de30 min a diferencia de los algoritmos de machine learning, cuyo tiempo de ejecución es 1h 30m – 2h dependiendo tipo de imágenes y las características del equipo. La precisión del algoritmo es más elevada comparada con los algoritmos de machine learning; ya que este cuenta con una serie de capas que emplea para que la clasificación de imágenes sea más acertada.

Errores presentados

Al ejecutar el programa arroja un error al procesar las características, esto ocurre por la gran cantidad de elementos que el programa detecta y debe analizar, las características de hardware del equipo son un poco limitadas para la cantidad excesiva de datos que el programa debe analizar y clasificar; el programa arroja el error, "*Out Memory*".

Ilustración 35.

Algoritmo 200 Imágenes arandela error

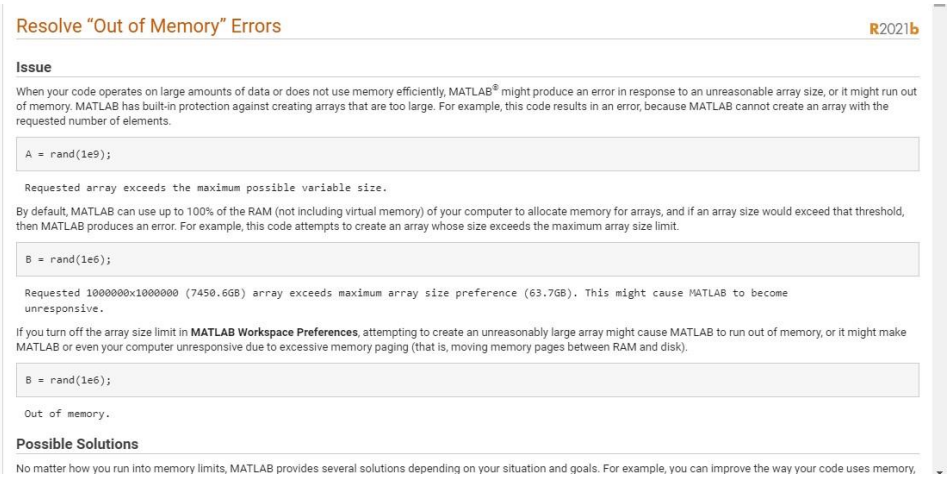


Fuente: Autor

La documentación del error presentado, se presenta por el exceso de memoria en las variables utilizadas, Matlab no puede crear el número de vectores que requiere cada una de las características extraídas ya que si se realizará dicho proceso este sobre pasaría los límites estipulados por el editor.

Ilustración 36.

Documentación error



Resolve “Out of Memory” Errors R2021b

Issue

When your code operates on large amounts of data or does not use memory efficiently, MATLAB® might produce an error in response to an unreasonable array size, or it might run out of memory. MATLAB has built-in protection against creating arrays that are too large. For example, this code results in an error, because MATLAB cannot create an array with the requested number of elements.

```
A = rand(1e9);
```

```
Requested array exceeds the maximum possible variable size.
```

By default, MATLAB can use up to 100% of the RAM (not including virtual memory) of your computer to allocate memory for arrays, and if an array size would exceed that threshold, then MATLAB produces an error. For example, this code attempts to create an array whose size exceeds the maximum array size limit.

```
B = rand(1e6);
```

```
Requested 1000000x1000000 (7450.6GB) array exceeds maximum array size preference (63.7GB). This might cause MATLAB to become
unresponsive.
```

If you turn off the array size limit in **MATLAB Workspace Preferences**, attempting to create an unreasonably large array might cause MATLAB to run out of memory, or it might make MATLAB or even your computer unresponsive due to excessive memory paging (that is, moving memory pages between RAM and disk).

```
B = rand(1e6);
```

```
Out of memory.
```

Possible Solutions

No matter how you run into memory limits, MATLAB provides several solutions depending on your situation and goals. For example, you can improve the way your code uses memory,

Fuente: Autor

Conclusiones

El módulo de aplicación *classification learner* permite realizar diferentes diagramas quemuestran el porcentaje de efectividad y emparejamiento respecto a cada una de las imágenes entrenadas.

El diagrama de Árbol y Diagrama múltiple presentan una efectividad menor comprada con el diagrama SVM (Maquina de vectores de soporte) ya que este ofrece una mayor precisiónen la clasificación de características.

Por otro lado, los diagramas que presentan una aproximación de eficacia del 92.2% son el diagrama árbol de decisiones y modelo lineal para el entrenamiento del conjunto de 100 imágenes, este valor puede variar dependiendo el número de características que extraigan;en base a los resultados obtenidos se puede evidenciar que estos algoritmos machine learning son los más apropiados para el entrenamiento de algoritmos de clasificación de imágenes.

El entrenamiento de algoritmos, permite agrupar una gran cantidad de características,extraídas de imágenes, con el fin de clasificar y analizar los datos de una manera másprecisa; este proceso es posible gracias al *machine learning y deep learning*.

Referencias

(Meneses & Marcelo, 2021) Aler Mur, R. (2015). Ricardo Aler Mur CLASIFICADORES KNN-I. *Universidad Carlos III de Madrid*.

Arturo, K., & Zavalza, R. (n.d.). *Robot inteligente recolector de basura asistido por redes neuronales artificiales.*

Basado, A., Learning, D., Asistentes, R., & Multiherramienta, P. (2018). *Robinson Jiménez Moreno*.

Betancourt, G. A. (2005). Las maquinas de soporte vectorial. *Scientia et Technica*, *27*(27), 67–72.

Carmona, E. (2016). *Abstract Support Vector Machine 1 Introducción*. *November*, 1–27. https://www.researchgate.net/publication/263817587_Tutorial_sobre_Maquinas_de_Vectores_Soporte_SVM

Conti, D., & Martinez, F. J. (2007). Reglas de Asociación en Series Temporales: panorama referencial y tendencias. *II Congreso Internacional de Informatica*, 215–221. https://www.academia.edu/8816240/Reglas_de_Asociación_en_Series_Temporales_panorama_referencial_y_tendencias

Gil, P., Pomares, J., Puente, S. T., Torres, F., Candelas, F., & Ortiz, F. G. (n.d.). *VISUAL : Herramienta para la Enseñanza Práctica de la Visión Artificial* .

Grado, T. F. De. (2019). *Manipulación de objetos mediante un brazo robótico usando*

técnicas de aprendizaje por refuerzo.

Gu, S., Holly, E., Lillicrap, T., & Levine, S. (2018). *Deep Reinforcement Learning for Robotic Manipulation with*. *July*.

Learning, D. R. (2017). *Olitécnica de*.

Matich, D. J. (2001). Redes Neuronales: Conceptos Básicos y Aplicaciones. *Historia*, 55. ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf

Mecánica, I. T. I., Un, D. D. E., & Vapor, C. D. E. (2012). *Escuela Politécnica Superior*.

Meneses, G., & Marcelo, D. (2021). *Diseño y desarrollo de un sistema de video vigilancia basado en dispositivos embebidos, técnicas de visión artificial y algoritmos inteligentes*.

Singh, A., Yang, L., Finn, C., & Levine, S. (2019). *End-To-End Robotic Reinforcement Learning without Reward Engineering*. https://doi.org/10.15607/rss.2019.xv.073

Tejada-Begazo, M. F. (2018). *Control Visual de un Brazo Manipulador con 7GDL, en Base a Visión Monocular, para el Seguimiento de Objetivos*.

V, B. C. A. R. (2009). *´ De T Ecnicas*.

Aler Mur, R. (2015). Ricardo Aler Mur CLASIFICADORES KNN-I. *Universidad Carlos III de Madrid*.

Arturo, K., & Zavalza, R. (n.d.). *Robot inteligente recolector de basura asistido por redes neuronales artificiales.*

Basado, A., Learning, D., Asistentes, R., & Multiherramienta, P. (2018). *Robinson Jiménez Moreno*.

Betancourt, G. A. (2005). Las máquinas de soporte vectorial. *Scientia et Technica*, *27*(27), 67–72.

Carmona, E. (2016). *Abstract Support Vector Machine 1 Introducción*. *November*, 1–27. https://www.researchgate.net/publication/263817587_Tutorial_sobre_Maquinas_de_Vectores_Soporte_SVM

Conti, D., & Martinez, F. J. (2007). Reglas de Asociación en Series Temporales: panorama referencial y tendencias. *II Congreso Internacional de Informatica*, 215–221. https://www.academia.edu/8816240/Reglas_de_Asociación_en_Series_Temporales_panorama_referencial_y_tendencias

Gil, P., Pomares, J., Puente, S. T., Torres, F., Candelas, F., & Ortiz, F. G. (n.d.). *VISUAL: Herramienta para la Enseñanza Práctica de la Visión Artificial.*

Grado, T. F. De. (2019). *Manipulación de objetos mediante un brazo robótico usando técnicas de aprendizaje por refuerzo*.

Gu, S., Holly, E., Lillicrap, T., & Levine, S. (2018). *Deep Reinforcement Learning for Robotic Manipulation with*. *July*.

Learning, D. R. (2017). *Olitécnica de*.

Matich, D. J. (2001). Redes Neuronales: Conceptos Básicos y Aplicaciones. *Historia*, 55. ftp://decsai.ugr.es/pub/usuarios/castro/Material-Redes-Neuronales/Libros/matich-redesneuronales.pdf

Mecánica, I. T. I., Un, D. D. E., & Vapor, C. D. E. (2012). *Escuela Politécnica Superior*.

Meneses, G., & Marcelo, D. (2021). *Diseño y desarrollo de un sistema de video vigilancia basado en dispositivos embebidos, técnicas de visión artificial y algoritmos inteligentes*.

Singh, A., Yang, L., Finn, C., & Levine, S. (2019). *End-To-End Robotic Reinforcement Learning without Reward Engineering*. https://doi.org/10.15607/rss.2019.xv.073

Tejada-Begazo, M. F. (2018). *Control Visual de un Brazo Manipulador con 7GDL, en Base a Visión Monocular, para el Seguimiento de Objetivos*.

V, B. C. A. R. (2009). *´ De T Ecnicas*.