



Departamento de Posgrado en Ingeniería e Innovación
CETYS UNIVERSIDAD

Título de Tesis

DISEÑO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA LA
INSPECCIÓN DE CALIDAD AUTOMATIZADA DE PRODUCTOS
AGRÍCOLAS

Que para obtener el grado de
MAESTRIA EN INGENIERÍA E INNOVACIÓN

Presenta:

Kevin Fausto Ruiz Reyes

Directora de Tesis:
Dr. Kenia Picos Espinoza

Co-Director de tesis:
Dr. Ulises Orozco Rosas

Tijuana, Baja California a 24 de Marzo de 2021.



ACTA DE REVISIÓN DE TESIS

En la ciudad de Tijuana Baja California siendo el día 24 de marzo del 2021 se reunieron los miembros del Comité de Revisión de Tesis designada en el Departamento de Posgrado de CETYS Universidad Campus Tijuana para examinar la tesis titulada:

DISEÑO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA LA INSPECCIÓN DE CALIDAD AUTOMATIZADA DE PRODUCTOS AGRÍCOLAS

Tesis para obtener el grado académico de la Maestría en Ingeniería e Innovación, presentada por el alumno:

Kevin Fausto Ruiz Reyes

Los miembros del Comité de Revisión de Tesis manifestaron **APROBAR LA TESIS**, en virtud que satisface con los requisitos establecidos por el Reglamento de Posgrado.

COMITÉ DE TESIS

Dra. Kenia Picos Espinoza
Directora de Tesis

Dr. Ulises Orozco Rosas
Co-director de Tesis

Digitally signed by Lizbeth Escobedo
DN: cn=Lizbeth Escobedo, o, ou,
email=lizbeth.escobedo@cetys.mx,
c=MX
Date: 2021.03.26 08:22:18 -07'00'

Dra. Lizbeth Olivia Escobedo Bravo
Miembro del Comité

Dr. Ricardo Martínez Soto
Coordinador del Posgrado en Ingeniería
Miembro del Comité

DEDICATORIA

Esta tesis se la dedico a mi familia por darme fuerzas de seguir adelante y no desmayar en los problemas que se presentaban, enseñandome a encarar las adversidades sin perder nunca la dignidad ni desfallecer en el intento.

A mis padres por haberme forjado como la persona que soy en la actualidad; mucho de mis logros se los debo a ustedes entre los que se incluye este.

RESUMEN

En esta tesis, se presenta un método confiable para la inspección de calidad automatizada de productos agrícolas basado en un sistema de visión artificial. Los arreglos de cámaras son una estrategia eficaz para obtener imágenes de alto rendimiento e interpolación de vistas. Sin embargo, pueden presentarse desafíos como el tamaño de los datos, el procesamiento en tiempo real y la necesidad de una calibración automática. Analizamos varios métodos para la digitalización de objetos tridimensionales utilizando matrices de cámaras. Este análisis nos ayudará a comparar diferentes algoritmos y combinar varios métodos de reconstrucción 3D de una manera diferente, eficiente y rápida. Los resultados experimentales se presentan utilizando objetos de laboratorio reales y sintéticos en términos de métricas objetivas para la precisión de la reconstrucción 3D. Se utiliza de la misma manera técnicas de inteligencia artificial con relación a la clasificación adecuada en los productos agrícolas, para esto la clasificación se utiliza color, tamaño y forma. Con todos estos métodos en conjunto se obtiene un sistema de visión artificial adecuado para la aplicación de inspección de calidad de productos agrícolas.

CONTENIDO

	Page
RESUMEN	iii
LISTA DE FIGURAS	vi
AGRADECIMIENTOS	ix
1 Introducción	1
1.1 Antecedentes	2
1.2 Planteamiento del Problema	5
1.3 Preguntas de Investigación	6
1.4 Hipótesis	6
1.5 Objetivo General	6
1.6 Objetivos Específicos	7
2 Marco Teórico	8
2.1 Reconstrucción Euclidiana	8
2.2 Reconstrucción Afín	9
2.3 Reconstrucción Proyectiva	12
2.4 Estructura a partir del Movimiento	13
2.5 Clasificación	13
2.6 Detección del Tamaño	15

2.7	Detección de Forma	16
2.8	Detección de Color	17
2.9	LeNet-5 para Clasificación de imágenes	18
2.10	AlexNet para detección de imágenes	20
2.11	Arquitectura Convulocional VGG-19	22
2.12	Arquitectura ResNet	23
2.13	Data Augmentation	24
3	Propuesta metodológica	26
3.1	Reconstrucción para inspección Volumétrica	26
3.2	Clasificación de forma en producto agrícola	31
3.3	Medición del Tamaño y Color de productos agrícolas	33
4	Resultados	36
4.1	Resultados Experimentales con Imágenes Estéreo	36
4.2	Resultados Experimentales con Arreglos Multicámara	39
4.3	Resultados Experimentales para Clasificación del Pepino	42
4.4	Resultados Experimentales para la Medición y Color del Pepino	49
5	Conclusiones	52
	Bibliografía	53

LISTA DE FIGURAS

	Page
1.1 Maquinaria de Empaque Agrícola.	1
1.2 Rodajas normales y con defectos leves y graves de pepinos encurtidos.	3
1.3 Ejemplo de iluminación direccional orientada a 45°	5
2.1 (a) Tres imágenes de alta resolución (3000×2000 píxeles) de un conjunto de once del ayuntamiento de Lovaina, Bélgica. (b) Tres vistas de una reconstrucción euclidiana calculada a partir del conjunto de imágenes que muestran las 11 posiciones de la cámara y la nube de puntos.	10
2.2 Geometría epipolar afín. (a) Geometría de correspondencia. (b) Las líneas y los planos epipolares son paralelos.	11
2.3 Reconstrucción proyectiva. (a) Par de imágenes originales. (b) 2 vistas de una reconstrucción proyectiva 3D de la escena. La reconstrucción no requiere información sobre las matrices de la cámara ni información sobre la geometría de la escena. La matriz fundamental F se calcula a partir de las correspondencias de puntos entre las imágenes, las matrices de la cámara se recuperan de F y luego los puntos 3D se calculan mediante triangulación a partir de las correspondencias. Las líneas de la estructura alámbrica vinculan los puntos 3D calculados.	12
2.4 Técnica para estimar el movimiento de la cámara.	14
2.5 Diámetros de la fruta.	16

2.6	a) Región para detectar color, tamaño y forma, b) Región para línea de luz. . .	17
2.7	Representa al espacio de color RGB al HSI.	18
2.8	Arquitectura LeNet-5	20
2.9	Arquitectura AlexNet.	21
2.10	Arquitectura VGG-19.	23
2.11	Arquitectura ResNet.	24
2.12	Método Data Augmentation.	25
3.1	Diagrama de bloques del algoritmo propuesto para la reconstrucción de objetos 3D mediante matrices de cámaras.	27
3.2	Diagrama de bloques del algoritmo propuesto para la clasificación del pepino.	32
3.3	Ejemplos de visualización del pepino en la inspección de calidad mediante tamaño y color.	33
3.4	Diagrama de bloques del algoritmo propuesto para la Medición del tamaño y color del pepino.	35
4.1	Resultados experimentales de imágenes estéreo. (a) Par de imágenes izquierda y (b) derecha de una cámara estéreo. (c) Imagen de profundidad dada por disparidad estéreo, y (d) Reconstrucción de escena en archivo de nube de puntos.	38
4.2	Imágenes de ejemplo para la reconstrucción de escenas I_1, I_2, I_3 y I_4 capturadas desde el sistema de matriz multicámara propuesto.	40
4.3	Malla resultante de la reconstrucción de la escena (a) antes del proceso de optimización y (b) después del refinamiento del proceso de optimización.	40
4.4	Resultados preliminares de la reconstrucción de escenas en 3D a partir de una matriz de cámaras. (a) Datos de salida de la nube de puntos y (b) nube de puntos con representación del color de la escena.	41
4.5	Diferentes ejemplos de imágenes de pepino para la clase Curva.	43
4.6	Diferentes ejemplos de imágenes de pepino para la clase No Curva.	44

4.7	Aumento de datos de la clase Curva a partir de una entrada de imagen original con funciones aleatorias para aplicar el reflejo horizontal, reflejo vertical, y escalamiento. (a) Imagen original. (b) Imágenes aumentadas.	45
4.8	Aumento de datos de la clase No Curva a partir de una entrada de imagen original con funciones aleatorias para aplicar el reflejo horizontal, reflejo vertical, y escalamiento. (a) Imagen original. (b) Imágenes aumentadas.	46
4.9	Ejemplo de Imágenes (a) Entrenamiento y (b) Prueba.	47
4.10	Se muestra el resultado de las predicciones cuando es curva y no curva.	47
4.11	Se muestra el resultado de la gráfica de 3 capas, donde se ve la Precisión y función de perdida.	48
4.12	Se muestra el resultado de la gráfica de 5 capas, donde se ve la Precisión y función de perdida	48
4.13	Muestra de imagen de entrada.	50
4.14	Resultado de la medición automática del pepino mediante los pixeles a pulgadas.	50
4.15	Resultado de la medición de imagen por cada canal de color RGB.	51

AGRADECIMIENTOS

Agradezco a dios por haber bendecido mi vida y guiado a cada uno de mis pasos.

A mi Asesora de tesis la Dr. Kenia Picos Espinoza por haberme brindado la oportunidad de recurrir a su capacidad y conocimiento científico, así como también haberme tenido la paciencia del mundo para guiarme durante todo el proceso de tesis.

Y para finalizar, también agradezco a todos aquellos que fueron mis compañeros de clase, así mismo también a todos los profesores que me han aportado un gran porcentaje de conocimientos adquiridos.

Capítulo 1

Introducción

En las últimas décadas, el crecimiento de las actividades industriales ha creado una demanda continua de nuevos sistemas inteligentes con el propósito de satisfacer sus necesidades. En este sentido los sistemas inteligentes tienen la capacidad de satisfacer las actividades realizadas por una persona continuamente. A continuación se muestra la imagen de una maquina de empaque agrícola, Figura 1.1.



Figura 1.1: Maquinaria de Empaque Agrícola.

1.1 Antecedentes

A continuación se muestran antecedentes con relación a la clasificación y selección de frutas y Hortalizas.

Las imágenes hiperespectrales son útiles para detectar defectos internos de los pepinos encurtidos. Sin embargo, la técnica aún no es adecuada para la implementación en línea de alta velocidad debido a los desafíos en el análisis de imágenes hiperespectrales a gran escala. Esta investigación tuvo como objetivo seleccionar las bandas de ondas óptimas a partir de los datos de la imagen hiperespectral, de modo que se puedan implementar en un sistema de inspección basado en imágenes hiperespectrales o multispectrales para la detección automática de defectos internos de los pepinos encurtidos. Se adquirieron imágenes de reflectancia hiperespectral (400-700 nm) y transmitancia (700-1.000 nm), utilizando un sistema de imagen hiperespectral desarrollado internamente que funcionaba a dos velocidades de transportador de 85 y 165 mm/s, para 300 pepinos encurtidos "Journey" antes y después de que el daño interno fuera inducido por carga mecánica. Redundancia mínima - relevancia máxima (MRMR) se utilizó para la selección óptima de bandas de ondas, y también se aplicaron las cargas del análisis de componentes principales (PCA) para identificar cualitativamente las bandas de ondas importantes que están relacionadas con las características específicas. Se realizó un análisis discriminante con el clasificador de distancia de Mahalanobis para las clasificaciones de dos clases (es decir, normal y defectuoso) y de tres clases (es decir, normal, ligeramente defectuoso y severamente defectuoso) utilizando los espectros medios y las características de textura (energía y varianza) de las regiones de interés en las imágenes espectrales en relaciones de banda de ondas seleccionadas. Los resultados de clasificación basados en la selección de bandas de ondas MRMR fueron generalmente mejores que los de las clasificaciones basadas en PCA. La relación de dos bandas de 887/837 nm de MRMR dio los mejores resultados de clasificación general, con una precisión de 95,1 y 94,2% a las velocidades del transportador de 85 y 165 mm/s, respectivamente, para la clasificación de dos



Figura 1.2: Rodajas normales y con defectos leves y graves de pepinos encurtidos.

clases. Las precisiones de clasificación más altas para la clasificación de tres clases basada en la relación óptima de dos bandas de 887/837 nm fueron 82,8 y 81,3% a las velocidades del transportador de 85 y 165 mm/s, respectivamente. La clasificación basada en espectros medios logró mejores resultados que la clasificación basada en características de textura, excepto en la clasificación de tres clases para la mayor velocidad del transportador. Las precisiones de clasificación general para todas las relaciones de banda de ondas seleccionadas a la velocidad del transportador baja fueron ligeramente más altas que las de la velocidad del transportador más alta, ya que la velocidad baja resultó en más líneas de escaneo, por lo tanto, imágenes hiperespectrales de mayor resolución espacial. La relación de dos bandas identificada de 887/837 nm en el modo de transmitancia podría aplicarse para la detección rápida de defectos internos en tiempo real de los pepinos encurtidos [5]. A continuación se muestra la imagen de la calidad de las rodajas de los pepinos, figura 1.2.

La aplicación de la visión artificial en la agricultura ha aumentado considerablemente en los últimos años. Hay muchos campos en los que se trata de la visión computarizada, incluyendo el mapeo terrestre y aéreo de recursos naturales, monitoreo de cultivos, agricultura de precisión, robótica, orientación automática, inspección no destructiva de las propiedades del producto, control de calidad y clasificación en líneas de procesamiento y, en general, automatización de procesos. Esta amplia gama de aplicaciones es el resultado del hecho de que los sistemas de visión artificial proporcionan cantidades sustanciales de información sobre la naturaleza y los atributos de los objetos presentes en una escena. Además, la visión artificial abre la posibilidad de estudiar estos objetos en regiones del espectro electromagnético en las que el ojo humano no es sensible, como es el caso de las regiones ultravioleta (UV) o infrarroja. La inspección automatizada de los productos agrícolas muestra ciertas particularidades y problemas que no están presentes en otros campos debido a su naturaleza biológica. Mientras que los productos manufacturados a menudo presentan colores, formas, tamaños y otras características externas similares, las frutas y verduras pueden mostrar características muy diferentes de un artículo a otro. Una sola fruta puede tener un color, tamaño y forma diferentes de otro, a pesar de que ambos fueron recogidos el mismo día del mismo árbol. Las frutas y verduras cambian naturalmente su color o textura después de ser cosechadas, y estas características dependen de su madurez y cómo se almacenan (humedad ambiente y temperatura, enfermedades fúngicas, presencia de volátiles, duración del almacenamiento, etc.). Además, el color en una zona particular de la piel de una fruta sana puede coincidir con el color de una mancha en la superficie de otro fruto de la misma variedad. Además, es esencial que se identifique la presencia de extremos de tallo, hojas, suciedad o cualquier material extraño y no se confunda con defectos verdaderos de la piel. Por otro lado, los mercados exigen un procesamiento de imágenes muy rápido, y por esta razón, se debe encontrar un equilibrio entre velocidad y precisión [19]. A continuación se muestra un ejemplo de la iluminación para la calidad las frutas y hortalizas, figura 1.3.

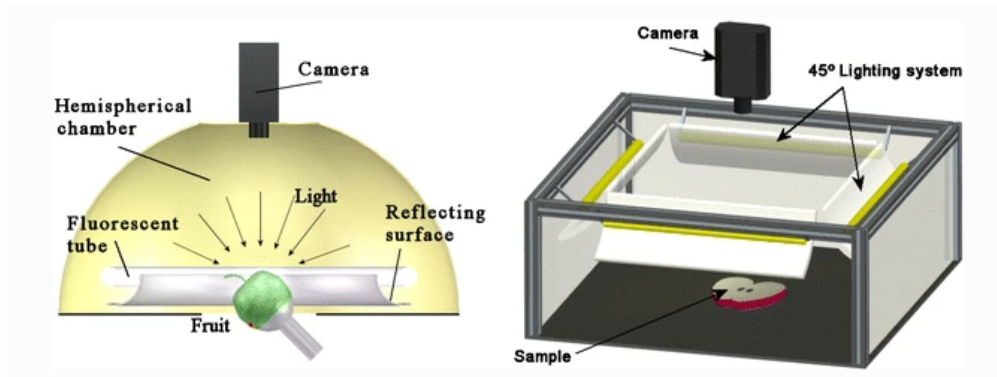


Figura 1.3: Ejemplo de iluminación direccional orientada a 45°.

1.2 Planteamiento del Problema

En la Industria Agrícola se importan un gran número de frutas y Hortalizas (Pepino, Tomate, chile morrón etc), por día que en su mayoría requieren de una buena clasificación y selección para su calidad adecuada. Este proceso se realizan mediante manera manual y en algunas industrias de manera semiautomática, los operadores de manera visual ven si tienen defectos y también el tamaño adecuado para su empaque final, por otra parte cuando es de manera semiautomática cuando ya hay una preselección de las hortalizas mediante una máquina de selección y después el operador selecciona de manera manual, así de esta manera llega a su proceso final para la importación o exportación. Este proceso dura 6 horas por turno aproximadamente, tiempo excesivo para producción de la empresa, considerando tiempos muertos, por cansancio físico y posibles errores del operario debido a su intervención manual; lo que ha provocado que no se cumplan con las demandas de producción y una insatisfacción para el cliente.

Por ello, se requiere de un programa (algoritmo) y automatización adecuado que sea capaz de realizar de manera automática la selección y así validar cada hortaliza de manera más controlada. Esto debido a los incidentes que anteriormente se tenían en los cuales el operador por cansancio selecciona mal la hortaliza, lo que provoca que al empacarse no es de la clasificación adecuada. Estos al llegar al cliente y ser inspeccionados se encontraban un

producto que no cumplía con alguno de los requerimientos necesarios y, como consecuencia se regresaba todo el lote, generando pérdidas económicas para la industria agrícola en transportación e incumplimiento en tiempo de entrega del producto requerido.

1.3 Preguntas de Investigación

- ¿Es posible realizar la inspección volumétrica de productos agrícolas mediante técnicas de reconstrucción digital?
- ¿Qué técnicas de inteligencia artificial pueden realizar la extracción de parámetros tales como longitud, área, forma y defectos de productos agrícolas?
- ¿Se puede realizar la inspección de calidad de productos agrícolas mediante sistemas de visión automatizada?

1.4 Hipótesis

Al implementar un sistema de visión artificial se podrá mejorar la inspección automatizada de la calidad de productos agrícolas.

1.5 Objetivo General

- Diseñar un sistema de visión automatizado mediante métodos eficientes de reconstrucción digital y técnicas de inteligencia artificial para la clasificación de productos agrícolas.

1.6 Objetivos Específicos

- Implementar métodos de reconstrucción digital para la inspección volumétrica de productos agrícolas.
- Aplicar técnicas de inteligencia artificial para la clasificación de parámetros tales como longitud, área, forma y defectos de productos agrícolas.
- Integrar un sistema de visión automatizada para la inspección de calidad de productos agrícolas.

Capítulo 2

Marco Teórico

El marco teórico que fundamenta toda la investigación proporcionará una idea mas clara de este tema. Se encontraran conceptos básicos y específicos de reconstrucción digital y clasificación de productos agrícolas mediante técnicas de visión artificial.

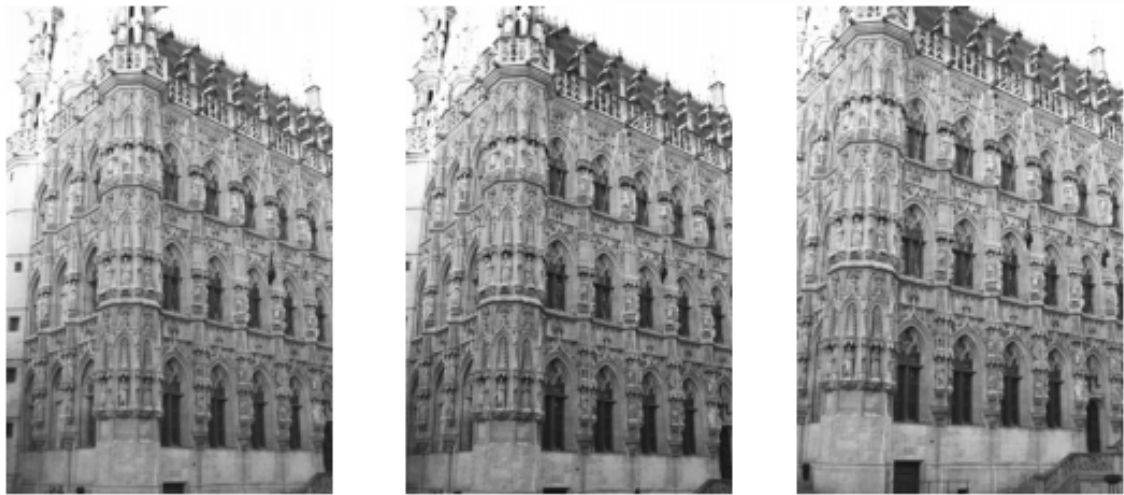
2.1 Reconstrucción Euclidiana

Para obtener una reconstrucción del modelo en el que los objetos tienen su forma correcta (euclidiana), es necesario determinar la calibración de las cámaras. Es fácil ver que esto es suficiente para determinar la estructura euclidiana de la escena. Como hemos visto, determinar la estructura euclidiana del mundo es equivalente a especificar el plano en el infinito y la cónica absoluta. De hecho, dado que la cónica absoluta se encuentra en un plano, el plano en el infinito, es suficiente para encontrar la cónica absoluta en el espacio. Ahora, supongamos que hemos calculado una reconstrucción proyectiva del mundo, utilizando cámaras calibradas. Por definición, esto significa que la imagen cónica absoluta (IAC) es conocida en cada una de las imágenes; déjelo denotar por ω_i en la i -ésima imagen. La retroproyección

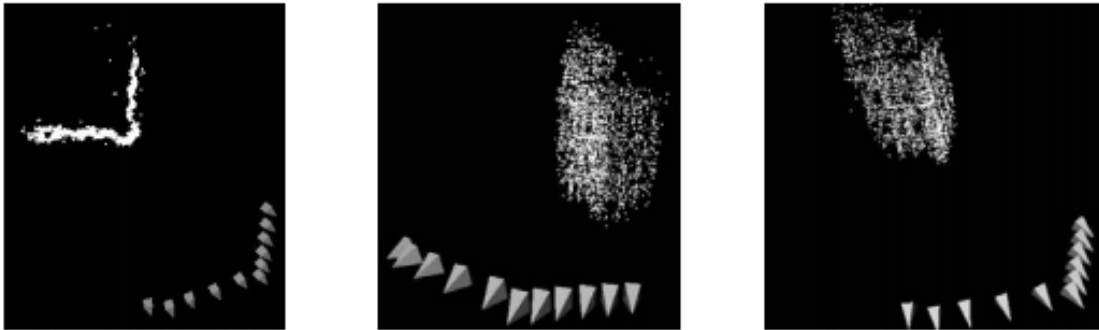
de cada ω_i es un cono en el espacio, y la cónica absoluta debe estar en la intersección de todos los conos. Dos conos en general se cruzan en una curva de cuarto grado, pero dado que deben cruzarse en una cónica, esta curva debe dividirse en dos cónicas. Por lo tanto, la reconstrucción de la cónica absoluta a partir de dos imágenes no es única, sino que hay dos posibles soluciones en general. Sin embargo, a partir de tres imágenes o más, la intersección de los conos es única en general. Así se determina la cónica absoluta y con ella la estructura euclidiana de la escena. Por supuesto, si se conoce la estructura euclidiana de la escena, también se conoce la posición de la cónica absoluta. En este caso, podemos proyectarlo nuevamente en cada una de las imágenes, produciendo el IAC en cada imagen y, por lo tanto, calibrando las cámaras. Por lo tanto, el conocimiento de la calibración de la cámara es equivalente a poder determinar la estructura euclidiana de la escena [10].

2.2 Reconstrucción Afín

La cámara afín es una aproximación extremadamente útil y bien condicionada en muchas situaciones prácticas. Su ventaja es su linealidad, que se puede implementar mediante álgebra lineal (matrices inversas o descomposición de valores singulares). Algunos de los métodos proyectivos involucran polinomios de alto orden (como para la triangulación) o solo son posibles mediante minimización numérica (como en la estimación Gold Standard de la matriz fundamental \mathbf{F}). Primero describimos las propiedades de la geometría epipolar de dos cámaras afines y su cálculo óptimo a partir de correspondencias de puntos. A esto le sigue la triangulación y la reconstrucción afina. Finalmente, se esbozan las ambigüedades en la reconstrucción que resultan de la proyección paralela y se calculan los parámetros de movimiento no ambiguos a partir de la geometría epipolar. Como se muestra en la Fig. 2.2, la geometría epipolar de dos cámaras afines es idéntica a la de dos cámaras en perspectiva. En geometría epipolar, un punto en una vista define una línea epipolar en la otra vista, y luego, tales



a



b

Figura 2.1: (a) Tres imágenes de alta resolución (3000×2000 píxeles) de un conjunto de once del ayuntamiento de Lovaina, Bélgica. (b) Tres vistas de una reconstrucción euclidiana calculada a partir del conjunto de imágenes que muestran las 11 posiciones de la cámara y la nube de puntos.

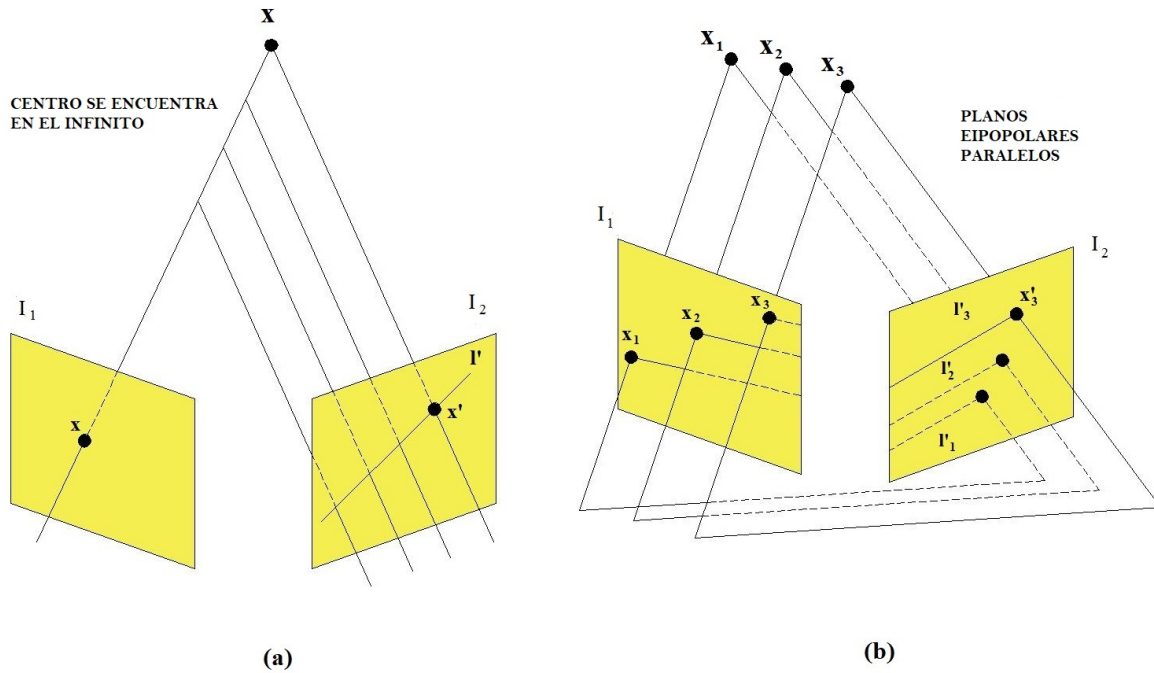


Figura 2.2: Geometría epipolar afín. (a) Geometría de correspondencia. (b) Las líneas y los planos epipolares son paralelos.

líneas epipolares se cruzan en el epipolar. Los rayos de proyección son paralelos y se cruzan en el infinito. Un punto \mathbf{x} se proyecta hacia atrás a un rayo en el espacio 3D definido por el primer centro de la cámara (en el infinito) y \mathbf{x} . Este rayo se representa como una línea l' en la segunda vista. El punto \mathbf{x} del espacio 3D que se proyecta a \mathbf{x} se encuentra en este rayo, por lo que la imagen de \mathbf{X} en la segunda vista se encuentra en l' . La diferencia es que debido a que las cámaras están afines, sus centros están en el infinito y hay una proyección paralela de una escena a otra. Esto conduce a ciertas simplificaciones en la geometría epipolar afín [10].

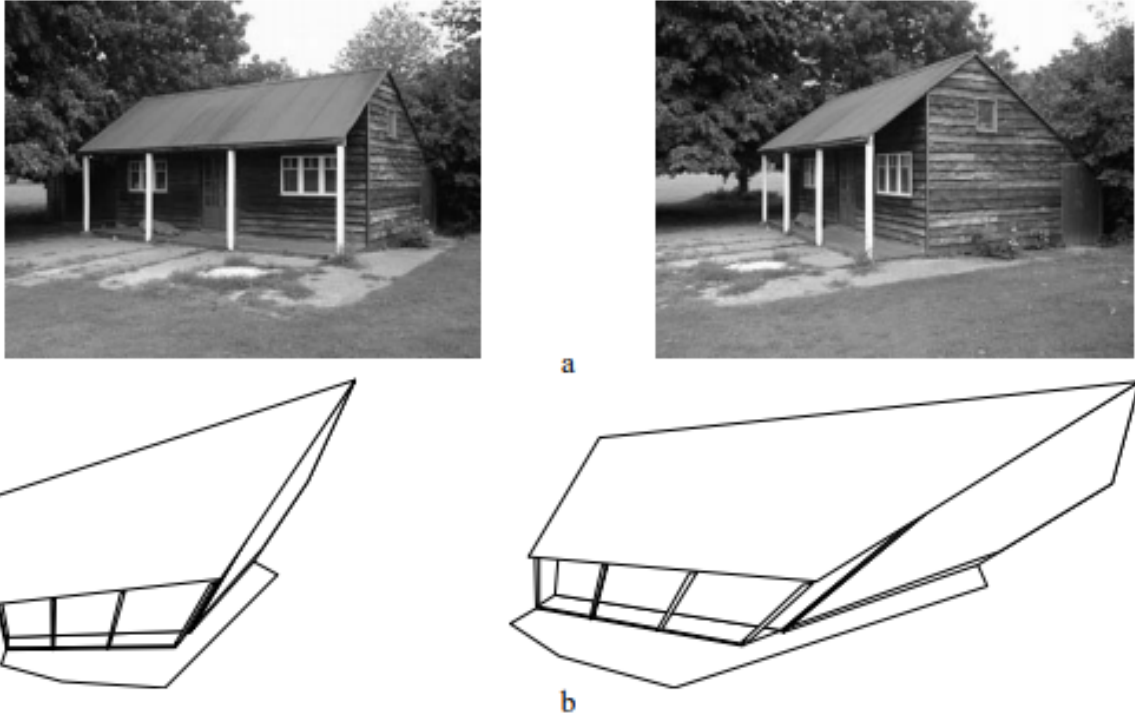


Figura 2.3: Reconstrucción proyectiva. (a) Par de imágenes originales. (b) 2 vistas de una reconstrucción proyectiva 3D de la escena. La reconstrucción no requiere información sobre las matrices de la cámara ni información sobre la geometría de la escena. La matriz fundamental F se calcula a partir de las correspondencias de puntos entre las imágenes, las matrices de la cámara se recuperan de F y luego los puntos 3D se calculan mediante triangulación a partir de las correspondencias. Las líneas de la estructura alámbrica vinculan los puntos 3D calculados.

2.3 Reconstrucción Proyectiva

En la reconstrucción proyectiva, sea una escena de entrada en la que un conjunto de puntos 3D \mathbf{X}_j es visto por un conjunto de cámaras con matrices \mathbf{P}^i . Denote con \mathbf{x}_j^i las coordenadas del punto j tal como lo ve la cámara i -ésima. Queremos resolver el siguiente problema de reconstrucción: dado el conjunto de coordenadas de imagen \mathbf{x}_j^i , encontramos el conjunto de matrices de cámara, \mathbf{P}^i , y los puntos \mathbf{X}_j de modo que $\mathbf{P}^i \mathbf{X}_j = \mathbf{x}_j^i$. Se da una reconstrucción proyectiva sin más restricciones porque los puntos \mathbf{X}_j pueden diferir por una transformación proyectiva 3D arbitraria de la reconstrucción verdadera [10].

2.4 Estructura a partir del Movimiento

La estructura a partir del movimiento (SFM) es el proceso de estimación de la estructura 3D de una escena a partir de un conjunto de imágenes 2D. El SFM se utiliza en muchas aplicaciones, como el escaneo 3D y la realidad aumentada. SFM se puede calcular a medida que se aborda el problema, que a menudo depende de diferentes factores, como el número de cámaras, el orden de las imágenes que se presentan, el tipo de cámara y la resolución. Si las imágenes se toman con una sola cámara calibrada, entonces la estructura 3D y el movimiento de la cámara solo se pueden recuperar a escala. En este proceso se realiza un reescalado de estructura, en el que los objetos más cercanos a la cámara se representan en la misma imagen en comparación con agrandar el objeto y alejar la cámara [10]. Para calcular la escala real de la estructura y el movimiento en unidades, se necesita información adicional. Esta información se puede representar por el tamaño de un objeto en la escena y la información de otro sensor, por ejemplo, un odómetro. Un requisito previo importante es la determinación de la posición y la calibración de la cámara, que puede expresarse mediante una matriz de proyección. La teoría geométrica de la estructura a partir del movimiento permite que las matrices de proyección y los puntos 3D se calculen simultáneamente utilizando solo los puntos correspondientes en cada vista. La estructura a partir de técnicas de movimiento se utiliza en una amplia gama de aplicaciones, incluida la reconstrucción fotogramétrica, de objetos y de escenas. La estrategia de SFM incremental tiene ventajas en precisión, integridad y solidez, que incluyen el proceso de iteración para agregar imágenes una por una [13].

2.5 Clasificación

La clasificación de objetos a partir de imágenes se realiza principalmente mediante un conjunto de características de clasificación y un algoritmo de clasificación. En la práctica, no es

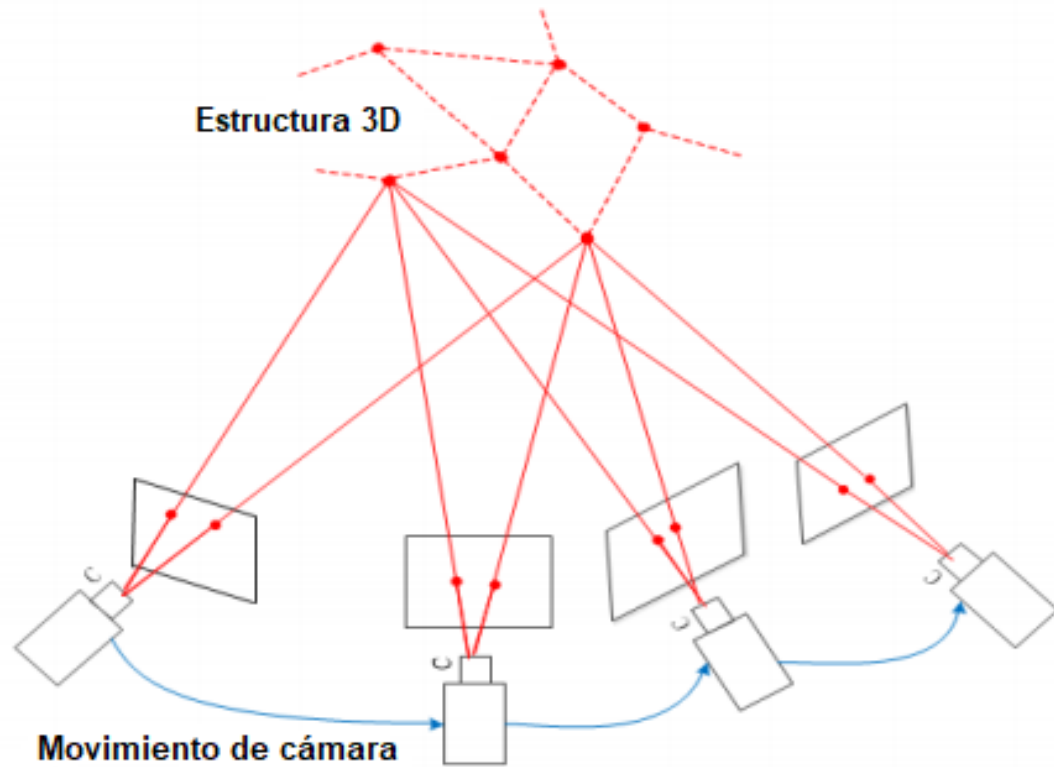


Figura 2.4: Técnica para estimar el movimiento de la cámara.

posible una clasificación perfecta debido a las diferencias dentro de las clases, las similitudes entre las clases y el ruido no deseado introducido en la imagen. La elección del clasificador y la regla de decisión juegan un papel importante en la clasificación precisa. El proceso de selección y extracción de características de clasificación debe decidirse con mucho cuidado analizando el dominio del problema. El conjunto de características se extrae de las imágenes de entrenamiento y las etiquetas de clase correspondientes se asignan a un algoritmo de clasificación para entrenar el modelo de predicción. El mismo conjunto de características que se dan durante el entrenamiento se extraen de la imagen de prueba y se pasan al modelo de predicción entrenado. Este modelo devuelve la etiqueta de clase de una muestra de prueba desconocida en función de la probabilidad de coincidencia con una clase en particular [14].

2.6 Detección del Tamaño

El tamaño es una de las características importantes de la fruta, de acuerdo con el tamaño de la fruta para detectar la clasificación, seleccionar el tamaño de la fruta es básicamente el mismo, es propicio para el envasado y procesamiento y procesamiento. Se utiliza para medir el tamaño del área de la fruta, el perímetro, la longitud y el tamaño del eje, el volumen, etc. La apariencia de los productos agrícolas es compleja, de forma irregular, desde diferentes ángulos, mostrando diferentes formas y colores, por lo que estas características dependen de la cámara para obtener el ángulo de imagen. Combinando la información de la imagen de diferentes ángulos, se puede obtener el tamaño relativo del objetivo. Throop y col. [15] mediante la traslación y rotación de la manzana para obtener diferentes ángulos de la imagen, de acuerdo con estas imágenes para calcular el radio y área ecuatorial de la manzana; luego, la manzana como un elipsoide, calcule el eje largo y el eje corto de la manzana, y de acuerdo con Esto estime el tamaño de la manzana. El volumen también es un indicador común del tamaño de la fruta, pero este método es más complejo y computacionalmente más grande que el perímetro basado en el área, etc. Koc [2] estudió el algoritmo de cálculo de volumen de la sandía. Al procesar las imágenes recolectadas, los contornos de las imágenes se obtuvieron en diferentes planos de proyección, y se formaron los cortes de contorno, y luego los cortes de contorno se rotaron a lo largo del eje x. Las rodajas se combinan para medir el volumen de la sandía. En el aspecto doméstico, Ying Yi Bin et al. [28] usó la tecnología de visión artificial para detectar con precisión el tamaño de la fruta, estableció la relación cuantitativa entre el punto en la imagen y el punto en el objeto medido, y usó la información de los límites del objeto para encontrar las coordenadas del centroide del objeto, y luego propuso un algoritmo de corrección de área, precisión de detección del 96%. Sobre la base de esto, Rao Xiuqin et al. [28] analizó los diversos errores y causas de la detección del tamaño basándose en la relación entre la fruta, el lente de la cámara y la imagen, y dio la calibración La fórmula para calcular el error y la fórmula para calcular el radio [30, 22].

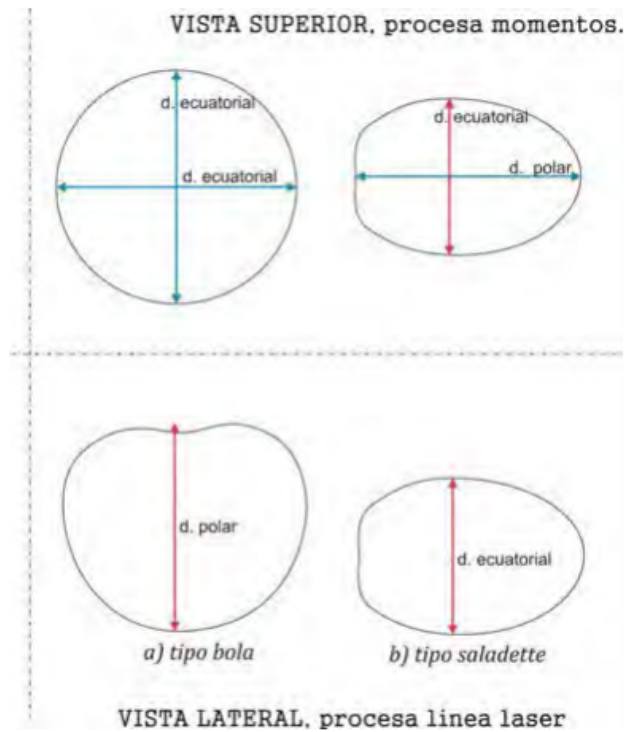


Figura 2.5: Diámetros de la fruta.

2.7 Detección de Forma

La forma de la fruta es una de las cualidades de apariencia importantes de la fruta. La deformación de forma extraña de la fruta es muy baja, o incluso simplemente no se puede vender, lo que debe considerarse una prueba de calidad de la fruta. La forma de los productos agrícolas y otros productos agrícolas y secundarios es irregular, cómo describir con precisión la forma de los productos agrícolas, es una de las cuestiones importantes. Xu y col. [27] estudió el método de clasificación de la forma de la fresa. Primero, definimos cuatro tipos de fresas, como cono largo, cuadrado, cónico y redondo, y extraemos la secuencia lineal de los contornos de cada fresa. Unay et al. [24] utilizó características de circularidad para caracterizar la forma de la fruta [30, 22].

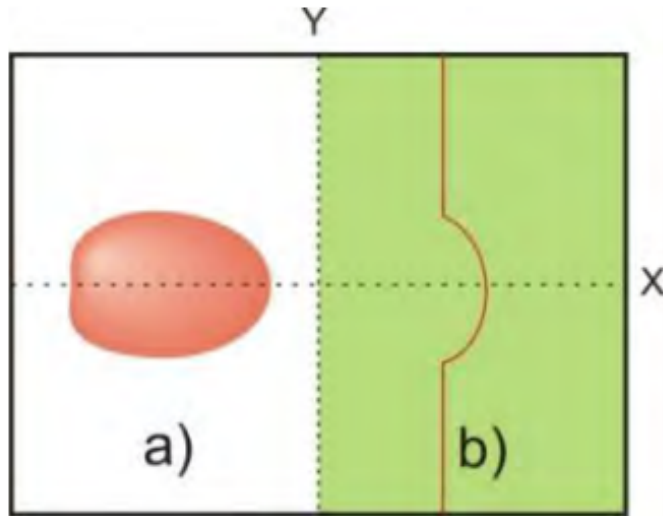


Figura 2.6: a) Región para detectar color, tamaño y forma, b) Región para línea de luz.

2.8 Detección de Color

El color de la apariencia es uno de los parámetros de apariencia más importantes de la fruta, los consumidores a menudo de acuerdo con el color de la fruta para decidir si comprar. En un sistema de imágenes de visión por computadora, el color de un píxel generalmente se representa mediante tres valores de coordenadas de la imagen de la fruta en el espacio de color RGB, es decir, (R, G, B). Blasco y col. [3] utilizó los promedios RGB de píxeles y los umbrales simples para R/G para distinguir cuatro variedades diferentes de granadas en tiempo real, con una tasa de éxito de más del 90%. Como se muestra en la Fig. 2.7, en la clasificación del color de la manzana, Hon Wenjun [26] propuso usar el modelo HIS (Tono, intensidad y saturación) para describir sus signos de color, y de acuerdo con las características del histograma de croma, usar la media de cuatro cromas para reemplazar el valor del color de la manzana.

Sobre esta base, se establece el sistema de clasificación de frutas de la red neuronal de propagación hacia atrás (BP) y la precisión de clasificación es alta, lo que puede cumplir con los requisitos de clasificación. Kan Qing [20] sobre características de color de la superficie

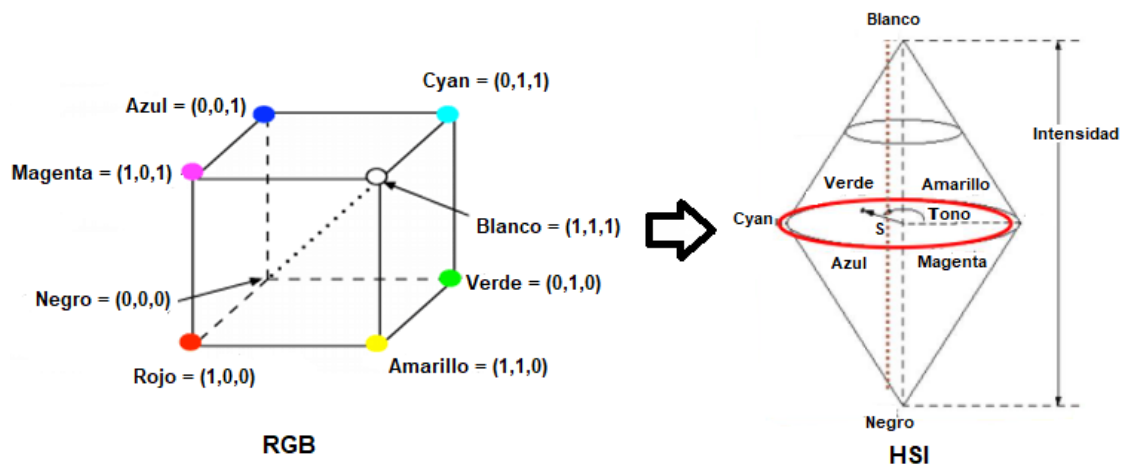


Figura 2.7: Representa al espacio de color RGB al HSI.

de la manzana se analizaron, obtener tres parámetros característicos: la relación del área de la superficie roja, el color de la superficie promedio y la uniformidad del color de la superficie. Los dos primeros valores propios se obtienen calculando la frecuencia acumulada correspondiente al valor de cromaticidad apropiado y el promedio de la cromaticidad de cada píxel en la imagen de la manzana. El tercer valor propio se obtiene calculando la distancia entre el centroide del área roja en la imagen de la manzana y el centroide de toda la imagen de la manzana, y la diferencia en la cromaticidad promedio de la misma manzana. Los resultados de la prueba muestran que la precisión de clasificación es del 88,9% o más [30, 4].

2.9 LeNet-5 para Clasificación de imágenes

LeNet-5 consta de 7 capas, sin contar la entrada, todas las cuales contienen parámetros entrenables (pesos) como se muestra en la figura 2.8. La entrada es una imagen de 32x32 píxeles. Esto es significativamente más grande que el carácter más grande de la base de datos (como máximo 20x20 píxeles centrados en un campo de 28x28). La razón es que es deseable que en el centro del campo receptivo de los detectores de características de más alto nivel

puedan aparecer rasgos distintivos potenciales como los puntos finales de trazo o la esquina. La capa C1 es una capa convolucional con 6 mapas de características. Cada unidad en cada mapa de características está conectada a un vecindario de 5×5 en la entrada. el tamaño de los mapas de características es 28×28 , lo que evita que la conexión de la entrada se caiga del límite. C1 contiene 156 parámetros entrenables y 122.304 conexiones. La capa S2 es una capa de submuestreo con 6 mapas de características de tamaño 14×14 . Cada unidad en cada mapa de características está conectada a un vecindario de 2×2 en el mapa de características correspondiente en C1. Se suman las cuatro entradas a una unidad en S2, se multiplican por un coeficiente entrenable y se suman a un sesgo entrenable. El resultado pasa a través de una función sigmoidea. los campos receptivos 2×2 no se superponen, por lo tanto, los mapas de características en S2 tienen la mitad del número de filas y columnas que los mapas de características en C1. La capa S2 tiene 12 parámetros entrenables y 5880 conexiones. La capa C3 es una capa convolucional con 16 mapas de características. Cada unidad en cada mapa de características está conectada a varios vecindarios de 5×5 en ubicaciones idénticas en un subconjunto de mapas de características de S2. La capa S4 es una capa de submuestreo con 16 mapas de características de tamaño 5×5 . Cada unidad en cada mapa de características está conectada a vecindarios de 2×2 en el mapa de características correspondiente en C3, en una guerra similar a C2 y S2. La capa S4 tiene 32 parámetros entrenables y 2000 conexiones. La capa C5 es una capa convolucional con 120 mapas de características. Cada unidad está conectada a vecindarios de 5×5 en los 16 mapas de características de S4. Aquí, debido a que el tamaño de S4 también es 5×5 , el tamaño de los mapas de características de C5 es 1×1 : esto equivale a una conexión completa entre S4 y C5. C5 está etiquetado como una capa convolucional, en lugar de una capa completamente conectada, porque si la entrada LeNet-5 se hiciera más grande con todo más grande que 1×1 . La capa F6, contiene 84 unidades y está completamente conectada a C5. Tiene 10.164 parámetros entrenables. Finalmente, la capa de salida se compone de unidades de función de base radial euclidiana (RBF), una para cada clase, con 84 entradas cada una [29].

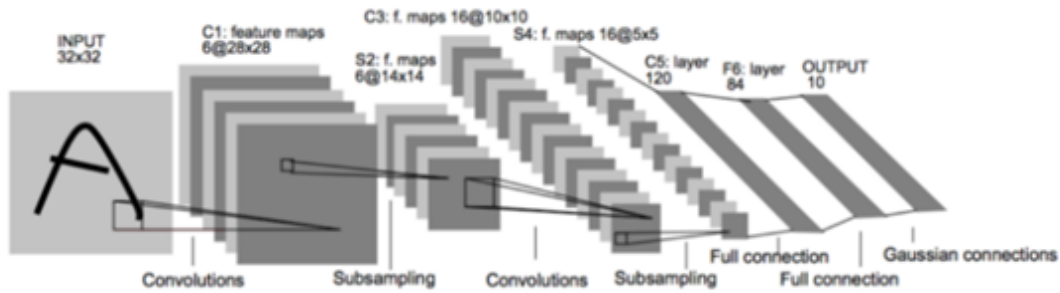


Figura 2.8: Arquitectura LeNet-5

2.10 AlexNet para detección de imágenes

Como se muestra en la Figura 2.9, la red contiene ocho capas con pesos; los primeros cinco son convolucionales y los tres restantes están completamente conectados. La salida de la última capa completamente conectada se alimenta a un softmax de 1000 vías que produce una distribución sobre las etiquetas de 1000 clases. Nuestra red maximiza el objetivo de regresión logística multinomial, que equivale a maximizar el promedio en los casos de entrenamiento del log-probabilidad de la etiqueta correcta bajo la distribución de predicción. Los núcleos de las capas convolucionales segunda, cuarta y quinta están conectados solo a los mapas de núcleos de la capa anterior que residen en la misma GPU. Los núcleos de la tercera capa convolucional están conectados a todos los mapas de núcleos en la segunda capa. Las neuronas de las capas completamente conectadas están conectadas a todas las neuronas de la capa anterior. Las capas de normalización de respuesta siguen a la primera y segunda capas convolucionales. Las capas de agrupación máxima, siguen tanto las capas de normalización de respuesta como la quinta capa convolucional. La no linealidad ReLU se aplica a la salida de cada capa convolucional y completamente conectada. La primera capa convolucional filtra la imagen de entrada de $224 \times 224 \times 3$ con 96 núcleos de tamaño $11 \times 11 \times 3$ con un paso de 4 píxeles (esta es la distancia entre los centros de campo receptivo de las neuronas vecinas en un mapa del núcleo). La segunda capa convolucional toma como entrada la

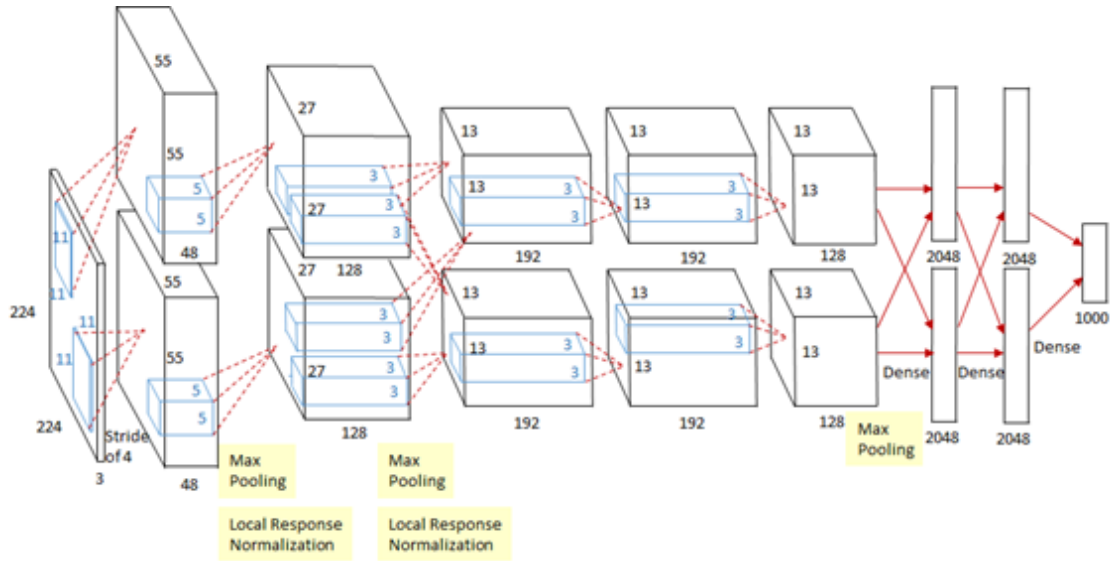


Figura 2.9: Arquitectura AlexNet.

salida (con respuesta normalizada y agrupada) de la primera capa convolucional y la filtra con 256 núcleos de tamaño $5 \times 5 \times 48$. La tercera, cuarta y quinta capas convolucionales están conectadas entre sí sin cualquier capa intermedia de agrupación o normalización. La tercera capa convolucional tiene 384 núcleos de tamaño $3 \times 3 \times 256$ conectados a las salidas (normalizadas, agrupadas) de la segunda capa convolucional. La cuarta capa convolucional tiene 384 núcleos de tamaño $3 \times 3 \times 192$, y la quinta capa convolucional tiene 256 núcleos de tamaño $3 \times 3 \times 192$. Las capas completamente conectadas tienen 4096 neuronas cada una [1].

2.11 Arquitectura Convulocional VGG-19

Como se muestra en la Figura 2.10. El VGG-19 a la entrada a nuestras ConvNets es una imagen RGB de tamaño fijo 224×224 . El único preprocesamiento que hacemos es restar el valor RGB medio, calculado en el conjunto de entrenamiento, de cada píxel. La imagen pasa a través de una pila de capas convolucionales (conv.), Donde usamos filtros con un campo receptivo muy pequeño: 3×3 (que es el tamaño más pequeño para capturar la noción de izquierda / derecha, arriba / abajo, centro). En una de las configuraciones también utilizamos filtros de convolución 1×1 , que se pueden ver como una transformación lineal de los canales de entrada (seguida de no linealidad). La zancada de convolución se fija en 1 píxel; el relleno espacial de conv. La entrada de la capa es tal que la resolución espacial se conserva después de la convolución, es decir, el relleno es de 1 píxel para 3×3 conv. capas. La agrupación espacial se lleva a cabo mediante cinco capas de agrupación máxima, que siguen algunas de las conv. capas (no todas las capas de conv. van seguidas de la agrupación máxima). La agrupación máxima se realiza en una ventana de 2×2 píxeles, con paso 2. Una pila de capas convolucionales (que tiene una profundidad diferente en diferentes arquitecturas) va seguida de tres capas totalmente conectadas (FC): las dos primeras tienen 4096 canales cada una, la tercera realiza una clasificación ILSVRC de 1000 vías y, por lo tanto, contiene 1000 canales (una para cada clase). La última capa es la capa soft-max. La configuración de las capas completamente conectadas es la misma en todas las redes [17].

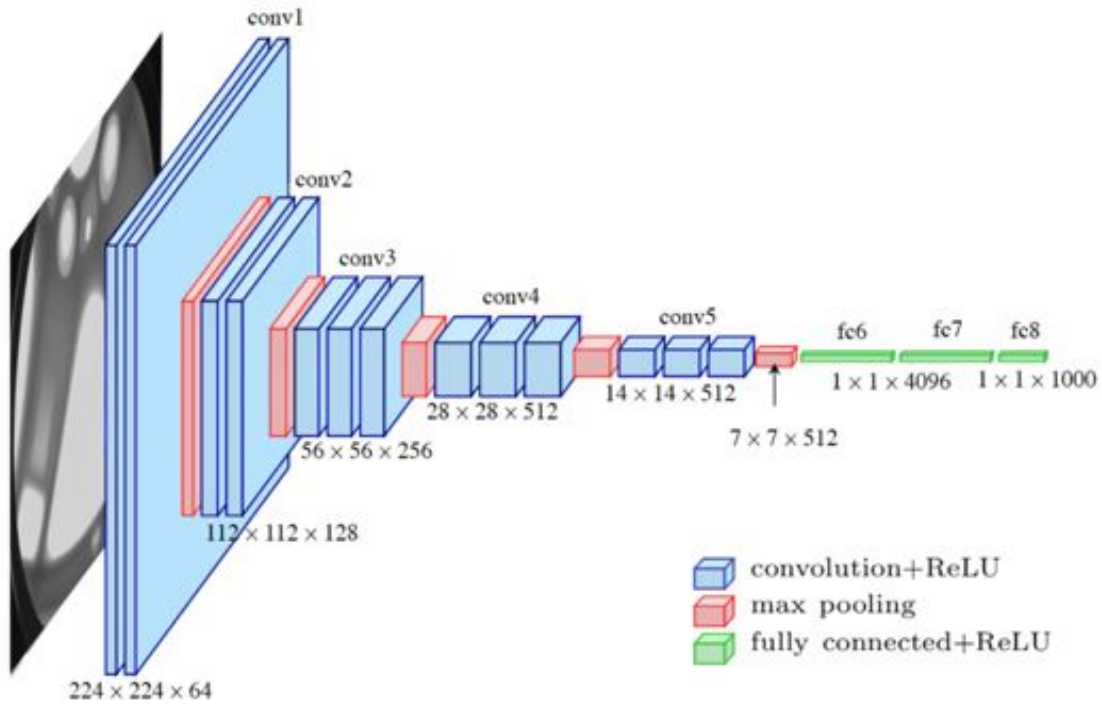


Figura 2.10: Arquitectura VGG-19.

2.12 Arquitectura ResNet

ResNet (Red Residual) en lugar de esperar que cada capa apilada se ajuste directamente a un mapeo subyacente deseado, dejamos explícitamente que estas capas se ajusten a un mapeo residual. Formalmente, al denotar el mapeo subyacente deseado como $H(x)$, dejamos que las capas no lineales apiladas se ajusten a otro mapeo de $F(x) = H(x) - x$. El mapeo original se vuelve a convertir en $F(x) + x$. Nuestra hipótesis es que es más fácil optimizar el mapeo residual que optimizar el mapeo original sin referencia. Al extremo, si un mapeo de identidad fuera óptimo, sería más fácil llevar el residuo a cero que ajustar un mapeo de identidad mediante una pila de capas no lineales. La formulación de $F(x) + x$ puede realizarse mediante redes neuronales de retroalimentación con "conexiones de atajo". Las conexiones de acceso directo son aquellas que omiten una o más capas. En nuestro caso, las conexiones de acceso directo simplemente realizan el mapeo de identidad y sus salidas se agregan a las salidas de las capas apiladas [16].

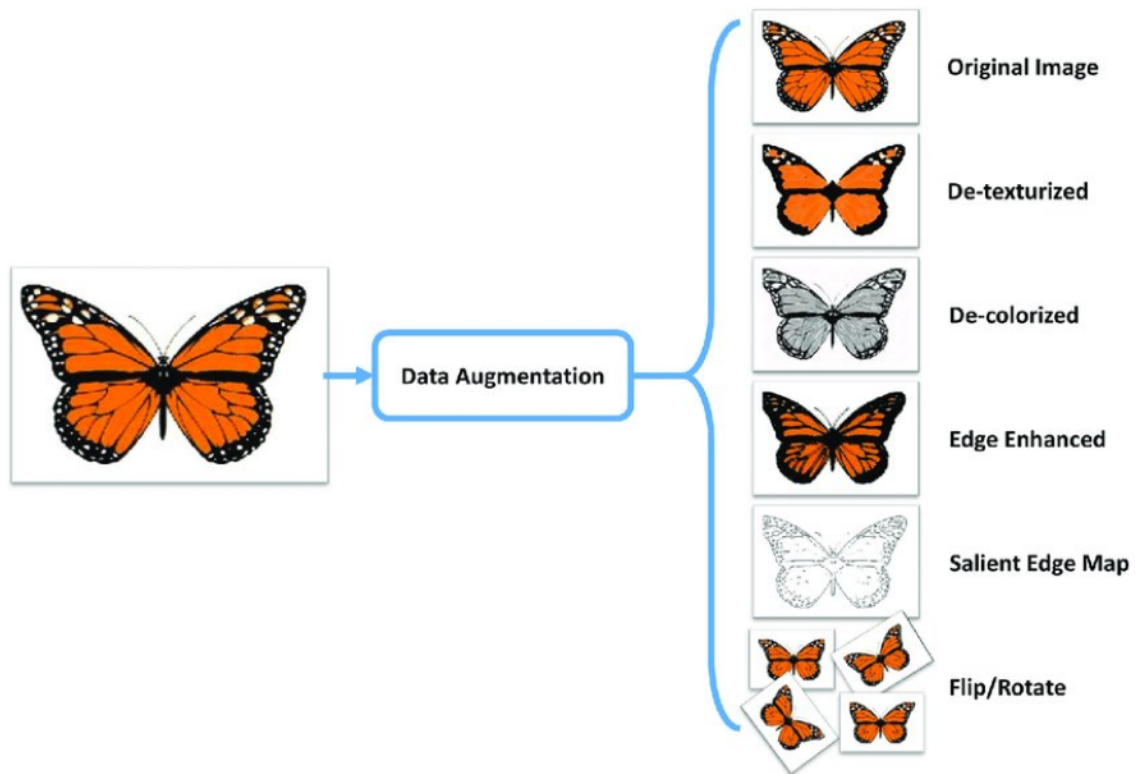


Figura 2.12: Método Data Augmentation.

entrenamiento. En comparación con el aumento de la fase de entrenamiento, el aumento de la fase de prueba se compone de transformación de imagen y fusión de características. La transformación de imágenes transforma una imagen de prueba en varias imágenes para la extracción de características. Las características de estas imágenes se fusionan luego en una característica con fusión de características. Si la CNN ya está capacitada, el aumento de fase de prueba puede aumentar aún más la precisión de la clasificación [8].

En la figura 2.12 se muestra el método de data augmentation.

Capítulo 3

Propuesta metodológica

En esta sección, describimos la metodología propuesta para la reconstrucción de objetos utilizando matrices multicámara para aplicaciones 3D y también para la clasificación, medición y color. El diagrama de bloques del método propuesto para la reconstrucción se ilustra en la Figura 3.1, que consta de los siguientes pasos:

3.1 Reconstrucción para inspección Volumétrica

- **PASO 1:** *Captura de Imágen*

En este paso, el escenario 3D observado es capturado por una matriz de cámaras, en la que las imágenes de entrada contienen un objetivo o varios objetivos representados por múltiples vistas. En este trabajo explicamos un ejemplo con imágenes estéreo y nuestro sistema propuesto con múltiples arreglos de cámaras, que procesan cuatro cámaras simultáneas para la captura de imágenes.

- **PASO 2:** *Procesamiento de Imágenes*

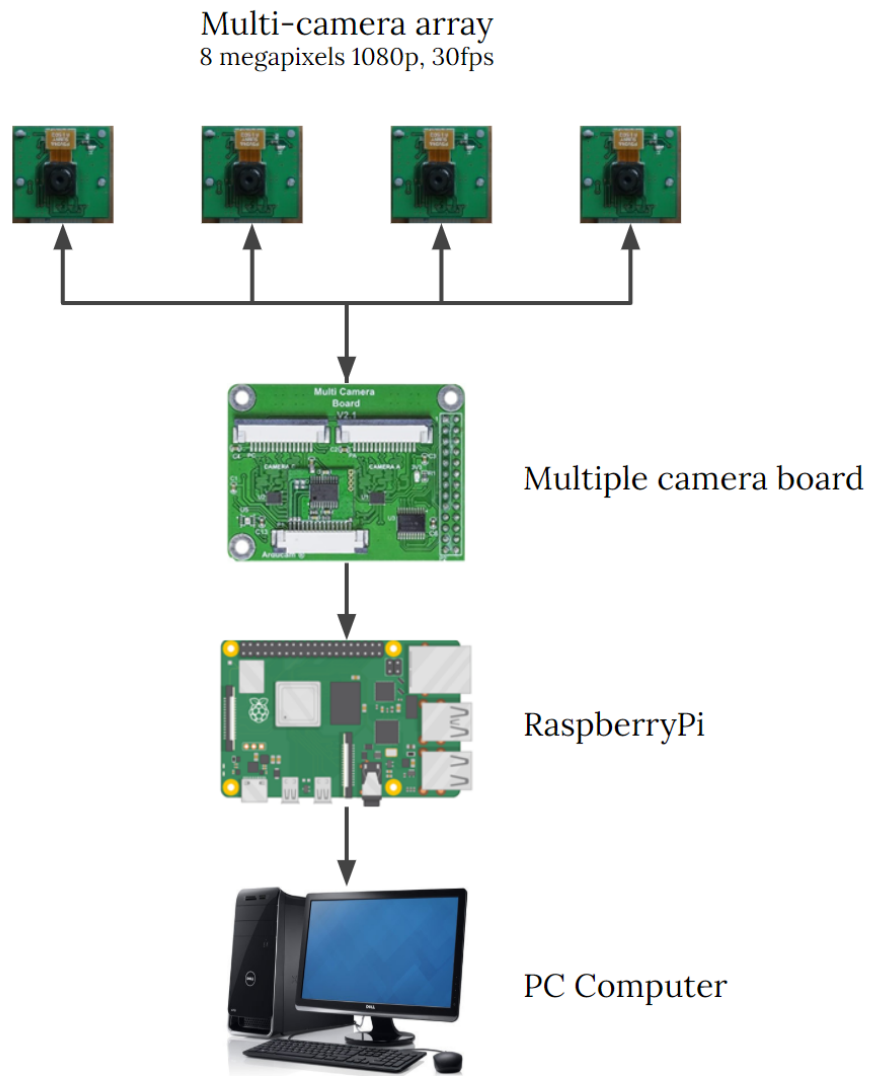


Figura 3.1: Diagrama de bloques del algoritmo propuesto para la reconstrucción de objetos 3D mediante matrices de cámaras.

Debido a que la captura de imágenes se realiza mediante un conjunto de cámaras, se necesita un dispositivo optoelectrónico para digitalizar y almacenar varias imágenes simultáneamente. Para estos, se utiliza una placa electrónica que admite hasta cuatro cámaras para multiplexar la captura y procesamiento de imágenes con la RaspberryPi. En este paso se emplea el filtrado y la alineación de imágenes.

- **Paso 3:** *Reconstrucción de Escena*

Finalmente, las imágenes procesadas del PASO 2 se utilizan para calcular el mapa de disparidad y generar una nube de puntos resultante para el algoritmo de reconstrucción 3D. Este paso produce la visualización de la escena reconstruida dada por múltiples vistas del objetivo utilizando matrices de cámaras.

Supongamos un ejemplo, cuando las imágenes de entrada consisten en un par estéreo, la reconstrucción se puede realizar calculando la disparidad estéreo. Primero, es necesario calcular un algoritmo de coincidencia, como el algoritmo de coincidencia semiglobal (SGBM) [11, 12]. Luego, el mapa de profundidad se obtiene por suma de diferencias absolutas (SAD) entre ambas imágenes. Una imagen de reproyección se convierte en información 3D que produce los puntos resultantes con un espacio de coordenadas tridimensional y un canal de tres colores Rojo-Azul-Verde (RGB). Un algoritmo resumido de la reconstrucción de una escena en 3D a partir de imágenes estéreo se representa mediante el algoritmo 1. Este algoritmo produce un archivo de salida PLY al convertir los datos finales, que contienen un conjunto de puntos, cada uno con 6 parámetros: 3 columnas para coordenadas en el espacio 3D y 3 columnas para la información RGB de cada canal de color.

Algoritmo 1: Reconstrucción a partir de imágenes estéreo

Entrada: Imágenes de entrada I_1, I_2 , forma de la imagen $imshape$, and longitud focal f
Salida: Nube de puntos de la reconstrucción de escena `outputfile.ply`
 $window \leftarrow$ Definir el tamaño de la ventana
 $mindisp \leftarrow$ Valor mínimo de disparidad
 $stereo \leftarrow$ Algoritmo de coincidencia por computadora($window, mindisp$)
 $disp \leftarrow$ Disparidad estéreo de computadora(I_1, I_2)
 $\mathbf{Q} \leftarrow \mathbf{Q}$ matriz
 $points \leftarrow$ Reproyectar imagen a 3d($disp, \mathbf{Q}$)
 $mask \leftarrow disp > min(disp)$
 $points \leftarrow$ Calcular puntos de color($mask$)
 $data \leftarrow$ Generar archivo de salida PLY

El método actual introduce una captura de imagen simultánea de cuatro imágenes con diferentes vistas de la escena. Un conjunto de puntos clave entre imágenes se calcula mediante un algoritmo de coincidencia de características [18, 21] que produce la estimación de matriz fundamental [7, 23]. La pose de la cámara se puede estimar calculando la matriz fundamental y los puntos clave. Aquí, la estimación se puede optimizar minimizando el residuo del vector de reproyección. Luego, los gráficos generados a partir de puntos coincidentes se pueden triangular y ajustar para fusionar todos los gráficos. El pseudocódigo resumido de una reconstrucción de escena 3D a partir de matrices de varias cámaras se describe en Algoritmo 2. Como podemos ver, el algoritmo presentado produce un procesamiento de reconstrucción de gráficos denso para generar datos de nube de puntos de salida con coordenadas 3D y canal de color RGB en formato PLY.

Algoritmo 2: Reconstrucción a partir de una matriz multicámara

Entrada: Imágenes de entrada I_1, I_2, I_3, I_4 , Resolución Máxima $maxsize$, and longitud focal f
Salida: Nube de puntos de la reconstrucción de la escena `outputfile.ply`
 $graphlist \leftarrow$ Lista de imágenes de entrada
for *longitud de la lista de imágenes* **do**
 $keypoints \leftarrow$ Calcular algoritmo SIFT
 $\mathbf{F} \leftarrow$ Calcular matriz fundamental($keypoints$)
 $\mathbf{K} \leftarrow$ Compute Matrix K
 $\mathbf{E} \leftarrow \mathbf{K}^T \mathbf{F}_0 \mathbf{K}$
 $campose \leftarrow$ Postura de cámara computarizada($\mathbf{E}, \mathbf{K}, keypoints$)
 $graphlist \leftarrow$ Generación de gráficos por computadora($campose, \mathbf{K}, keypoints, f$)
 $graphtriang \leftarrow$ Triangulación de malla informática($graphlist$)
 $graphadjust \leftarrow$ Ajuste del paquete de computadora($graphtriang$)
end
 $graphmerge \leftarrow$ Calcular gráficos de combinación
for *longitud de la lista de imágenes* **do**
 $graphlist \leftarrow$ Procesar gráficos densos(lista de imágenes, $maxsize$)
end
for *longitud de la lista de imágenes* **do**
 $graphlist \leftarrow$ Proceso de reconstrucción densa($graphmerge, \mathbf{K}, maxsize$)
end
 $data \leftarrow$ Generar archivo de salida PLY

3.2 Clasificación de forma en producto agrícola

Para la clasificación del pepino describimos la propuesta metodológica, el diagrama a bloques propuesto para la clasificación se ilustra en la Figura 3.2, que consta de los siguientes pasos:

- **PASO 1: Imágenes de Entrada**

En este paso, las imágenes son extraídas de la nube de computo, en la que las imágenes de entrada contienen un objetivo que es normalizar cada imagen en tamaño ($W=500$, $H=140$ pixeles) y en valores (0 a 1) donde 0 es recto y 1 es curvo, de esta manera se cargan los datos por clase: Curva y no curva.

- **PASO 2: Aumento de Datos**

Se define una función para crear diferentes versiones del conjunto original (Aumento de datos) esto se hace para tener mas cantidad de imágenes, se aumenta los datos para alcanzar cada clase con 2mil imágenes, después se concatena el conjunto de datos original y aumentado y hacerlo en 2D.

- **PASO 3: Modelo de Red**

Se Construye el modelo de la red, en donde se separa en 20% probar (test) y 80% entrenar (train), después se entrena con el conjunto de imágenes de prueba para tener una buena validación y por último se hacen predicciones(saber si es curvo o no curvo) y grafican los resultados.

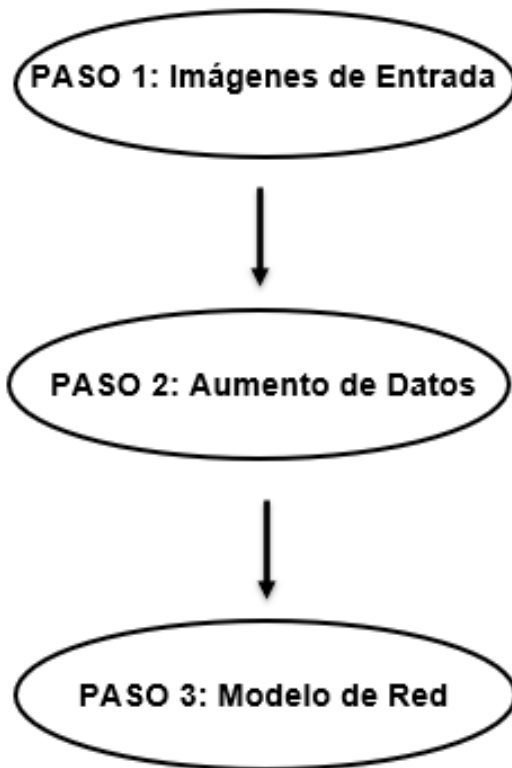


Figura 3.2: Diagrama de bloques del algoritmo propuesto para la clasificación del pepino.

3.3 Medición del Tamaño y Color de productos agrícolas

Los atributos de tamaño y color son de suma importancia para la inspección de productos agrícolas. Algunos ejemplos se muestran en la Figura 3.3 en la cual se puede observar las imágenes del pepino que pasan por una banda transportadora. Algunos de ellos son afectados por iluminación lo cual provoca que la medición del color no sea precisa.



Figura 3.3: Ejemplos de visualización del pepino en la inspección de calidad mediante tamaño y color.

Se diseña una metodología con técnicas de visión artificial para automatizar la medición del pepino dentro de la banda transportadora para incrementar la calidad de inspección, Estos pasos son diseñados para la medición del tamaño y color del pepino, donde describimos la propuesta metodológica, el diagrama a bloques propuesto para la Medición de Tamaño y color se ilustra en la Figura 3.4, que consta de los siguientes pasos:

- **PASO 1: Imágenes de Entrada**

En este paso, las imágenes son extraídas de la nube de computo, se lee conjunto de datos para la medición dimensional del pepino, luego se binariza imagen (escala a grises valores de 0 a 255) y se hace buscar objeto (para esto se toma en base la medición de la banda que son 18 pulgadas).

- **PASO 2: Medición Pulgadas**

Se hace la medición de ancho y largo del pepino en pixeles y en pulgadas, donde mediante la medida de la banda se hace la medición.

- **PASO 3: Medición Color**

Por último se hace la medición de color en RGB y se gráfica.

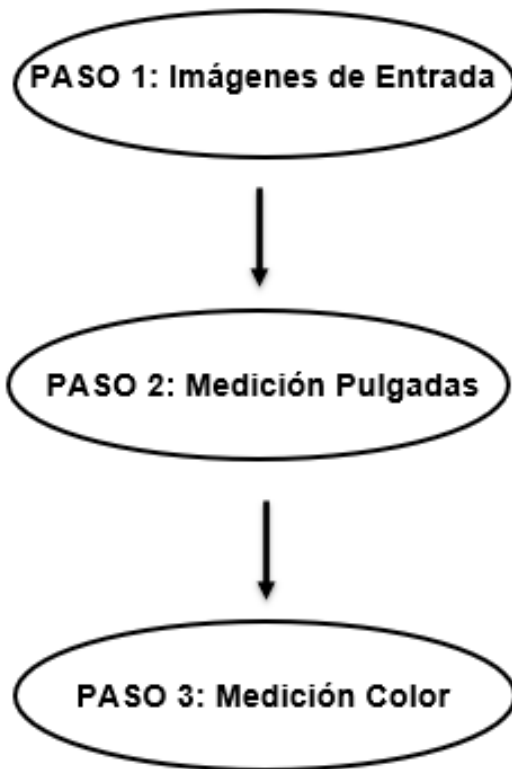


Figura 3.4: Diagrama de bloques del algoritmo propuesto para la Medición del tamaño y color del pepino.

Capítulo 4

Resultados

En esta sección, presentamos y discutimos los resultados obtenidos de la metodología propuesta para la reconstrucción de objetos 3D utilizando arreglos multicámara y también para la clasificación, medición y color del pepino. Se diseña un algoritmo que permite la reconstrucción tridimensional de un objeto contenido en imágenes estéreo, utilizando los principios de la visión, de esta manera también se diseña el algoritmo que nos permite la clasificación, medición y color del pepino utilizando principios de redes.

4.1 Resultados Experimentales con Imágenes Estéreo

El sistema desarrollado realiza su trabajo en dos etapas: En la primera, se prepara el entorno para la reconstrucción 3D, capturando las imágenes y rectificando las imágenes. En el segundo, la reconstrucción tridimensional se realiza simulando y visualizando el modelo de objeto, utilizando algoritmos de coincidencia de correlación cruzada y triangulación estéreo. Las imágenes estéreo izquierda y derecha se muestran en la Fig. 4.1 (a) y Fig. 4.1 (b), respectivamente. El par de imágenes tiene una diferencia longitudinal en la posición de la

cámara dada por la suma de diferencias absolutas (SAD) dada por [25]:

$$SAD(x, y, d_x, d_y) = \sum_{i=0}^W \sum_{j=0}^W \left| I_L(x + i, y + j) - I_R((x + d_x) + i, (y + d_y) + j) \right|, \quad (4.1)$$

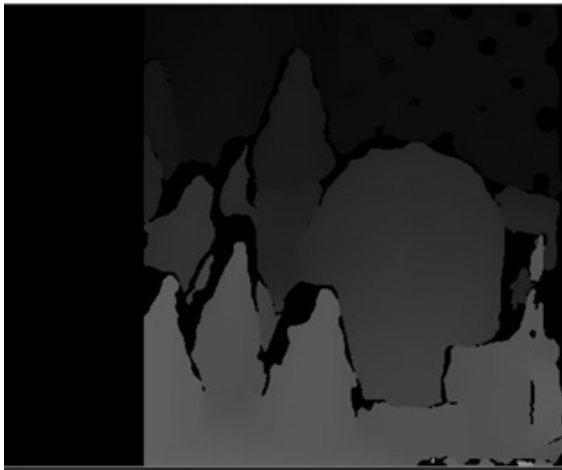
donde (x, y) es la posición actual del píxel, (d_x, d_y) es el desplazamiento entre las imágenes izquierda I_L y derecha I_R evaluar en W que representa una ventana cuadrada [9]. Para la implementación se desarrolla un programa en Python, en el cual se resume en Algoritmo 1. El programa produce una reconstrucción de la disparidad estéreo. Después de ejecutar el programa con las imágenes estéreo, el programa produce un archivo de salida en formato PLY, que forma una reconstrucción 3D simplificada como se muestra en la Fig. 4.1 (c). El archivo de salida podría visualizarse en el programa de código abierto Meshlab [6] para que la visualización de nuestro procesamiento de las imágenes estéreo se muestra a continuación en la Fig. 4.1 (d). El resultado es la estimación de la profundidad a partir de solo dos imágenes (par estéreo), el desafío sería convertirlo en múltiples imágenes para reconstruir la mayor superficie posible.



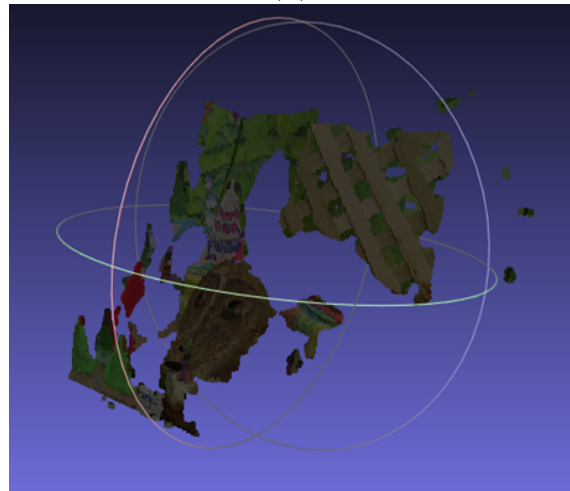
(a)



(b)



(c)



(d)

Figura 4.1: Resultados experimentales de imágenes estéreo. (a) Par de imágenes izquierda y (b) derecha de una cámara estéreo. (c) Imagen de profundidad dada por disparidad estéreo, y (d) Reconstrucción de escena en archivo de nube de puntos.

4.2 Resultados Experimentales con Arreglos Multicámara

El sistema desarrollado realiza una reconstrucción de la escena dada a partir del sistema de arreglo multicámara propuesto, como se muestra en la Figura 3.1. Primero, el sistema captura simultáneamente cuatro vistas diferentes de la escena proporcionadas por el conjunto de cámaras. Como se ilustra en la Fig. 4.2, usamos cuatro imágenes de entrada, cada una representa una vista diferente de la escena. Las imágenes contienen varios objetos con diferente textura, material y color.

Implementamos el algoritmo 2 en un código de programación Python. El programa produce una reconstrucción precisa a partir de las imágenes de entrada dadas en una nube de puntos como un archivo de salida en formato PLY. En la Fig. 4.3 la nube de puntos de salida se ilustra utilizando el programa de visualización de código abierto Meshlab [6]. Como podemos ver en la Fig. 4.3 (a) y Fig. 4.3 (b), la nube de puntos resultante de la reconstrucción de la escena se puede comparar antes de la optimización y después de la optimización, respectivamente, en el que los resultados optimizados producen una salida más limpia y ordenada de los puntos atípicos, y también la densidad de puntos se enfoca mejor en los objetos presentados en la escena. En la Fig. 4.4 se muestra una representación de diferentes vistas de los datos de la nube de puntos. La nube de puntos resultante se almacena en el archivo PLY de salida proporcionado por la reconstrucción de la escena del punto 3D y sus datos con la representación del canal de color RGB de la escena.



Figura 4.2: Imágenes de ejemplo para la reconstrucción de escenas I_1, I_2, I_3 y I_4 capturadas desde el sistema de matriz multicámara propuesto.

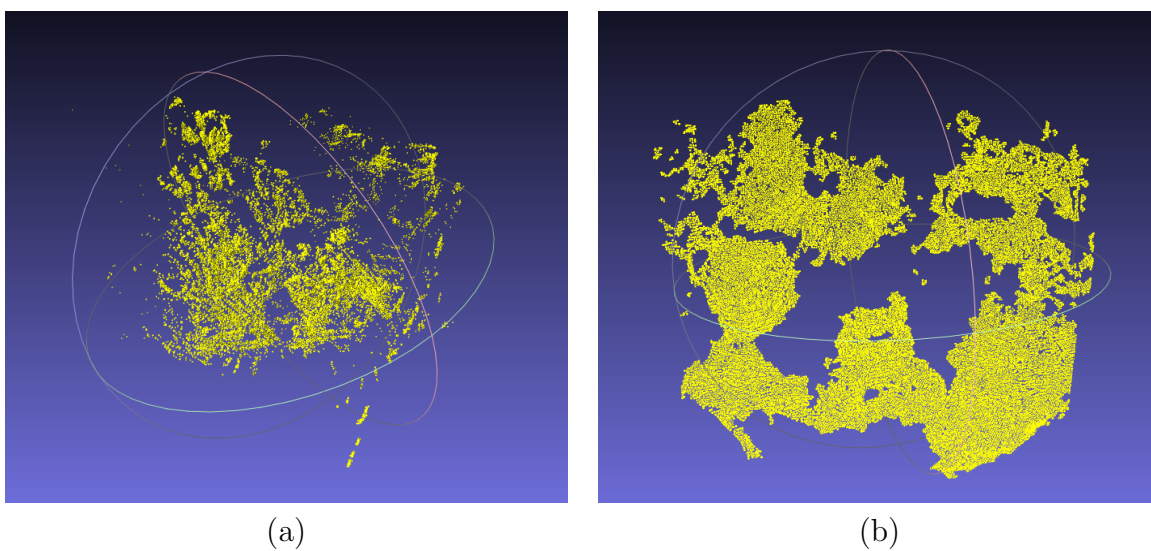


Figura 4.3: Malla resultante de la reconstrucción de la escena (a) antes del proceso de optimización y (b) después del refinamiento del proceso de optimización.

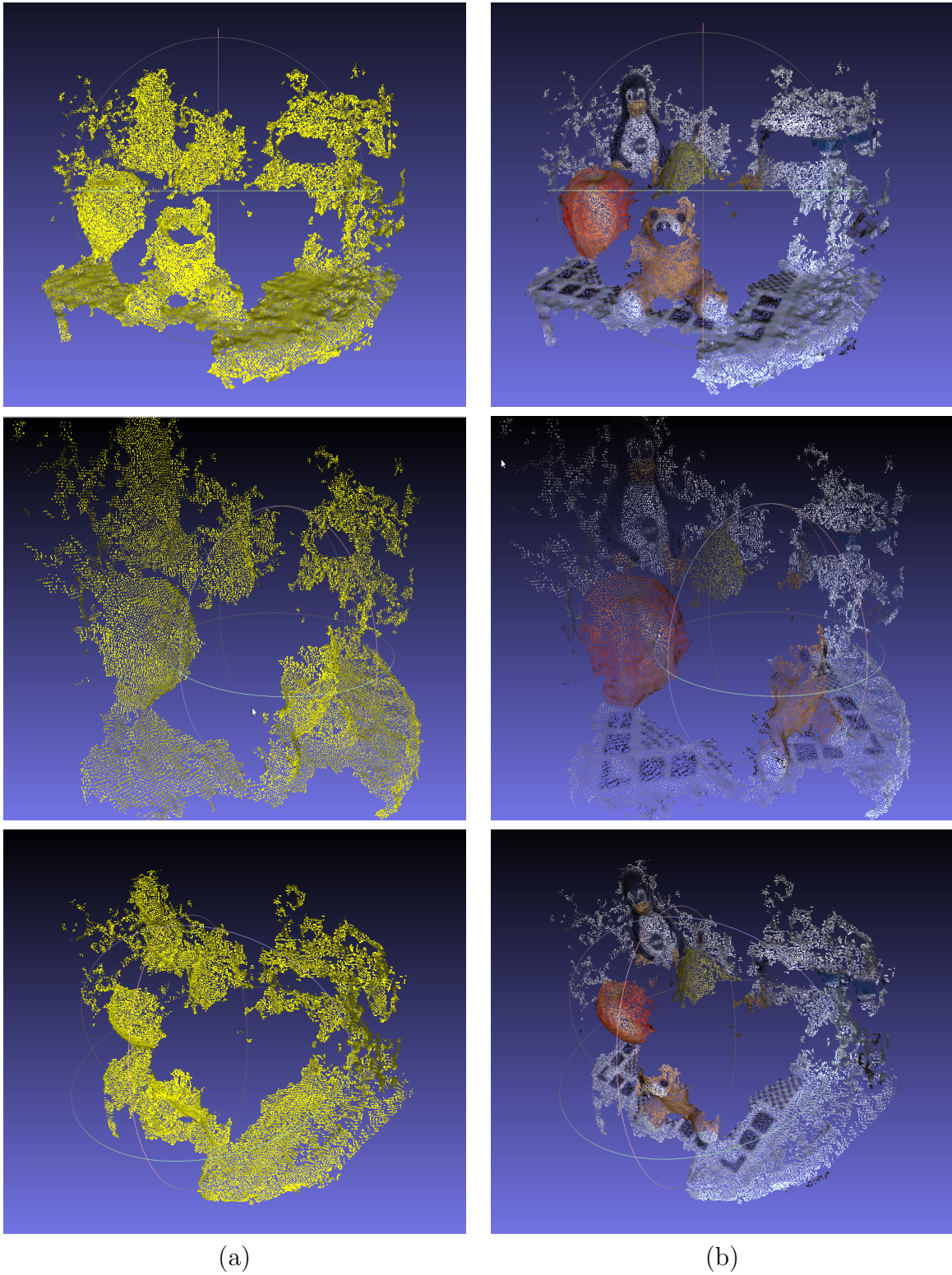


Figura 4.4: Resultados preliminares de la reconstrucción de escenas en 3D a partir de una matriz de cámaras. (a) Datos de salida de la nube de puntos y (b) nube de puntos con representación del color de la escena.

4.3 Resultados Experimentales para Clasificación del Pepino

El sistema desarrollado realiza una clasificación a partir del escenario que es las imágenes son extraídas de la nube de computo, como se muestra en la figura 3.2, primero, el sistema captura imágenes mediante la nube de computo, contiene las imágenes de entrada con el objetivo de normalizar el tamaño en valores 0 y 1, donde los datos se cargan por clase curva y no curva, como se muestran en la figuras 4.5 y 4.6.

De esta manera se define la función para crear diferentes versiones de las imágenes originales (Aumento de datos), como se muestran en las figuras 4.7 y 4.8, para esto se aumentan los datos para alcanzar a 2,000 imágenes curvas y no curvas para tener en total 4,000 imágenes, luego se concatena el conjunto de datos original y aumentado y hacerlo 2D.

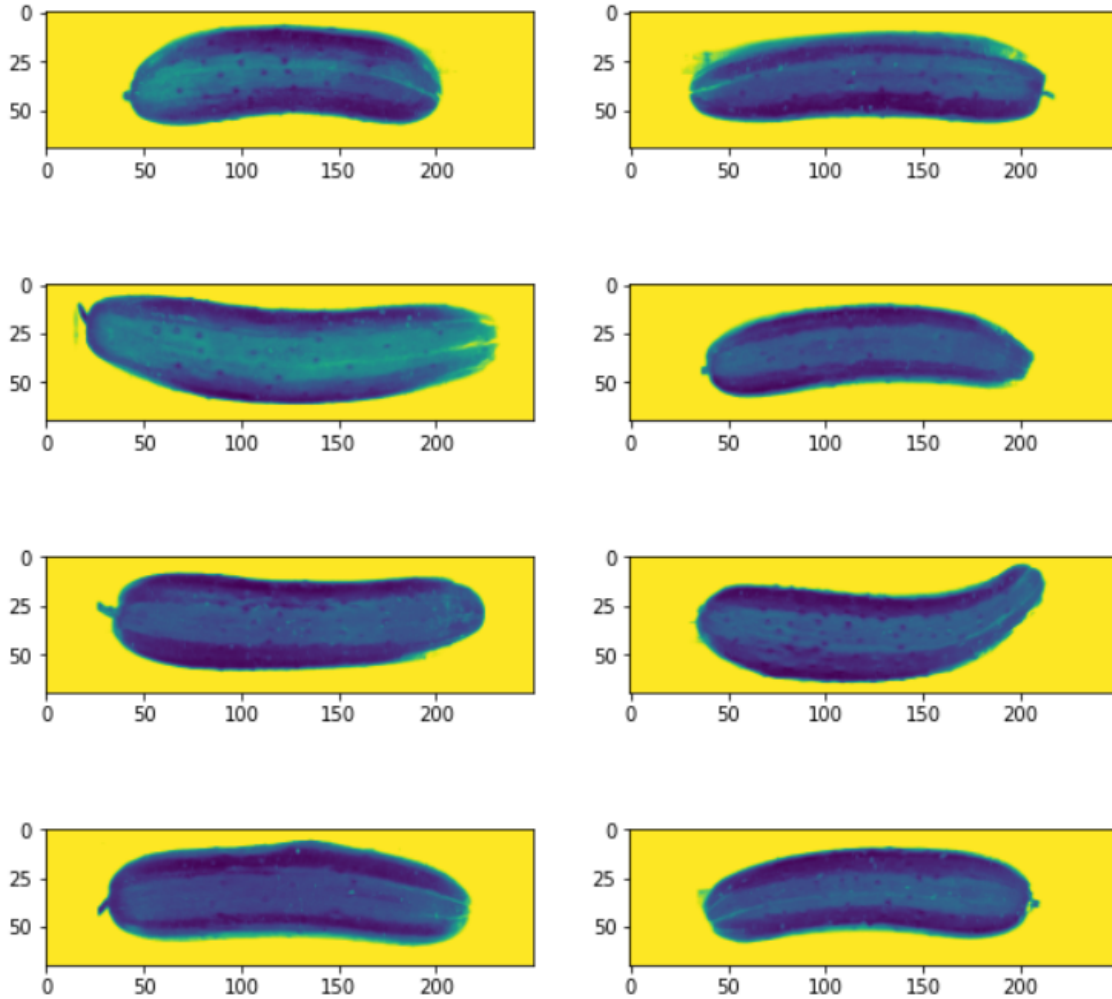


Figura 4.5: Diferentes ejemplos de imágenes de pepino para la clase Curva.

De esta manera, se construye el modelo de la red donde se separan por Prueba (Test) y Entrenamiento (Train), aquí mismo se nota la cantidad como se separan en 20% Prueba (Test) y 80% Entrenamiento (Train) como se muestra en la figura 4.9, para esto se hace un conjunto de imágenes de prueba para tener predicciones y graficas de resultados como se muestra en la figura 4.10.

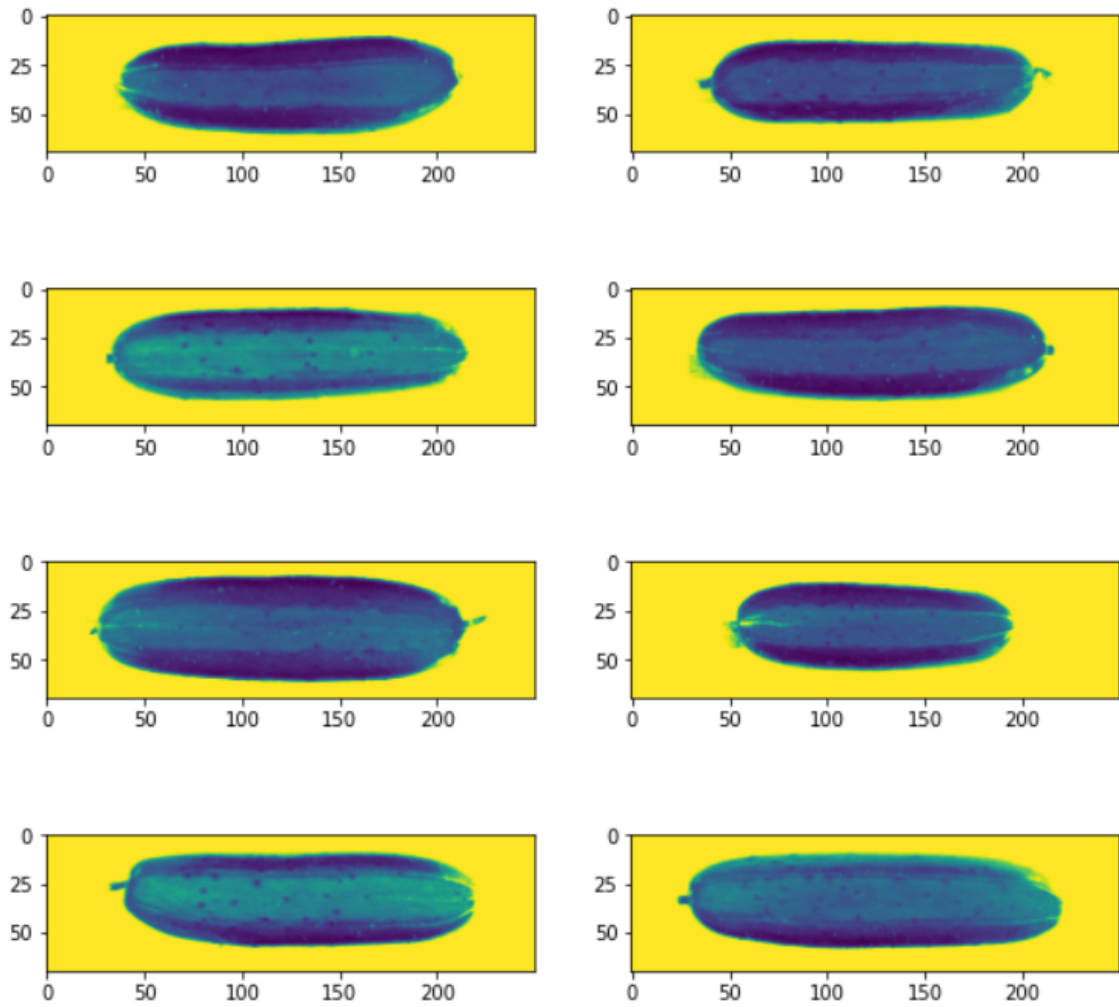


Figura 4.6: Diferentes ejemplos de imágenes de pepino para la clase No Curva.

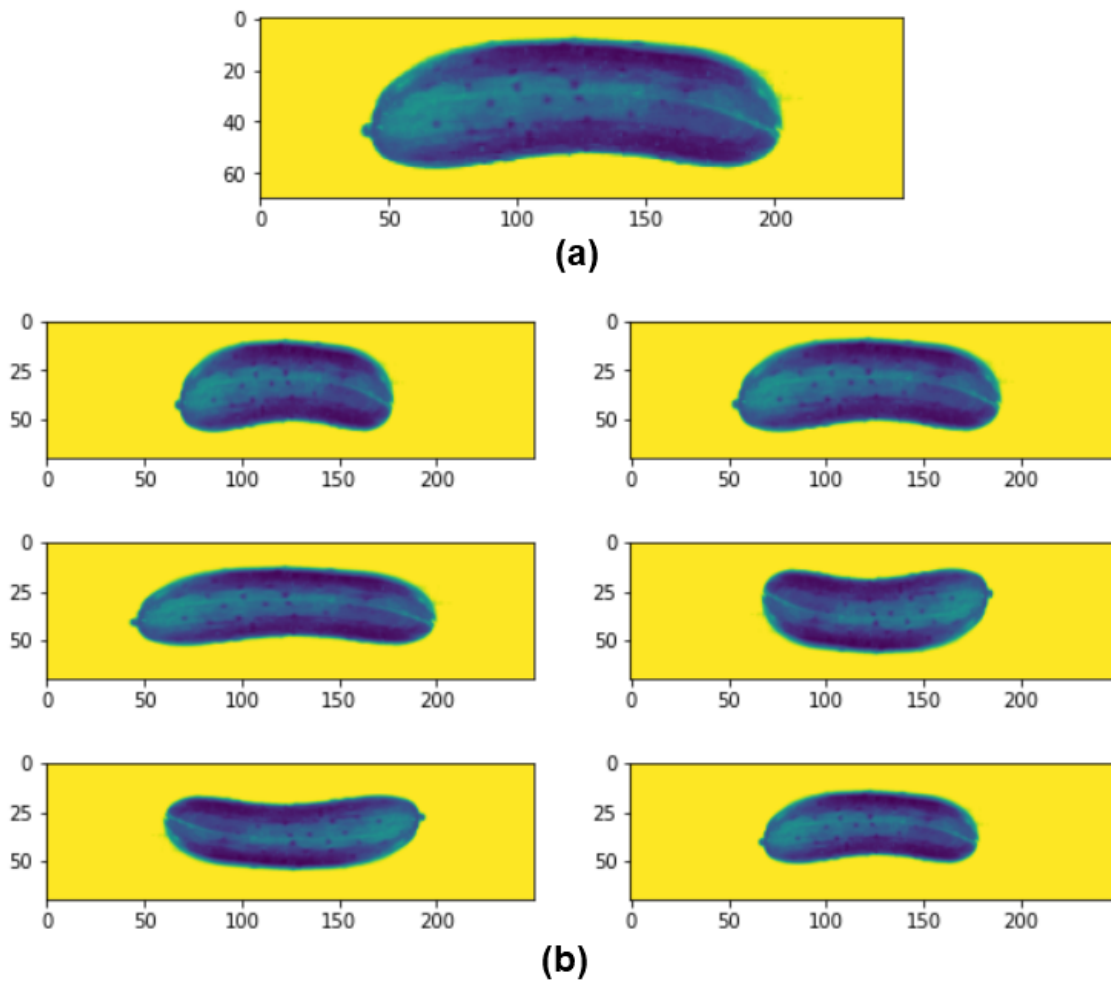


Figura 4.7: Aumento de datos de la clase Curva a partir de una entrada de imagen original con funciones aleatorias para aplicar el reflejo horizontal, reflejo vertical, y escalamiento. (a) Imagen original. (b) Imágenes aumentadas.

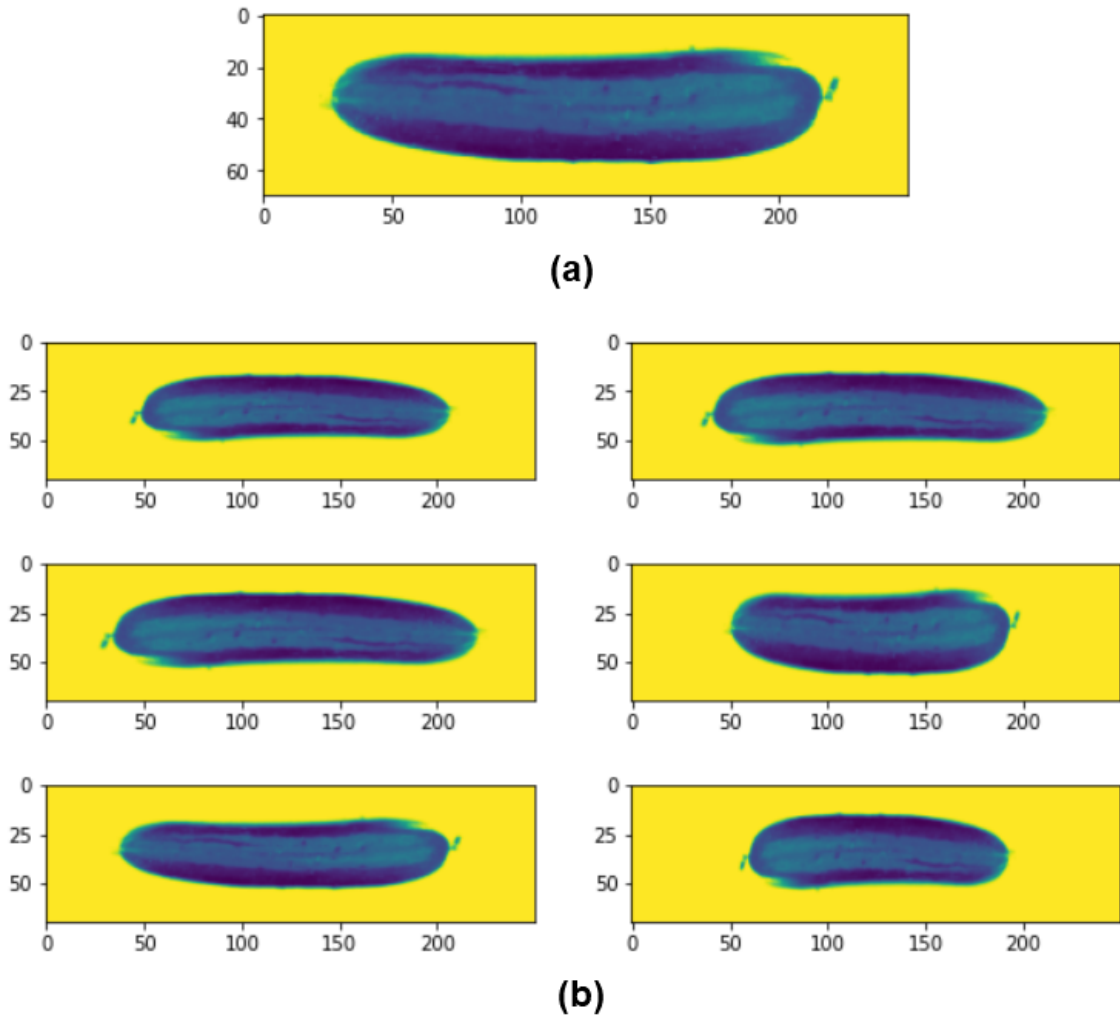


Figura 4.8: Aumento de datos de la clase No Curva a partir de una entrada de imagen original con funciones aleatorias para aplicar el reflejo horizontal, reflejo vertical, y escalamiento. (a) Imagen original. (b) Imágenes aumentadas.

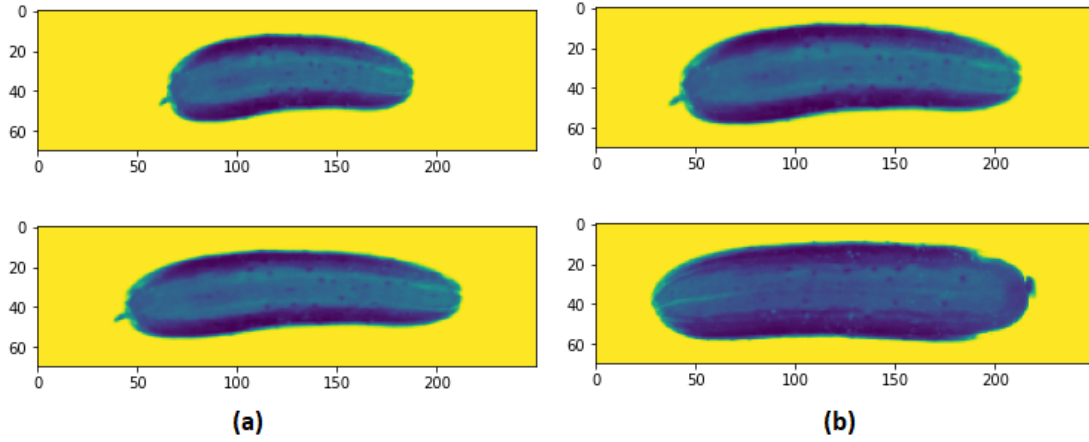


Figura 4.9: Ejemplo de Imágenes (a) Entrenamiento y (b) Prueba.

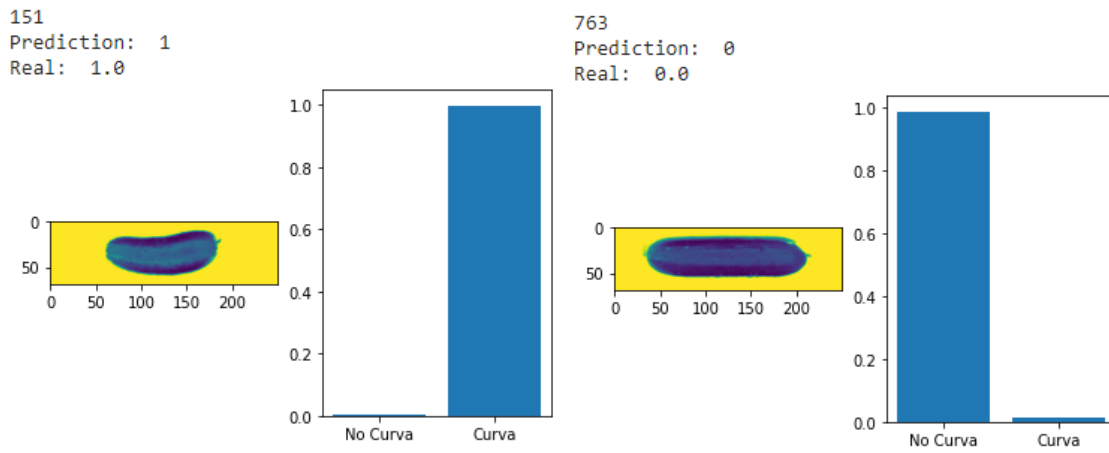


Figura 4.10: Se muestra el resultado de las predicciones cuando es curva y no curva.

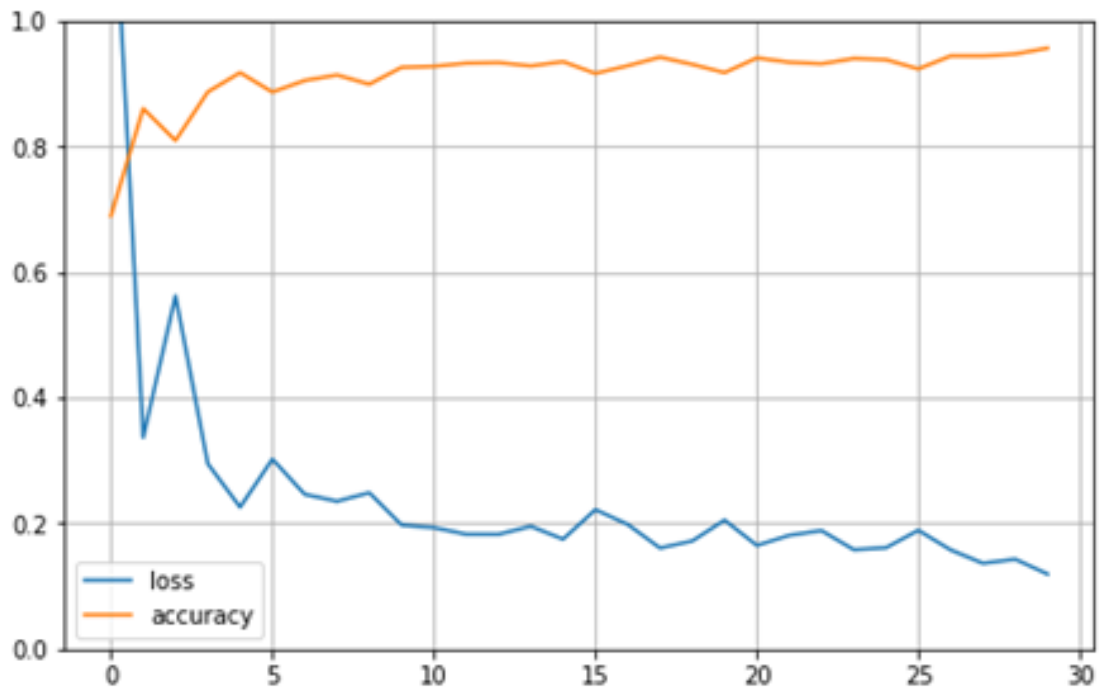


Figura 4.11: Se muestra el resultado de la gráfica de 3 capas, donde se ve la Precisión y función de perdida.

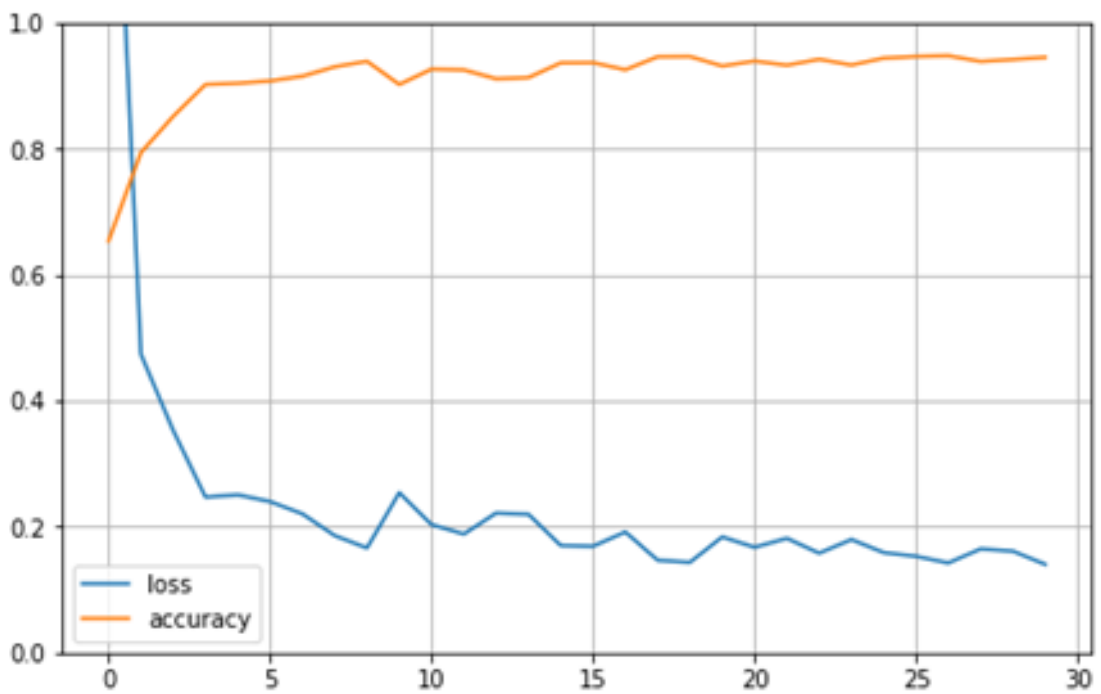


Figura 4.12: Se muestra el resultado de la gráfica de 5 capas, donde se ve la Precisión y función de perdida

4.4 Resultados Experimentales para la Medición y Color del Pepino

El sistema desarrollado realiza una clasificación a partir del escenario que es las imágenes son extraídas de la nube de computo como se muestra en la figura 3.4, primero, el sistema captura imágenes mediante la cámara cognex con una buena iluminación, el programa contiene las imágenes donde lee el conjunto de datos para la medición dimensional del pepino como se muestra en la figura 4.13, después binariza la imagen y busca el objeto, luego mediante los valores de una banda se hace la conversión de pixeles y pulgadas para obtener el ancho y largo del pepino como se ve en la figura 4.14, y por ultimo se hace la medición de color con un histograma como se ve en la figura 4.15.

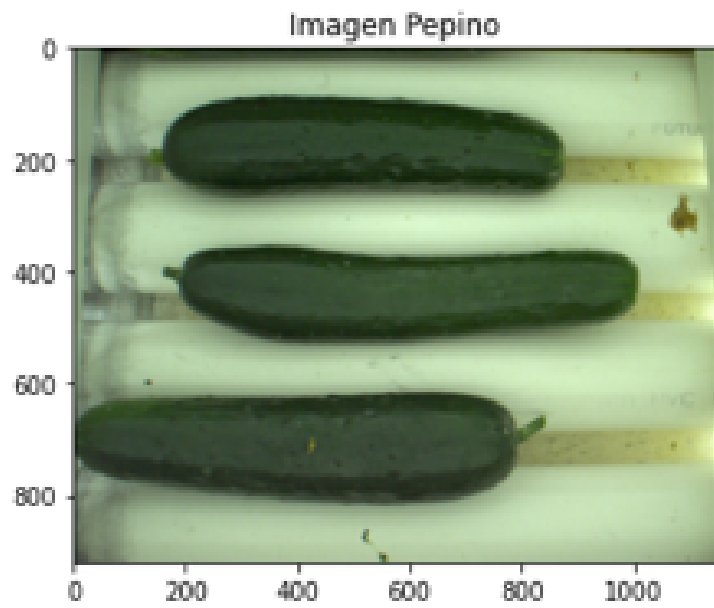


Figura 4.13: Muestra de imagen de entrada.

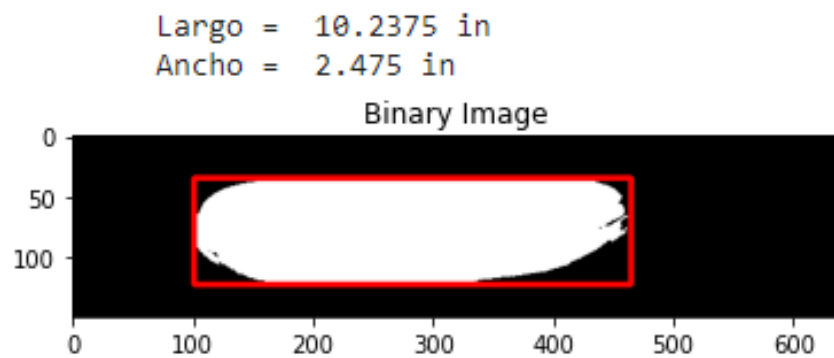


Figura 4.14: Resultado de la medición automática del pepino mediante los pixeles a pulgadas.

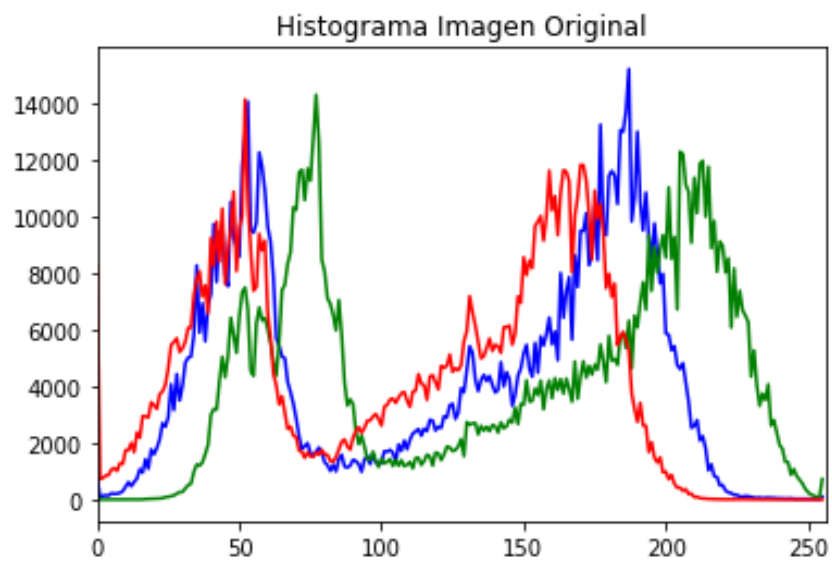


Figura 4.15: Resultado de la medición de imagen por cada canal de color RGB.

Capítulo 5

Conclusiones

Se desarrolla un algoritmo para la inspección de calidad automatizada de productos agrícolas mediante visión artificial. El algoritmo propuesto se probó en escenarios reales con varias condiciones de imagen, como desorden de fondo, luz y múltiples vistas de objetos de prueba. La metodología actual es confiable para implementar en aplicaciones reales para imágenes tridimensionales. El sistema automático de visión artificial en productos agrícolas, tales como el pepino, para clasificación de características como: Longitud de área, forma y defectos para la inspección de calidad, nos ayudo detectar de manera correcta para la toma de decisiones como se ve en los resultados por parte de la calidad si es curvo o no curvo, y por la parte de pulgadas para lo ancho y largo del pepino asi mismo el color se mostro en la gráfica de una manera satisfactoria. Este algoritmo presenta una propuesta eficiente para la inspección de calidad automática en la clasificación del pepino la cual se puede implementar para inspeccionar cualquier fruta y verdura.

Bibliografía

- [1] G. E. H. Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60, 2017.
- [2] K. B. Ali. Determination of watermelon volume using ellipsoid approximation and image processing. *Postharvest Biology and Technology*, 38:366–371, 2007.
- [3] J. Báez-Rojas and M. Alonso-Pérez. Machine Vision System for Automatic Quality Grading of Fruit. *Biosystems Engineering*, 85:415–423, 2003.
- [4] J. Báez-Rojas and M. Alonso-Pérez. Uso del sistema HSI para asignar falso color a objetos en imágenes digitales. *Revista Mexicana de Física*, 54, 2008.
- [5] H. Cen, R. Lu, D. P. Ariana, and F. Mendoza. Hyperspectral Imaging-Based Classification and Wavebands Selection for Internal Defect Detection of Pickling Cucumbers. *Food Bioprocess Technol*, 7:1689–1700, 2013.
- [6] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In V. Scarano, R. D. Chiara, and U. Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.
- [7] O. Faugeras, Q.-T. Luong, and T. Papadopolou. *The Geometry of Multiple Images: The Laws That Govern The Formation of Images of A Scene and Some of Their Applications*. MIT Press, Cambridge, MA, USA, 2001.
- [8] Z. X. Gao W. Auto data augmentation for testing set. *Pattern Recognition and Computer Vision*, 2019.
- [9] R. A. Hamzah, R. A. Rahim, and Z. M. Noh. Sum of absolute differences algorithm in stereo correspondence problem for stereo matching in computer vision application. In *3rd International Conference on Computer Science and Information Technology*, volume 1, pages 652–657, 2010.
- [10] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [11] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

- [12] H. Hirschmüller, M. Buder, and I. Ernst. Memory efficient semi-global matching. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3:371–376, 2012.
- [13] H. Hu, J. Luo, G. Hu, W. Duan, and H. Zhou. Generación de nubes de puntos 3D usando estructura incremental a partir del movimiento. *Conf. Series*, 1087, 2018.
- [14] S. Jana, R. Parekh, and B. Sarkar. Automatic classification of fruits and vegetables: A texture-based approach. *Studies in Computational Intelligence*, 870, 2020.
- [15] J.A.Throop, D.J.Aneshansley, W.C.Anger, and D.L.Petersonb. Quality evaluation of apples based on surface defects: Development of an automated inspection system. *Postharvest Biology and Technology*, 36:281–290, 2005.
- [16] S. R. J. S. Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [17] A. Z. Karen Simonyan. Deep residual learning for image recognition. *International Conference on Learning Representations*, 2015.
- [18] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.
- [19] E. Moltó, S. Cubero, N. Aleixos, J. Gómez, and J. Blasco. Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables. *Food and Bioprocess Technology*, 2010.
- [20] K. Qing. Study on the detecting and grading method of apples with machine vision. *Beijing: China Agricultural University, Master Dissertation.*, 2007.
- [21] P. P. Ramya and J. Ajay. Object recognition and classification based on improved bag of features using surf and msr local feature extraction. In *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, pages 1–4, 2019.
- [22] O. I. Rosario. Sistema para clasificación de jitomates basado en metrología laser y algoritmos computacionales. *Centro de Investigaciones en Optica, A.C.*, 2010.
- [23] N. Tatar and H. Arefi. Stereo rectification of pushbroom satellite images by robustly estimating the fundamental matrix. *International Journal of Remote Sensing*, 40(23):8879–8898, 2019.
- [24] D. Unay and B. Gosselin. Stem and calyx recognition on jonagold apples by pattern recognition. *Journal of Food Engineering*, 78:597–605, 2007.
- [25] S. Vassiliadis, E. A. Hakkennes, J. S. Wong, and G. G. Pechanek. The sum-absolute-difference motion estimation accelerator. In *Proceedings. 24th Euromicro Conference (Cat. No.98EX204)*, volume 2, pages 559–566, 1998.

- [26] H. Wenjun. Method of apple automatic grading based on machine vision. *Nanjing: Nanjing Forestry University, Master Dissertation.*, 2006.
- [27] L. M. Xu and Y. C. Zhao. Automated strawberry grading system based on image processing. *Computers & Electronics in Agriculture*, 71:32–39, 2003.
- [28] Y. Y., R. X., and H. Y. Controller for real-time sorting mechanism of fruit. *Transactions of the Chinese Society for Agricultural Machinery*, 35:117–121, 2004.
- [29] Y. B. P. H. Yann LeCun, Léon Bottou. Gradient based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998.
- [30] K. Zhang, X. Chen, and H. Wang. Research on external quality inspection technology of tropical fruits based on computer vision. *Springer Proceedings in Business and Economics*, 2018.