Trabajo Final Inteligencia Artificial I
Año 2023: Visión Artificial y Reconocimiento de Voz

Dalessandro, Francisco

28 de febrero de 2024

Índice

Resumen

Este trabajo aborda el desafío de desarrollar un agente de inteligencia artificial para la identificación de frutas en imágenes y grabaciones de voz asociadas. Principalmente, se emplearon algoritmos de clasificación, como K-Nearest Neighbors (KNN) para el reconocimiento de voz y K-Means para la clasificación de imágenes, además de diferentes técnicas de separacion de datos, entre las cuales se encuentran el Zero Crossing Rate (ZCR), analisis de espectro y Coeficientes Cepstrales en las Frecuencias de Mel (MFCC) para el audio; y filtros, búsqueda de contornos y máscaras para las imagenes. Los resultados obtenidos fueron altamente satisfactorios, destacándose la precisión en la clasificación de audios. Las conclusiones reflejan la eficacia de la combinación de Visión Artificial y Reconocimiento de Voz, destacando su potencial en aplicaciones comerciales y domésticas.

1. Introducción

La Visión Artificial es un campo de estudio que busca dotar a las máquinas con la capacidad de interpretar y entender visualmente el mundo que las rodea. En paralelo, el Reconocimiento de Voz permite a los sistemas entender y procesar el lenguaje hablado. Este trabajo combina ambas disciplinas para abordar el desafío de identificar frutas en estantes utilizando técnicas de Visión Artificial y Reconocimiento de Voz.

El problema a resolver consiste en desarrollar un agente inteligente capaz de reconocer frutas en imágenes mediante técnicas de Visión Artificial y, simultáneamente, interpretar y clasificar grabaciones de voz asociadas a dichas frutas. El agente debe proporcionar información precisa sobre la ubicación de las frutas en estantes específicos, ofreciendo una experiencia interactiva y multisensorial para el usuario.

Esta combinación de tecnologías ofrece un enfoque innovador para mejorar la eficiencia y accesibilidad en la identificación de productos en entornos comerciales y domésticos.

2. Especificación del Agente

2.1. Tipo de agente: Agente que aprende

Para abordar el desafío de reconocer frutas en imágenes y grabaciones de voz, se diseñó un agente que aprende. Este enfoque implica la utilización de modelos internos entrenados durante el proceso de reconocimiento. Específicamente, el agente emplea algoritmos KNN y KMeans durante el entrenamiento para construir modelos que capturan características y patrones en los datos de imágenes y audio. Ese entrenamiento consistió en alimentarlo con valores discretos elegidos a mano, en lugar de modelos matemáticos o funciones.

Durante el reconocimiento, estos modelos internos actúan como un estado interno del agente, permitiendo la interpretación de información no directamente observable en el entorno, como podrían ser los momentos de Hu para una imagen, o los coeficientes de Mel para un audio. Los modelos proporcionan una representación comprensible del mundo, lo que facilita la toma de decisiones del agente respecto a la clasificación de frutas.

Los modelos absorben las características más representativas de las frutas, y de la forma de pronunciar sus nombres, y en base a dichas características, determina cómo clasificar nuevas muestras de audio e imagen, con el fin de identificar dónde está la fruta deseada. El agente puede recibir "feedback"de una forma indirecta. Cuando el algoritmo devuelve valores que no son lógicos, el autor puede ajustar parámetros para cambiarlos. Es como una "crítica" que recibe el programa.

Aunque luego el agente deje de aprender, respondiendo únicamente al usuario qué tipo de fruta está viendo, sigue siendo clasificado como agente que aprende.

2.2. Propiedades del entorno de trabajo

El entorno presenta las siguientes características:

- **Totalmente observable**: El agente tiene información de lo que hay en los 4 estantes al mismo tiempo. Además de que también escucha sonidos en espera de alguna palabra, y es toda la información que necesita para actuar.

- **Determinista**: Cuando el agente escucha un sonido o ve una imagen y la procesa, obtiene resultados numéricos. Si escucha exactamente el mismo audio y ve exactamente la misma imagen, los resultados seran idénticos a los anteriores. Seguirá así aunque se procese la misma entrada 100 veces.

- **Episódico**: Los audios e imagenes sucesivas no afectan al razonamiento de una entrada anterior. Cuando el programa está en funcionamiento, el agente NO aprende de las nuevas entradas, tan solo las compara con un modelo ya creado.

- **Estático**: Actúa con las imagenes que se le da, y calcula resultados una sola vez. Es decir, determina al inicio de la ejecución qué frutas hay y en qué estantes están. Desde el punto de vista del agente, eso es un entorno estático pues si luego se cambian de lugar, no se dará cuenta.

- **Discreto**: Existen tantas acciones y percepciones como frutas. El agente no conoce más allá de lo que es una pera, una banana, una manzana y una naranja.

- **Monoagente**: Sólo es necesario un agente que haga ambas categorizaciones.

2.3. Tabla REAS

Criterio	**Descripción**
Rendimiento	Tiempo de proceso entre que se dice una palabra y se muestra la imagen. El tiempo es bastante corto pese a la cantidad de análisis que conlleva, salvo la primer palabra despues de iniciado el programa, por motivos desconocidos. Esta rapidez se debe a que el modelo Knn y Kmeans están guardados en archivos aparte, para que el agente no tenga que volver a calcular los puntos cada vez que se ejecuta el programa. Porcentaje de aciertos.
Entorno	Estantes y sonidos de ambiente
Sensores	Cámara fotográfica del celular. Micrófono de la laptop. Teclado para interacción con el usuario.
Actuadores	Cálculos matemáticos para la extracción de características. Presentación de resultados al usuario por pantalla.

Cuadro 1: Tabla REAS para el agente de reconocimiento de frutas.

3. Diseño del Agente

3.1. Algoritmos Utilizados

3.1.1. K-Means

K-Means es un algoritmo de clustering no supervisado que agrupa datos en k grupos basándose en similitudes. En el contexto de este agente, se aplicó K-Means a las imágenes de frutas para agrupar píxeles similares. El número de grupos (k) se fijó en 4, representando las categorías de frutas: manzana, naranja, banana y pera.

El algoritmo K-Means funciona de la siguiente manera:

- **Inicialización:** Se seleccionan aleatoriamente k centroides iniciales.
- **Asignación:** Cada imagen se asigna al centroide más cercano.
- **Actualización:** Se recalculan los centroides como la media de las imágenes asignadas a cada cluster.
- **Iteración:** Se repiten los pasos de asignación y actualización hasta que los centroides convergen.

Después de la aplicación de K-Means, los centroides resultantes se etiquetaron manualmente para asignar las categorías correspondientes.

3.1.2. K-Nearest Neighbors (KNN)

KNN es un algoritmo supervisado utilizado para clasificación y reconocimiento de patrones. Funciona asignando una etiqueta a un punto de datos basándose en las etiquetas de los puntos vecinos más cercanos en el espacio de características. En este agente, KNN se entrenó con un conjunto de datos de grabaciones de voz etiquetadas con categorías de frutas. Cuando se presenta una nueva grabación, KNN encuentra los k vecinos más cercanos y asigna la etiqueta más frecuente.

La elección de KNN se basa en su capacidad para adaptarse a nuevos datos y en la simplicidad de su implementación en tareas de clasificación de patrones.

Ambos algoritmos, K-Means y KNN, se integran sinérgicamente para proporcionar una solución completa y robusta para el reconocimiento de frutas tanto en imágenes como en grabaciones de voz.

3.2. Librerias utilizadas

3.2.1. OpenCV (cv2)

OpenCV es una biblioteca de visión artificial de código abierto. En este proyecto, se utiliza para el procesamiento de imágenes, incluyendo la segmentación y clasificación de frutas en las imágenes de los estantes. Fue esencial para el cálculo de los momentos de Hu, así como los distintos tipos de filtros aplicados

3.2.2. NumPy (np)

NumPy es una biblioteca fundamental para la computación científica en Python. Se utiliza para realizar operaciones numéricas eficientes en matrices, especialmente en el contexto de procesamiento de imágenes, aunque también fue utilizado para el audio.

3.2.3. Librosa

Librosa es una biblioteca especializada en análisis de audio y música. En este proyecto, se utiliza para extraer características relevantes de las grabaciones de voz, como el Zero Crossing Rate (ZCR), el análisis de espectro y los coeficientes de Mel.

3.2.4. Matplotlib

Matplotlib es una biblioteca para la creación de visualizaciones estáticas, animadas e interactivas en Python. Se utiliza para la representación gráfica de datos, como la visualización de las características extraídas del audio.

3.2.5. Tempfile

Tempfile es un módulo de Python para la creación de archivos temporales y directorios temporales. En este proyecto, se emplea para almacenar temporalmente las grabaciones de voz antes de procesarlas.

3.2.6. Sounddevice (sd)

Sounddevice es una biblioteca para la reproducción y grabación de audio en Python. Se utiliza para grabar las respuestas del usuario durante la interacción del agente.

3.2.7. Pickle

Pickle es un módulo de serialización en Python. Se utiliza para almacenar y cargar objetos serializados, como modelos de aprendizaje automático y datos, en archivos.

3.2.8. Joblib

Joblib es una biblioteca para la ejecución eficiente de trabajos en paralelo en Python. Se utiliza para guardar y cargar modelos de aprendizaje automático entrenados con scikit-learn, aunque también sirve para guardar los que hayan sido creados por los algoritmos escritos por el autor.

4. Código

El código fuente se puede encontrar en el siguiente repositorio en línea: `https://github.com/Panchardo/IA-final/tree/note3`

Los archivos más importantes se encuentran en la carpeta códigos. El archivo *main.py* es el que interactua con el usuario, recibiendo sonidos e imagenes y respondiendo. Los llamados *main_audio.py* y *main_imagen.py* son los utilizados para la creacion de los modelos. Es decir, son los programas de entrenamiento.

5. Ejemplo de Aplicación

El funcionamiento es sencillo: Hay 4 frutas distribuidas en 4 estanterias. El usuario dice una de las 4 palabras (naranja, banana, pera o manzana) y el agente debe decirle en qué estantería está y mostrar la imagen de la fruta.

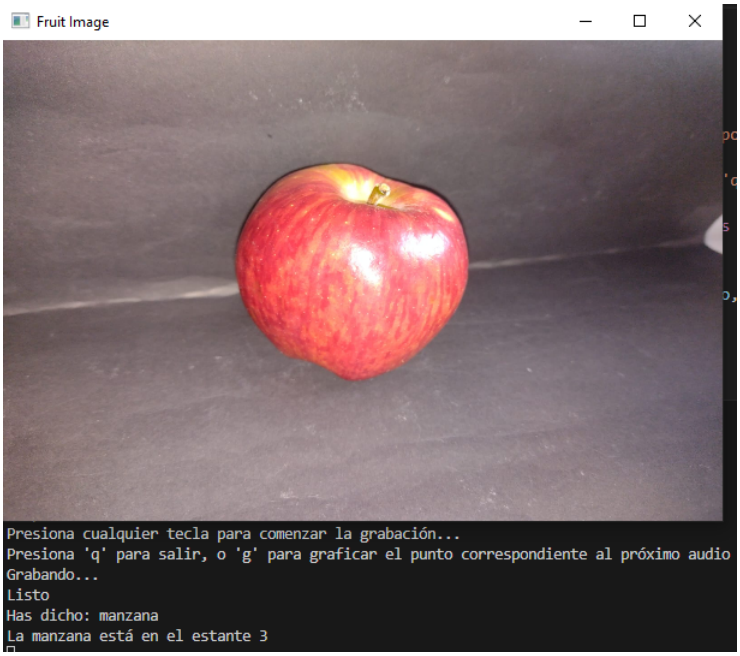

Figura 1: Manzana

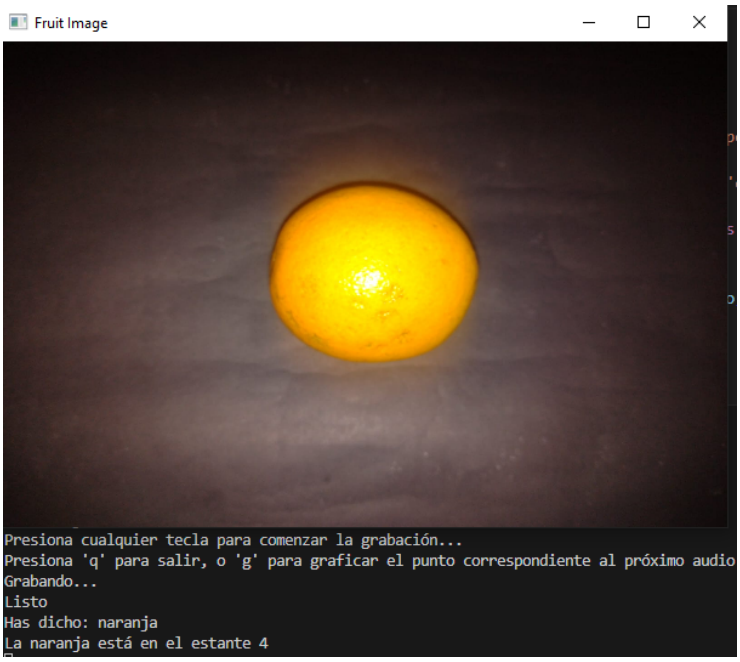

Figura 2: Naranja

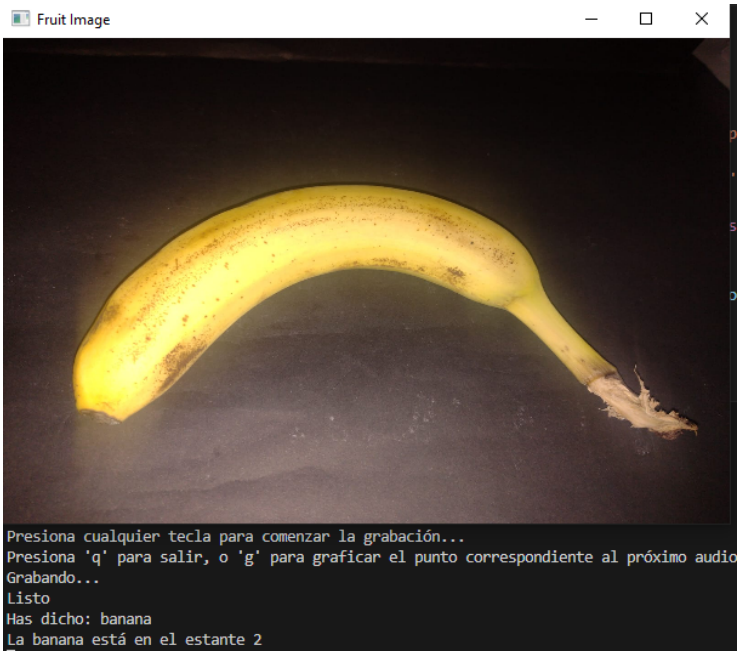

Figura 3: Banana

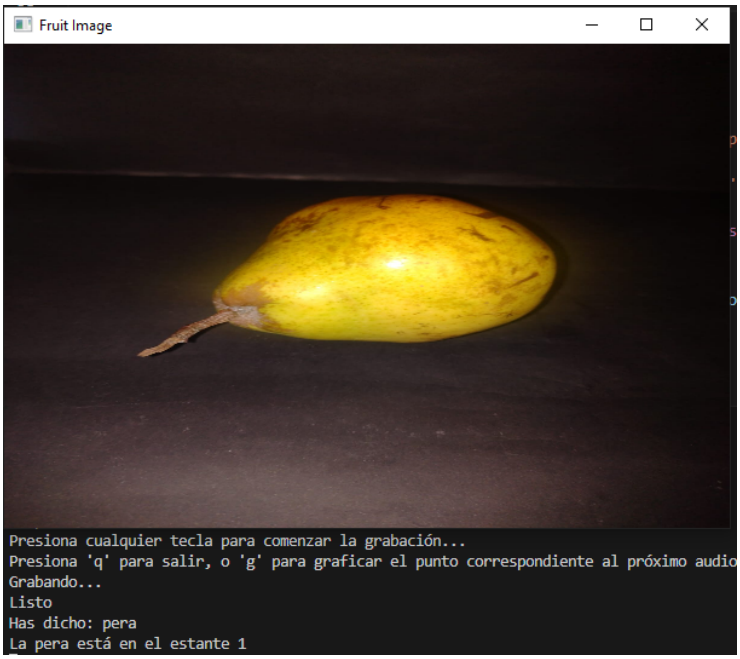

Figura 4: Pera

6. Implementaciones y resultados

6.1. Audio

Para la separación de los audios mencionando los nombres de las frutas, se ha hecho uso del algoritmo Knn. Para poder graficar los puntos en el espacio, era necesario encontrar números que sirvieran de coordenadas para dichos puntos. La idea está en buscar que dichos números sean representativos a las caracteristicas de las frutas y una vez encontrados, amplificarlos para que sean aún más representativos.

La primer componente contempla una operación entre Coeficientes Cepstrales en las Frecuencias de Mel. Los MFCC son características ampliamente utilizadas en el procesamiento de señales de audio y reconocimiento de voz. Representan la información espectral de un segmento de audio en función de las frecuencias mel, que están diseñadas para imitar las características de percepción auditiva humana.

En la Figura 5, aparecen representados 15 coeficientes de 4 audios aleatorios correspondientes a las 4 clases de fruta. Los coeficientes corresponden a las "filas"(se pueden ver como cuadrados o pixeles de colores) y cambian respecto al tiempo.

Luego de comparar gráficas, se puede apreciar que para la pera, los coeficientes 2 y 4 exhiben energías cercanas (los colores amarillos se acercan bastante),

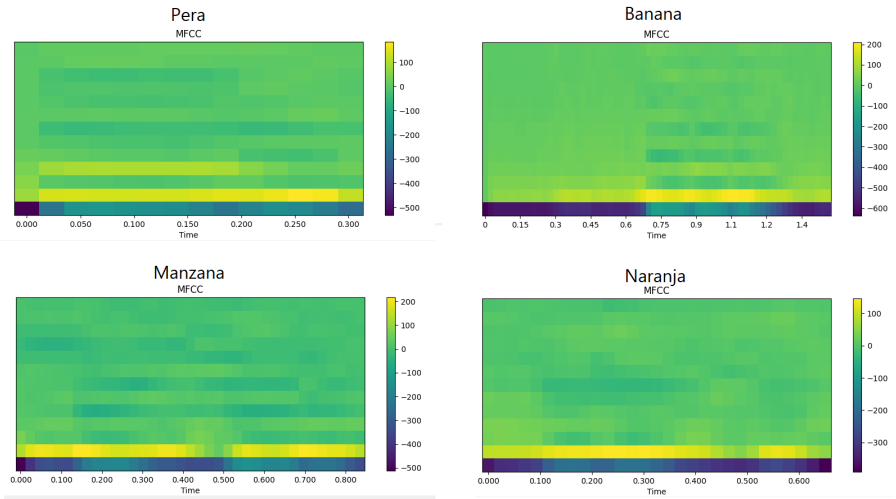

Figura 5: Coeficientes Cepstrales en las Frecuencias de Mel (MFCC)

una característica consistente en la mayoría de los audios de pera.

Entonces, lo que se hizo fue restar los máximos valores de los coeficientes 2 y 4 en la ventana de tiempo que dura el audio, y si está por debajo de cierto valor obtenido experimentalmente (segun cuadro 2, este valor podría ser alrededor de 105 o 110), entonces es una pera. Para asegurar que esta componente en las peras se distinga mucho de la misma en las demás frutas, se multiplicará por 10 (valor arbitrario) si está por debajo del umbral establecido. Se 'amplifica' la característica de la pera. Este enfoque asegura que las características específicas de las peras se destaquen claramente frente a otras frutas. La amplificación selectiva de características contribuye a la precisión del modelo de clasificación de frutas. Nótese la diferencia entre las restas de los coeficientes de Mel de una pera y los de una banana (cuadro 3)

Audio Name	Mel Coefficients Difference
pera1.wav	170.8408508
pera10.wav	150.7820129
pera11.wav	89.75494385
pera12.wav	89.75494385
pera13.wav	69.03981018
pera14.wav	99.70948792
pera15.wav	91.07131958
pera16.wav	115.5879745
pera17.wav	46.96472168
pera18.wav	64.56350708
pera19.wav	42.12239075
pera2.wav	172.6532288
pera20.wav	60.454422
pera21.wav	93.3765564

Cuadro 2: Diferencia de coeficientes Mel para archivos de audio de peras.

Audio Name	Mel Coefficients Difference
banana1.wav	189.3668518
banana10.wav	208.5338745
banana11.wav	142.1755524
banana12.wav	156.8311005
banana13.wav	155.9838715
banana14.wav	188.2622681
banana15.wav	166.7988586
banana16.wav	175.3930664
banana17.wav	132.8165741
banana18.wav	119.4607697
banana19.wav	128.9922485

Cuadro 3: Diferencia de coeficientes Mel para archivos de audio de bananas.

Aún queda por separar la banana, la manzana y la naranja. Para ello, se utilizó otro análisis: el Zero Crossing Rate (ZCR). El ZCR es una medida que indica la cantidad de veces que una señal cruza el eje horizontal (cero) en una unidad de tiempo. Es útil para caracterizar la frecuencia de cambios en una señal, como las transiciones entre sonidos positivos y negativos en una onda de audio.

En las Figuras 6 y 7, se aprecia que los valores de ZCR máximos de la pera y la banana son menores a los mismos de la manzana y la naranja. Entonces, se decidió establecer otro umbral, el cual si era superado (usualmente es el caso de las manzanas), se multiplicaría por 1000 el valor de ZCR, y si no llegaba a un valor minimo, se multiplicaria por -1000. De esta forma, las manzanas y

naranjas se alejan de las otras dos.

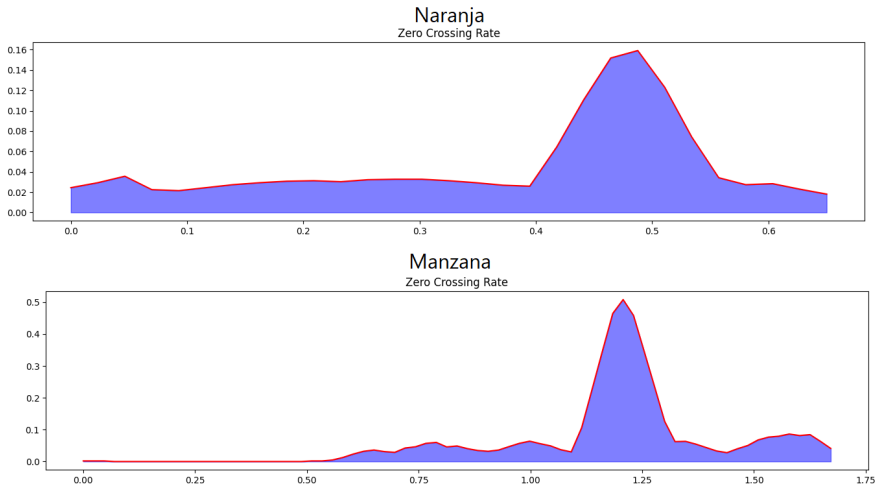

Figura 6: ZCR - Manzana y naranja.

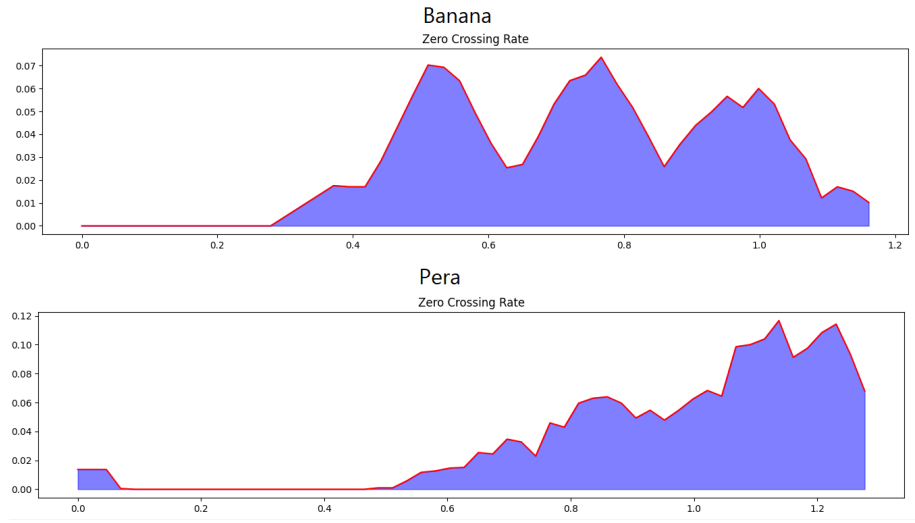

Figura 7: ZCR - Banana y pera.

Además, si ese máximo se encuentra entre los 3/5 y 5/5 de audio, hay una gran probabilidad de que sea una naranja. Por eso, para la tercer componente se analizó la posición del máximo y si yacía dentro de ese intervalo y superando cierto valor, que entonces el agente lo multiplique por otro número arbitrario, y

lo coloque en la tercer componente.

La clasificación se realizo con un valor de k = 7.

Nombre	Max ZCR
banana1.wav	0.069824219
banana10.wav	0.05859375
banana11.wav	0.060546875
banana12.wav	0.079101563
banana13.wav	0.059082031
banana14.wav	0.068359375
banana15.wav	0.073242188
banana16.wav	0.089355469
banana17.wav	0.054199219
banana18.wav	0.064453125
banana19.wav	0.083496094
banana2.wav	0.068847656

Cuadro 4: Valores máximos de ZCR para archivos de audio de bananas.

Nombre	Max ZCR
pera1.wav	0.067871094
pera10.wav	0.059570313
pera11.wav	0.074707031
pera12.wav	0.074707031
pera13.wav	0.056152344
pera14.wav	0.053222656
pera15.wav	0.055664063
pera16.wav	0.051757813
pera17.wav	0.10546875
pera18.wav	0.099121094
pera19.wav	0.078613281
pera2.wav	0.064453125

Cuadro 5: Valores máximos de ZCR para archivos de audio de peras.

Nombre	Max ZCR
manzana1.wav	0.445800781
manzana10.wav	0.282714844
manzana11.wav	0.217773438
manzana12.wav	0.217773438
manzana13.wav	0.233398438
manzana14.wav	0.182617188
manzana15.wav	0.205566406
manzana16.wav	0.187988281
manzana17.wav	0.436523438
manzana18.wav	0.466796875
manzana19.wav	0.37890625
manzana2.wav	0.362304688

Cuadro 6: Valores máximos de ZCR para archivos de audio de manzanas.

Nombre	Max ZCR
naranja1.wav	0.18359375
naranja10.wav	0.159667969
naranja11.wav	0.133300781
naranja12.wav	0.142578125
naranja13.wav	0.178222656
naranja14.wav	0.133789063
naranja15.wav	0.119628906
naranja16.wav	0.133300781
naranja17.wav	0.230957031
naranja18.wav	0.231445313
naranja19.wav	0.166992188
naranja2.wav	0.178710938

Cuadro 7: Valores máximos de ZCR para archivos de audio de naranjas.

Siguiendo lo establecido, al correr y plotear el modelo Knn, se verá como muestra la Figura 8. La estrella de la figura indica la palabra nueva pronunciada al correr el programa. Como se puede ver, el usuario pronunció "pera", y el algoritmo Knn lo etiquetará como tal al encontrarse cerca de otras peras. Cabe aclarar que los audios de banana se diferencian del resto al no cumplir con las caracteristicas antes mencionadas.

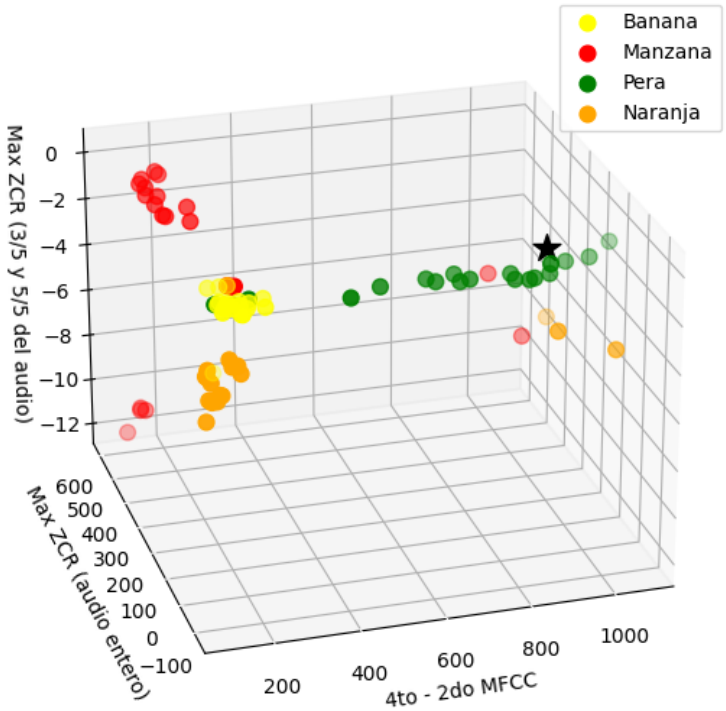

Figura 8: Modelo Knn con frutas etiquetadas

6.2. Imagen

Para el procesamiento de imágenes, se aplicaron diversas técnicas con el objetivo de preparar los datos para el algoritmo K-Means. Se emplearon operaciones como el filtro gaussiano, filtro negativo, búsqueda de contornos, filtrado de fondo negro, máscaras, escala de grises y momentos de Hu. La finalidad es similar a la del análisis de audio: manipular los vectores para que ocupen posiciones definidas y así facilitar la formación de clusters mediante K-Means.

El proceso comienza con la filtración del fondo negro de la imagen y su conversión a blanco. Esto se debe a que las fotos de la maqueta real tienen fondo negro, mientras que las de la base de datos utilizada para el entrenamiento presentan fondo blanco. A pesar de esta conversión, se aplica un filtro de escala de grises y posteriormente, un filtro gaussiano para reducir el ruido y facilitar la detección de bordes. A continuación, se invierte la imagen para que el fondo vuelva a ser negro y uniforme, permitiendo que OpenCV detecte los bordes, ya que no lo hace eficientemente en fondos blancos.

Una vez obtenido el contorno más grande, se calculan los momentos de Hu y se les aplica una transformación logarítmica. Además, se utilizan máscaras para filtrar la imagen por colores. Se tienen cuatro máscaras que manejan rangos de

colores característicos de cada fruta.

La primera componente del vector resultante refleja el porcentaje de píxeles detectados por cada máscara. Por ejemplo, si la máscara de manzana tiene el mayor porcentaje en comparación con las otras, indica que hay más píxeles rojos, sugiriendo que la imagen es probablemente una manzana. En el código, este porcentaje se multiplica por distintos valores dependiendo de la máscara correspondiente al máximo (cuadro 8). Se utilizaron máscaras con escala de colores HSV.

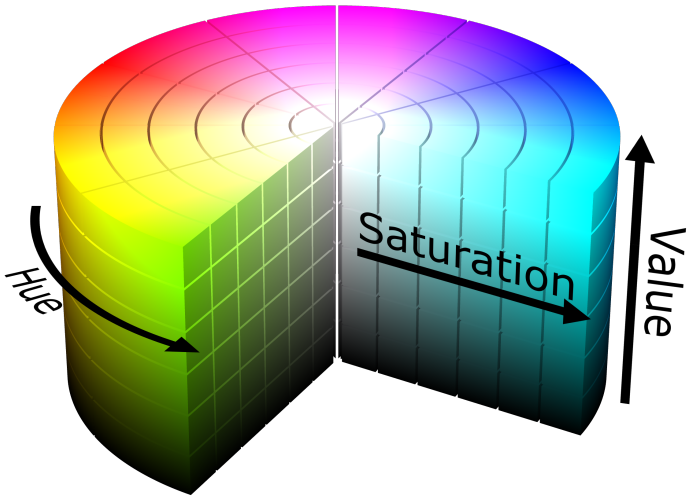

Figura 9: Cilindro de escala de colores HSV

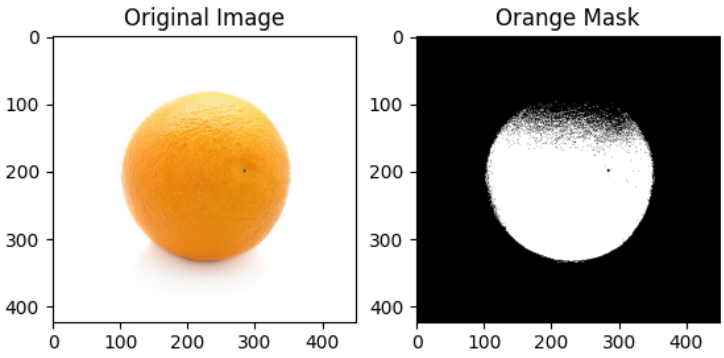

Figura 10: Ejemplo de aplicación de máscara.

Mascara predominante	Coeficiente
Banana	2
Pera	1
Manzana	-50
Naranja	4

Cuadro 8: Coeficientes para mascaras predominantes.

La segunda componente refleja el segundo momento de Hu. Este da información sobre la forma de la fruta y además es invariante a la rotación. Según el cuadro 9, para las bananas este momento no supera los 1.75. Entonces, si no supera ese valor, se multiplica por -90. Para las peras, este momento se encuentra entre 1.75 y 3 (cuadro 10), por lo que las segundas componentes de las peras son el valor del segundo momento multiplicado por 60. Si es mayor a 3 (cuadros 11 y 12), se penaliza multiplicándolo por 0.5, como en el caso de las manzanas y naranjas.

La tercera componente se seleccionó como el septimo momento de Hu, para dar un poco más de variabilidad a los puntos, pues es muy fluctuante entre imagen e imagen.

Nombre	Hu Moment 2	Hu Moment 7
banana1.jpg	1.252360441	4.825411926
banana10.jpg	1.236801181	-6.315256948
banana11.jpg	1.279948372	-5.317903301
banana12.jpg	1.225484402	-5.825594353
banana13.jpg	1.692355778	-9.160388391
banana2.jpg	1.550267648	6.725516442
banana3.jpg	1.091963369	5.652056711
banana4.jpg	1.063037455	-4.697589655
banana5.jpg	1.218144972	-5.078601864
banana6.jpg	1.61374888	-6.362375843
banana7.jpg	1.286421217	-5.601882796
banana8.jpg	1.028667671	-6.474687132
banana9.jpg	1.271500644	5.338953561

Cuadro 9: Valores del segundo y séptimo momento de Hu para bananas.

Nombre	Hu Moment 2	Hu Moment 7
pera1.jpg	2.425173105	8.87885001
pera10.jpg	2.733311044	9.287238683
pera11.jpg	2.359439331	-8.650134105
pera12.jpg	1.885716422	-10.40319537
pera13.jpg	2.327322931	-7.212546842
pera14.jpg	2.079979154	6.510962784
pera15.jpg	2.188082383	6.827849673
pera2.jpg	2.175779762	8.644654429
pera3.jpg	2.383011642	7.175275906
pera5.jpg	3.531961234	9.585405942
pera7.jpg	2.477859985	-9.078100938
pera8.jpg	2.778817771	-10.12447873
pera9.jpg	2.601995948	-6.610497893

Cuadro 10: Valores del segundo y séptimo momento de Hu para peras.

Nombre	Hu Moment 2	Hu Moment 7
manzana10.jpg	5.701498333	-15.253017
manzana11.jpg	3.849583392	12.94280609
manzana12.jpg	3.622174845	11.76060725
manzana2.jpg	3.745307398	11.60280968
manzana3.jpg	2.898119586	-10.77571207
manzana4.jpg	4.191646503	-10.81311242
manzana5.jpg	3.970092508	-13.78480696
manzana6.jpg	4.20938723	-11.51886683
manzana8.jpg	3.197154967	-9.409638565
manzana9.jpg	3.73419675	-12.12895282
manzanatest.jpg	4.343151733	-14.93124227

Cuadro 11: Valores del segundo y séptimo momento de Hu para manzanas.

Nombre	Hu Moment 2	Hu Moment 7
naranja1.jpg	2.289304477	7.349728077
naranja10.jpg	4.586871146	15.94460892
naranja11.jpg	5.456565147	-20.64164712
naranja2.jpg	3.777857158	12.96272406
naranja3.jpg	3.342629057	-12.4527854
naranja5.jpg	3.248552334	12.14970752
naranja6.jpg	3.170866292	13.95344088
naranja8.jpg	2.838029247	-10.97994179
naranja9.jpg	4.391032307	-19.44680882

Cuadro 12: Valores del segundo y séptimo momento de Hu para naranjas.

Los clústeres del modelo K-Means varian dependiendo de donde se produzcan los primeros centroides. Al ejecutarlo varias veces, la mejor distribución de clústeres se muestra en la Figura 10. K-Means, si bien es un algoritmo no supervisado, tiene cierto grado de supervisación al agregarse manualmente etiquetas para los centroides.

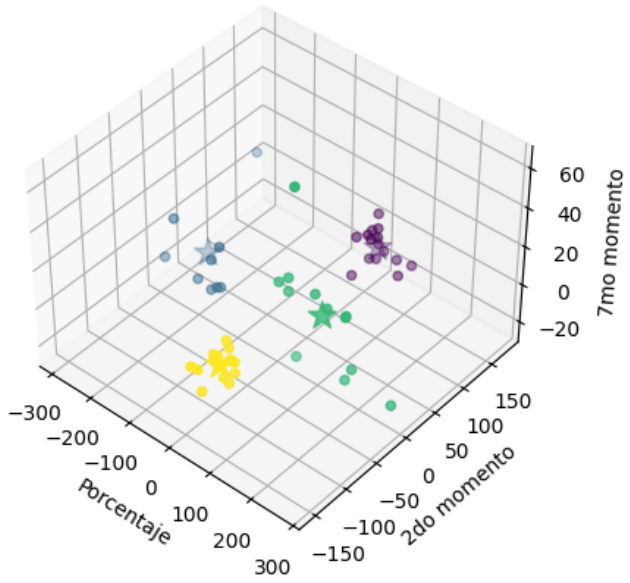

Figura 11: Modelo K-Means

Se ha testeado en numerosas imagenes de frutas, y los resultados de la efectividad del algoritmo (porcentaje de aciertos) se muestra en el cuadro 13.

Fruta	Efectividad
Banana	73 %
Manzana	69 %
Naranja	89 %
Pera	77 %

Cuadro 13: Efectividad de clasificación para cada fruta.

7. Conclusiones

El programa tiene un objetivo sencillo, y en general, los análisis realizados han sido suficientes para lograrlo con una eficiencia muy alta. Se destaca la capacidad de reconocimiento de imagen, teniendo una tasa de aciertos relativamente alta para cada fruta. Esta ha sido la parte más desafiante de este proyecto, pues

la imagen es bastante sensible a muchos factores. Por otro lado, el audio no se queda atrás. La palabra que más le cuesta reconocer al agente es "pera", pero las demás suelen acertar sin ningún problema.

Una de las ventajas es su rápida respuesta. El algoritmo, al generarse un modelo Knn y K-Means, los guarda para después utilizarlos en el futuro. Esto hace que no requiera tener que recalcular los vectores para cada audio/imagen de la base de datos, lo que eleva su rendimiento.

La desventaja que tiene este programa, es que sólo conoce 4 palabras. Si una estantería está vacía (una foto sin fruta), va a identificar una fruta de igual manera erróneamente. Para arreglarlo, posiblemente funcionaría poner una 5ta clase llamada "desconocido"ó "vacío". Con el audio sucede lo mismo.

Otra posible mejora es aumentar la cantidad de frutas que sea posible reconocer.

8. Bibliografía y/o Referencias

- Deruty, E. (2022, January 20). Intuitive understanding of MFCCs. Medium. https://medium.com/@derutycsl/intuitive-understanding-of-mfccs-836d36a1f779

- NeuralNine (2023). K-Means Clustering From Scratch in Python (Mathematical) [Video]. YouTube. https://www.youtube.com/watch?v=5w5iUbTlpMQ

- NeuralNine (2023). K-Nearest Neighbors Classification From Scratch in Python (Mathematical) [Video]. YouTube. https://www.youtube.com/watch?v=xtaom__-drE

- Russell, S. J., Norvig, P. (2004). Inteligencia artificial: un enfoque moderno. (Segunda edición). Pearson Educación, S.A., Madrid.

- Bachu, R. G., Kopparthi, S., Adapa, B., Barkana, B. D. (2023). Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy. En Avances en Ciencias de la Computación - Computación Inteligente y Aplicaciones (pp. 291-302). Springer Nature. https://doi.org/10.1007/978-90-481-3660-5_47