Comparación de Clasificadores Basados en Árboles

Machine Learning

4 de septiembre de 2024

Cervantes Martínez Rodrigo Fabián

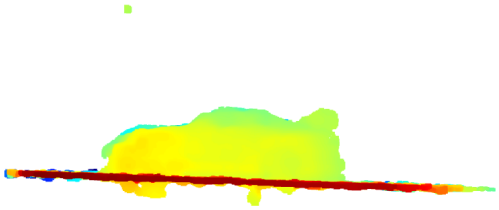
Campus Irapuato-Salamanca | División de Ingenierías

Resumen

En este trabajo se presenta la comparación entre diferentes clasificadores basados en árboles aplicados a un dataset de nubes de puntos capturadas con una cámara Intel RealSense D435. Se capturaron 200 ejemplos de nubes de puntos para cada uno de los 6 objetos diferentes. Las nubes de puntos fueron procesadas aplicando la norma L2 para el aislamiento de los objetos en el plano XYZ. Se calcularon vectores de características utilizando diferentes tamaños de puntos. Se construyó el dataset para finalmente implementar los modelos de clasificación así cómo su evaluación mediante el uso de distintas métricas.

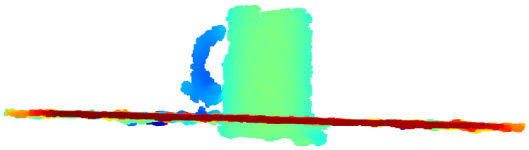
1. Descripción del Dataset

Para este estudio, se utilizó una camara Intel RealSense D415 para capturar un total de 1200 nubes de puntos, distribuidas equitativamente entre 6 categorías de objetos: un carro de juguete, un ordenador de monedas, una taza, un rollo de filamento, unos audífonos, y una figura del personaje Pikachu. Cada objeto fue capturado desde diferentes ángulos para obtener una mayor variedad en los datos. (Véase 1)

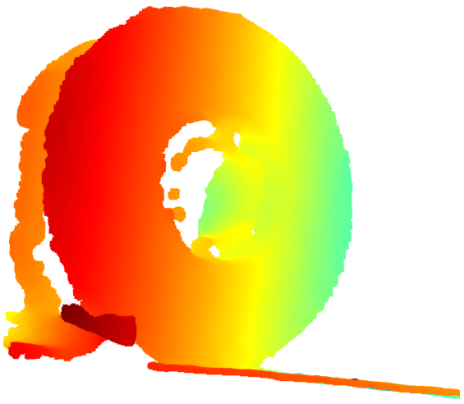


(a) Carro de jueguete

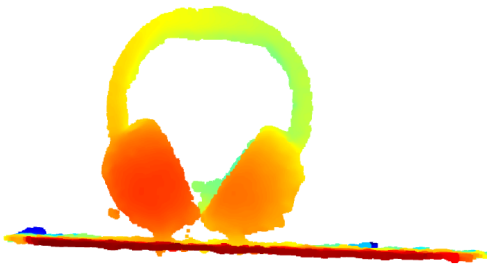
(b) Ordenador de monedas



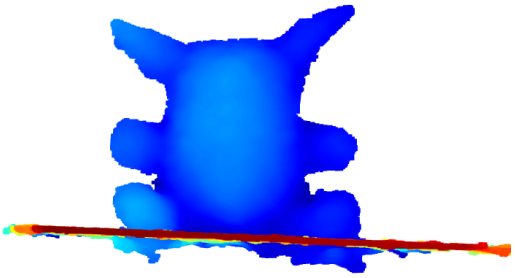
(c) Taza



(d) Rollo de filamento



(e) Audífonos



(f) Figura de Pikachu

Figura 1: Ejemplos de nubes de puntos de todas las clases capturadas.

2. Preparación de Datos y Extracción de Características

Se utilizó la cámara Intel RealSense D435 para capturar las nubes de puntos. Los datos fueron preprocesados aplicando un filtro de distancia basado en la norma L2, para aislar cada objeto de su fondo. Posteriormente, se extrajeron conjuntos de [5k, 10k, 15k y 2k] puntos aleatorios para cada nube de puntos, teniendo así, 800 instancias por cada clase.

Las siguientes características fueron calculadas para cada instancia:

- **Concavidad**: Diferencia entre el volumen de la nube de puntos y el volumen de la envolvente convexa que los encierra, representando qué tan cóncava es la forma del objeto.
- **Curvatura**: Mide el cambio de la tangente a la curva en una superficie a partir de los puntos Z, indicando cómo se curva la superficie del objeto.
- **Densidad de las Normales**: Variación en las direcciones de las normales calculadas en la nube de puntos, lo cual refleja la complejidad de la superficie.
- **Volumen de la Envolvente Convexa**: Volumen calculado de la envolvente convexa que cubre todos los puntos de la nube, útil para estimar el espacio tridimensional mínimo que contiene el objeto.
- **Área de Superficie Total**: Área total de la superficie de la envolvente convexa de la nube de puntos, proporcionando una medida de la extensión superficial del objeto.
- **Radio de Curvatura Medio**: Promedio del radio de curvatura en las coordenadas Z, lo cual indica la suavidad o aspereza de la superficie.
- **Desviación Estándar de la Altura (Z)**: Variabilidad en las coordenadas Z, ofreciendo una medida de la dispersión de los puntos en la altura.
- **Compacticidad**: Relación entre el volumen de la envolvente convexa y el área de superficie total, que indica cuán compacto es el objeto.
- **Desviación Estándar de las Distancias al Centroide**: Mide la dispersión de los puntos respecto al centroide de la nube de puntos, reflejando la uniformidad o dispersión en la distribución de puntos.
- **Asimetría Promedio (Skewness)**: Promedio de la asimetría en las distribuciones de las coordenadas X, Y y Z, para describir la simetría o sesgo de la forma del objeto.

3. Clasificadores Entrenados y Parámetros Utilizados

Se emplearon varios clasificadores basados en árboles, incluyendo el árbol de decisión, Random Forest, AdaBoost, Gradiente Boosting y el clasificador Extra Trees. Los parámetros fueron seleccionados mediante validación cruzada y son los siguientes:

Clasificador	**Parámetros**
DecisionTreeClassifier	`max_depth=5`, `min_samples_split=4`, `criterion='gini'`
RandomForestClassifier	`n_estimators=100`, `max_depth=10`, `min_samples_split=5`, `criterion='entropy'`
AdaBoostClassifier	`n_estimators=50`, `learning_rate=1.0`, `algorithm='SAMME'`
GradientBoostingClassifier	`n_estimators=100`, `learning_rate=0.1`, `max_depth=3`, `min_samples_split=4`
ExtraTreesClassifier	`n_estimators=100`, `max_depth=None`, `min_samples_split=2`, `criterion='gini'`

Cuadro 1: Valores de los parámetros para cada clasificador

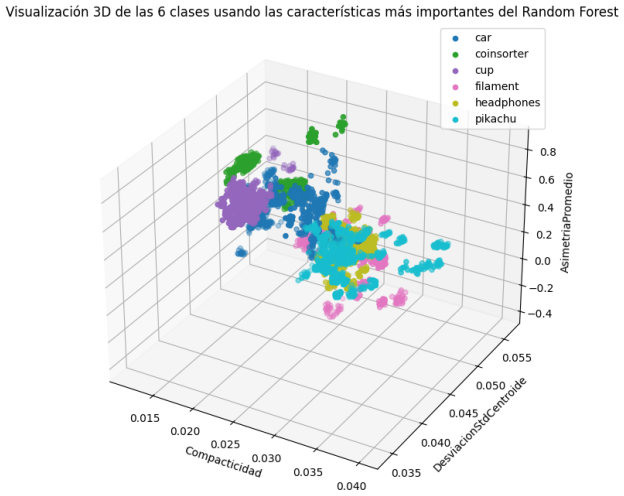


Figura 2: Visualización de clases

4. Resultados

La figura 2 muestra las diferentes clases del conjunto de datos en 3 dimensiones, usando como ejes las 3 características más importantes para RandomForest, esto con la finalidad de representar de una manera gráfica el comportamiento de los datos. Cabe aclarar que el cálculo de importancia de características sólo fue aplicado en esta representación, los clasificadores fueron entrenados con las 10 características calculadas inicialmente. También se identifica que el conjunto de datos NO es linealmente separable, lo que implica un aumento en la dimensionalidad del problema.

Los clasificadores fueron evaluados utilizando la precisión, recall y F1-score. Los resultados se muestran en la Tabla 2. Además, se reportaron los tiempos de entrenamiento y validación para cada clasificador:

Clasificador	**Precisión**	**Recall**	**F1-score**	**Tiempo de Entrenamiento (s)**
Decision Tree	0.92	0.92	0.92	0.03
Random Forest	1.00	1.00	1.00	0.68
Ada Boost	0.62	0.61	0.57	0.31
Gradiente Boosting	0.99	0.99	0.99	5.33
Extra Trees	1.00	1.00	1.00	0.13

Cuadro 2: Resultados de los clasificadores entrenados.

5. Conclusiones

En este estudio se compararon cinco clasificadores basados en árboles aplicados a un conjunto de datos de nubes de puntos. Los resultados revelan una notable variabilidad en precisión y tiempos de entrenamiento.

El Random Forest y el Extra Trees sobresalieron por su alta precisión y tiempos de entrenamiento relativamente bajos. En contraste, el Gradient Boosting mostró alta precisión, pero requirió un tiempo de entrenamiento significativamente mayor. Por otro lado el AdaBoostClassifier presentó la menor precisión y, aunque su tiempo de entrenamiento fue corto, su desempeño general fue inferior.

En resumen, el Random Forest y el Extra Trees ofrecen el mejor equilibrio entre precisión y eficiencia temporal, siendo las opciones preferidas para aplicaciones que requieren alta precisión sin comprometer el tiempo de ejecución.