**Desarrollo de un modelo para la detección y clasificación de enfermedades en la planta del Tomate utilizándo Swin Transformer**

Oscar David Valencia - 20192182074

Maria José Capera Firigua – 20201188262

Facultad de Ingeniería, Universidad Surcolombiana

Electiva Ciencias de la computación II - Visión Artificial

Ferley Medina Rojas

Agosto de 2023

**TABLA DE CONTENIDOS**

[**INTRODUCCIÓN 3**](#_heading=h.gjdgxs)

[**PROBLEMA 4**](#_heading=h.30j0zll)

[**OBJETIVOS 5**](#_heading=h.4d34og8)

[Objetivo general 5](#_heading=h.1fob9te)

[Objetivos especificos 5](#_heading=h.3znysh7)

[**REVISIÓN DE LITERATURA 6**](#_heading=h.2et92p0)

[**MARCO TEÓRICO 17**](#_heading=h.4d34og8)

[**REFERENCIAS 22**](#_heading=h.17dp8vu)

# INTRODUCCIÓN

La detección temprana y precisa de enfermedades en el tomate se ha convertido en una necesidad imperante para mitigar su impacto devastador. Aunque los avances en la visión artificial y el aprendizaje automático han proporcionado soluciones potenciales, estos métodos han sido insuficientes en términos de velocidad, costo, y precisión. Las prometedoras técnicas de Deep Learning, particularmente las redes neuronales convolucionales (CNN por sus siglas en inglés), no han vuelto a plantear nuevas arquitecturas ni tampoco un cambio de paradigma, lo que exige un replanteamiento innovador.

Este estudio se propone explorar y validar un enfoque innovador mediante la aplicación del "Swin Transformer", una arquitectura novedosa en el dominio de la visión artificial. A través de un análisis sistemático y una metodología rigurosa, se buscará desarrollar un modelo que supere las limitaciones existentes en la detección y clasificación de enfermedades del tomate.

Se enfocará en responder preguntas críticas relacionadas con el desarrollo y validación de este nuevo modelo, así como en su comparación con los métodos basados en CNN. La investigación abordará aspectos esenciales como el aumento de datos, la integridad de los resultados, la configuración y entrenamiento óptimo de la arquitectura Swin Transformer, y la interpretación visual de las decisiones del modelo.

La investigación propuesta tiene un potencial significativo para transformar la práctica actual en la detección y clasificación de enfermedades del tomate, ofreciendo un modelo más preciso y eficiente.

El trabajo se estructura en introducción, problema, objetivos, estado del arte, marco teórico, metodología, materiales y métodos, resultados, conclusiones y trabajo futuro.

# PROBLEMA

El cultivo del tomate es esencial para la alimentación y la economía agrícola global debido a que es una de las hortalizas más producidas mundialmente. De acuerdo con la FAO (2021), la producción mundial superó los 189 millones de toneladas, abarcando más de 5 millones de hectáreas cultivadas. Nutricionalmente es una fuente rica de fitoquímicos, nutrientes, antioxidantes y compuestos fenólicos (Collins et al., 2022). Se ha determinado que su consumo aporta beneficios anticancerígenos y protección contra enfermedades cardiovasculares y neurodegenerativas (Collins et al., 2022).

No obstante, el tomate es vulnerable a una amplia variedad de enfermedades causadas por hongos, bacterias, fitoplasmas, virus y viroides (Panno et al., 2021) Tabla 1. Esta vulnerabilidad se ve incrementada por factores como su limitada diversidad genética resultado de la selección intensiva, la tendencia al monocultivo, el intercambio de material infectado a nivel internacional y el cambio climático (Panno et al., 2021). Estas enfermedades representan pérdidas económicas que superan los 30.000 millones de dólares al año (Caruso et al., 2022). Dichas pérdidas repercuten en la economía agrícola, la salud pública y la sostenibilidad ambiental, al afectar el rendimiento y la calidad del cultivo, desestabilizar ecosistemas y elevar los costos de producción por el uso intensivo de pesticidas y otras medidas de control (Panno et al., 2021).

La detección temprana y precisa de enfermedades en el tomate es primordial para mitigar su impacto, pero a pesar de los avances significativos en visión artificial en la última década, estos han presentado limitaciones. En el ámbito del machine learning, los métodos suelen recurrir a la extracción manual de características y algoritmos heurísticos, lo que puede ser lento, laborioso, costoso y susceptible a errores (Zahangir Alom et al., 2018). Estos enfoques han sido superados por técnicas de Deep Learning como las redes neuronales convolucionales (CNN) (Rawat et al. 2022). Sin embargo, estas últimas parecen haber llegado a un punto de estancamiento tras el desarrollo de arquitecturas como ResNet, DenseNet y EfficientNet (Li, 2020).

Es por ello, que la presente investigación sugiere un enfoque innovador basado la arquitectura "Swin Transformer", el cual ha demostrado ser altamente efectivo en tareas de visión artificial pero que nunca ha sido aplicado en la clasificación de enfermedades del tomate.

En este contexto, se plantea las siguientes preguntas:

* ¿Cómo desarrollar un modelo de detección y clasificación de enfermedades del tomate utilizando la arquitectura Swin Transformer que supere la precisión de las redes neuronales convolucionales (CNN)?
* ¿Cómo se puede construir un dataset balanceado en las diferentes clases de las enfermedades del tomate?
* ¿Cómo evaluar e interpretar las predicciones del modelo Swin Transformer para cada clase?
* ¿Cómo se compara la exactitud (accuracy) del modelo Swin Transformer con los modelos basados en redes neuronales convolucionales?

# OBJETIVOS

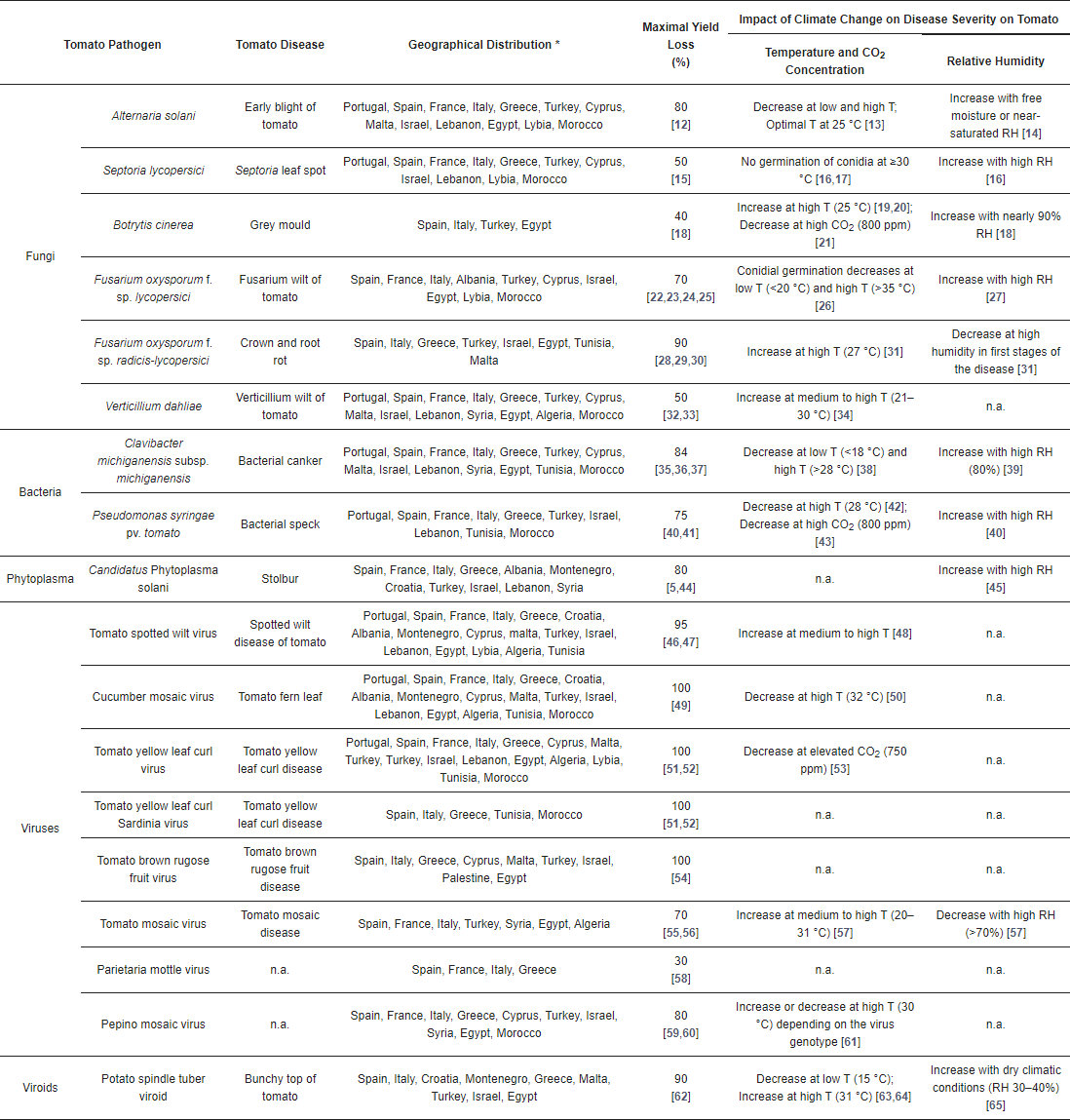
## Objetivo general

Desarrollar un modelo de detección y clasificación de enfermedades de plantas de tomate utilizando la arquitectura Swin Transformer, con el propósito de conseguir una precisión superior en comparación con las redes neuronales convolucionales.

## Objetivos específicos

* Construir un dataset balanceado en las diferentes clases de las enfermedades del tomate.
* Evaluar e interpretar las predicciones del modelo Swin Transformer para cada clase.
* Comparar y analizar la exactitud (accuracy) del modelo Swin Transformer con modelos basados en redes neuronales convolucionales.

**Tabla 1.** Lista de patógenos y enfermedades de la planta de tomate (Panno et al., 2021).



# ESTADO DEL ARTE

En el artículo ***"Machine Learning Approach towards Tomato Leaf Disease Classification"***, Gadade y Kirange, (2020) presentan un enfoque de machine learning para la identificación y clasificación de las enfermedades de las hojas del tomate. El proceso de investigación inició con la recolección de 9,000 imágenes de hojas de tomate del dataset de PlantVillage, enfocándose en siete clases de plantas enfermas y una clase de planta sana.

En la fase de preprocesamiento se minimizó el ruido de las imágenes utilizando un filtro de mediana, después se realizó la extracción de características mediante las técnicas GLCM (Gray-Level Co-Occurrence Matrix), Gabor y SURF (Speeded Up Robust Features). Para clasificar las imágenes en categorías normales o enfermas, se emplearon varios algoritmos de clasificación como la máquina de soporte vectorial (SVM), k-Nearest Neighbors (KNN), Naïve Bayes (NB) y árboles de decisión.

Los resultados mostraron que la combinación de SVM y Gabor obtuvo métricas superiores a todas las demás con un 'accuracy' de 0,7339, 'precision' de 0,2525, 'recall' de 0,9492 y 'f1-score' de 0,3989. A pesar de estos resultados, los autores indican que técnicas más avanzadas, como Adaptive Neuro Fuzzy, Neural Networks y algoritmos genéticos, podrían mejorar la clasificación.

No obstante, este trabajo evidencia que los métodos tradicionales de machine learning muestran métricas inferiores en comparación con las técnicas modernas de Deep learning basadas en redes neuronales convolucionales. Esta observación señala una posible obsolescencia de los métodos tradicionales en el ámbito de la clasificación de enfermedades en hojas de tomate a través de técnicas de visión artificial.

En el estudio titulado ***"ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network"*** de Mohit Agarwala et al. (2020) se propone un enfoque basado en una Red Neuronal Convolucional (CNN) para la detección y clasificación de enfermedades en las hojas del tomate. El estudio utiliza el dataset de Plant Village que contiene imágenes de hojas de tomate afectadas por nueve tipos de enfermedades, además de una clase de hojas sanas. Para equilibrar el dataset, se aplicaron técnicas de aumento de datos resultando en 10,000 imágenes para el entrenamiento, 7,000 para la validación y 500 para las pruebas.

La arquitectura CNN desarrollada en el estudio consta de 3 capas convolucionales, 3 capas de max-pooling y 2 capas completamente conectadas. Se evaluaron también modelos CNN preentrenados a través de transferencia de aprendizaje, empleando el mismo dataset, con el objetivo de contrastar la eficacia del modelo propuesto. Las métricas resultantes mostraron una precisión en un rango de 76% a 100% para diversas clases, con una precisión promedio de 91.2%. Este rendimiento supera al de modelos preentrenados como VGG16, con una precisión del 77.2%, InceptionV3, con 63.4%, y MobileNet, con 63.7%

El modelo propuesto tiene un número menor de parámetros entrenables comparado al de los modelos preentrenados, sugiriendo potenciales ventajas en almacenamiento y computación. No obstante, la variabilidad en la exactitud entre clases y la comparación con trabajos más actuales indican que el modelo, aunque eficaz, está superado por enfoques más recientes.

El artículo ***"Tomato plant disease detection using transfer learning with C-GAN synthetic images"*** de Abbas et al. (2021) aborda un enfoque de Deep learning para la detección y clasificación de enfermedades de la planta del tomate. La metodología implementa dos fases principales, la generación de imágenes sintéticas mediante Redes Antagónicas Generativas Condicionales (C-GAN) para extender el dataset y la clasificación de enfermedades utilizando un modelo DenseNet121 previamente entrenado.

El desempeño del método se evaluó con el dataset de PlantVillage, que incluye nueve clases de hojas de tomate con distintas enfermedades además de una clase de hojas sanas. Las imágenes sintéticas generadas con C-GAN se añadieron al dataset existente. Luego, se entrenó el modelo DenseNet121 con el conjunto de imágenes extendido.

Se realizaron dos series de experimentos, el primero centrado en el entrenamiento de C-GAN y la generación de imágenes sintéticas; el segundo, en el entrenamiento del modelo DenseNet121 utilizando el dataset ampliado. Un análisis comparativo con modelos preentrenados como VGG19, ResNet50, Inception-V3, Xception y MobileNet mostró que el modelo propuesto logró el más alto nivel de exactitud, con un 97.11% en la clasificación de las 10 clases del estudio.

No obstante, aunque el modelo propuesto demostró una alta eficacia en la clasificación de enfermedades de la planta de tomate y la aplicación de C-GAN ayudó a evitar el sobreajuste, es importante evaluar la fidelidad de las imágenes sintéticas generadas, ya que podrían no tener el nivel de detalle que presentan las imágenes reales y, por lo tanto, afectar negativamente el rendimiento del modelo en tareas de generalización.

El estudio ***"Early Detection and Classification of Tomato Leaf Disease Using High-Performance Deep Neural Network"*** de Trivedi et al. (2021) aborda la identificación y clasificación de enfermedades en hojas de tomate mediante Convolutional Neural Networks (CNN). Se empleó un conjunto de datos compuesto por 3000 imágenes, distribuidas en nueve clases de hojas afectadas y una clase de hojas sanas. En la fase de preprocesamiento, las imágenes se normalizaron y redimensionaron.

El proceso comprende etapas de preprocesamiento, segmentación y clasificación a través de CNN. La arquitectura CNN especificada consta de ocho capas convolucionales, ocho capas de max pooling, tres capas fully connected, con una tasa de dropout de 0.5 y función de activación Relu. Se realizaron experimentos variando parámetros como la tasa de aprendizaje y el número de épocas.

Los resultados de la métrica de exactitud fueron del 98.49%. Los experimentos adicionales mostraron que la precisión varía levemente con diferentes tasas de aprendizaje y épocas, oscilando en general alrededor del 98.4% al 98.58%. La comparación con modelos estándar como MobileNet, VGG16 e InceptionV3 indicó que el modelo propuesto supera a estos modelos CNN en exactitud.

El estudio resalta la eficacia del modelo propuesto, especialmente en comparación con otros modelos. Sin embargo, omite discusiones sobre limitaciones, aplicabilidad futura o la justificación para la elección de la arquitectura específica empleada.

El artículo titulado ***"Tomato leaf disease classification using supervised learning techniques: contrasting analysis"***, presentado por Rawat et al., (2022), ofrece una contribución al campo de la clasificación de enfermedades en hojas de tomate a través de técnicas de aprendizaje supervisado.

El estudio se articula en torno a cuatro componentes principales: selección del dataset de imágenes de tomates del archivo "Plant Village", el preprocesamiento de estas imágenes para la eliminación de ruido mediante un filtro de mediana, la aplicación de diversos algoritmos de aprendizaje automático como SVM, K-Nearest Neighbour, Naïve Bayes, Decision Tree, Feed Forward Neural Network, Back Propagation Neural Network, Deep Neural Network, Convolutional Neural Network y Multi Kernel SVM, y por último la evaluación de la efectividad de estos algoritmos mediante métricas como Precisión, F1-Score y Recall.

El estudio realiza un análisis comparativo de la eficacia de los diferentes algoritmos. En este análisis, la Red Neuronal Convolucional (CNN) logra una exactitud de 98,5%, superando a otros algoritmos como SVM, que alcanza un 90% de exactitud, y KNN con un 83,6%. La investigación concluye que las CNN ofrecen un rendimiento superior en la clasificación de enfermedades de hojas de tomate en comparación con métodos tradicionales de machine learning.

El estudio titulado ***"AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf"*** (Chen et al., 2022) aborda la detección y clasificación de enfermedades en hojas de tomate mediante una red neuronal convolucional (CNN) basada en la arquitectura de AlexNet. Los autores utilizaron un conjunto de datos de 18,345 imágenes de entrenamiento y 4,585 imágenes de prueba, distribuidas en nueve clases correspondientes a diversas enfermedades de la hoja del tomate y una clase de hoja sana.

La modificación realizada a la arquitectura de la red AlexNet involucra tres capas de convolución, tres capas completamente conectadas y una capa de salida. Se utilizó el optimizador Adam con una tasa de aprendizaje de 0.0005, 75 épocas y un tamaño de lote de 128. La función de pérdida fue la entropía cruzada. El modelo mostró una exactitud del 96%, una tasa de precisión del 98%, un valor de recall del 95% y un F1-score del 97%.

Las conclusiones destacan la eficacia del modelo para la clasificación de enfermedades en hojas de tomate. Se señala que el modelo es lo suficientemente ligero como para implementarse en dispositivos móviles, lo que es relevante dadas las limitaciones de capacidad de memoria en estos dispositivos.

El estudio ***"BotanicX-AI: Identification of Tomato Leaf Diseases Using an Explanation-Driven Deep-Learning Model"*** de Bhandari et al., (2023), aborda la clasificación de nueve clases de enfermedades en hojas de tomate y una clase para las hojas sanas utilizando un modelo de aprendizaje profundo basado en EfficientNetB5. El dataset se construye preprocesando las imagenes mediante redimensionamiento y aumentos de datos resultando en 11,000 imágenes de hojas de tomate distribuidas en 10 clases.

La metodología consiste en aplicar transferencia de aprendizaje utilizando un modelo de EfficientNetB5 al que se le agregaron capas adicionales como la normalización por lotes, capas densas y técnicas de regularización como dropout. Para incrementar la interpretabilidad del modelo se integran técnicas como GradCAM y LIME, que ayudan a identificar las regiones de la imagen que más contribuyen a la clasificación.

En cuanto a métricas el modelo propuesto alcanzó una exactitud del 99.07%, superando a otros modelos preentrenados, como MobileNet (94,00%), Xception (95,32%), VGG16 (93,35%), ResNet50 (96,03%) y DenseNet121 (96,30%). Las técnicas de GradCAM y LIME pudieron identificar las regiones más importantes en las imágenes de las hojas para la clasificación. Aunque se observó que GradCAM tenía limitaciones en la identificación de ciertas regiones en algunas clases de enfermedades.

Finalmente, los autores recomiendan la consideración de otras técnicas de interpretabilidad, como HiResCAM y SHAP, para trabajos futuros, argumentando que estas pueden aumentar la confianza en la aplicabilidad del modelo.

Los estudios revisados abordan el problema de la detección y clasificación de enfermedades en hojas de tomate utilizando diversas técnicas que van desde machine learning tradicional hasta deep learning. Los algoritmos de machine learning, como SVM y KNN, muestran métricas de desempeño inferiores a los modelos basados en deep learning. Entre los modelos de deep learning, las Redes Neuronales Convolucionales (CNN) y sus variaciones, como DenseNet121 y EfficientNetB5, muestran métricas superiores en términos de precisión, exactitud y F1-score.

Los métodos que incorporan imágenes sintéticas generadas mediante Redes Antagónicas Generativas Condicionales (C-GAN) y técnicas de transferencia de aprendizaje parecen ofrecer ventajas en términos de desempeño y prevención de sobreajuste. Sin embargo, la calidad de las imágenes sintéticas generadas es una variable por considerar para tareas de generalización.

La aplicabilidad en dispositivos móviles y la interpretabilidad del modelo son aspectos que se resaltan en algunos trabajos, evidenciando la importancia de estos factores en aplicaciones prácticas. Las técnicas de explicabilidad como GradCAM y LIME ofrecen posibilidades para entender las decisiones del modelo, aunque se sugiere que métodos adicionales pueden mejorar la confianza en el modelo.

Los métodos basados en deep learning muestran un desempeño superior y se destacan como el método más efectivo en la clasificación de enfermedades de la planta de tomate. Sin embargo, no han presentado innovaciones en su arquitectura recientemente. Además, no se ha investigado la aplicación de arquitecturas de visión transformer para esta problemática específica, lo cual representa un área no explorada que podría ofrecer soluciones a limitaciones como la interpretabilidad o una mayor exactitud y que por tanto justifica una investigación más exhaustiva.

# MARCO TEÓRICO

## Enfermedades de la planta de tomate

* + 1. ***Definición***

Las enfermedades en plantas de tomate son alteraciones fisiológicas o estructurales de la planta originadas por entidades biológicas conocidas como patógenos y que afectan negativamente la salud o el rendimiento de la planta. La manifestación de estas enfermedades requiere la concurrencia de tres elementos fundamentales; presencia de agentes patogénicos capaces de inducir enfermedad (por ejemplo, hongos, bacterias o virus); un huésped (la planta de tomate) que se encuentra en un estado susceptible, comúnmente debido a condiciones de estrés como sobrefertilización, salinidad del suelo, uso excesivo de pesticidas, competencia por espacio o excesiva irrigación; y por ultimo las condiciones ambientales que facilitan la acción del agente patogénico, tales como temperatura y humedad (Flores, 2012; Bernal, 2010).

### Enfermedades comunes en la planta de tomate

Las enfermedades en tomates pueden ser causadas por una variedad de patógenos, incluidos hongos, bacterias, fitoplasmas, virus y viroides (Panno et al., 2021). Según Corrales Paredes (2021), a nivel mundial existen más de 200 patógenos que afectan a las plantas de tomate. Los de mayor importancia son Alternaria solani causante del tizón temprano, Phytophthora infestans causante del tizón tardío, Fusarium oxysporum y Ralstonia solanacearum causantes de marchitamiento y las enfermedades virales.

Algunas de las enfermedades más comunes que afectan a los tomates en la región del Mediterráneo incluyen el tizón temprano provocado por Alternaria solani, la mancha foliar por septoria por Septoria lycopersici, el moho gris por Botrytis cinerea, así como el marchitamiento y la putrefacción de la raíz y la corona causados por Fusarium oxysporum. Además del Verticillium dahliae que es el agente causal del marchitamiento por vercitilum.

En cuanto a las enfermedades bacterianas, Clavibacter michiganensis provoca el cancro bacteriano y Pseudomonas syringae causa la mancha bacteriana. Candidatus Phytoplasma solani es responsable de la enfermedad conocida como Stolbur. Diversos virus, como Tomato spotted wilt virus, Cucumber mosaic virus, Tomato yellow leaf curl virus y Tomato yellow leaf curl Sardinia virus, están asociados con enfermedades como la marchitez manchada del tomate, el rizado amarillo de las hojas y el mosaico del tomate. Viroides como Potato spindle tuber viroid causan el síndrome de crecimiento excesivo o "bunchy top".

Blanchard (2011) indica que en regiones tropicales las patologías bacterianas en tomates incluyen sarna bacteriana originada por diversas especies de Xanthomonas y moteado originado por Pseudomonas tomato. Asimismo, en estas regiones, se registran virus como Tomato yellow curl virus (TYLCV) transmitidos por la mosca Bemisia tabaci. En contextos tropicales húmedos, el marchitamiento bacteriano generado por Ralstonia solanacearum es común. En regiones tanto templadas como tropicales, Tomato spotted wilt virus (TSWV) y Pepino mosaic virus (PepMV) impactan en el cultivo de tomates. Con relación a las enfermedades fúngicas, se registran patologías causadas por Phytophthora infestans, Alternaria tomatophila, Verticillium dahliae y Fusarium oxysporum. Otros agentes patogénicos relevantes del tomate incluyen nematodos, insectos y ácaros (Panno et al., 2021).

### Síntomas

Los síntomas de las enfermedades de la planta de tomate varían considerablemente y están estrechamente ligados al tipo de patógeno involucrado y a la enfermedad específica que se manifiesta. Los síntomas pueden incluir, pero no están limitados a, decoloración de hojas, marchitamiento, necrosis, manchas foliares, deformación de frutos y presencia de hongos o esporas en diversas partes de la planta. Además, la severidad y la aparición de síntomas pueden verse influenciadas por factores externos como condiciones ambientales y prácticas de manejo del cultivo. Es crucial para el manejo efectivo de la enfermedad identificar correctamente los síntomas asociados con cada patógeno o enfermedad específica (Blancard, 2011; Panno et al., 2021).

Las enfermedades que afectan la planta de tomate se pueden categorizar según los síntomas que manifiestan. El tizón temprano y la mancha foliar por septoria comparten características como defoliación y lesiones necróticas en diferentes partes de la planta. El marchitamiento por fusarium, putrefacción de la raíz y la corona y marchitamiento por verticillium como su nombre lo indica presentan síntomas de marchitamiento y pardeamiento vascular. El moho gris se distingue por síntomas como podredumbre blanda y encharcamiento de tejidos.

El cancro y la mancha bacterianos muestran lesiones en diferentes partes de la planta y enrollamiento de las hojas en el caso de la mancha bacteriana. Stolbur phytoplasma, Tomato yellow leaf curl virus y Tomato brown rugose fruit virus presentan síntomas como amarilleamiento y atrofia o escasez de hojas.

Tomato spotted wilt virus, Parietaria mottle virus y Cucumber mosaic virus incluyen síntomas de deformación foliar y lesiones necróticas. Tomato mosaic virus y Pepino mosaic virus se caracterizan por síntomas como moteado de hojas y deformación de los frutos. Potato spindle tuber viroid manifiesta una variedad de síntomas, incluyendo retraso del crecimiento y deformidad de los frutos (Panno et al., 2021).

### Métodos tradicionales de detección de enfermedades

* + - 1. La detección manual**.** Se lleva a cabo por expertos que necesitan un amplio rango de conocimientos y experiencia en diversas disciplinas. Estos conocimientos son esenciales debido a la variedad de enfermedades existentes. El diagnóstico no solo se basa en síntomas y signos, sino también en la comprensión de la biología de los agentes patógenos y el contexto en el que la enfermedad ocurre. El proceso de diagnóstico implica observaciones detalladas, preguntas dirigidas a contextualizar la enfermedad y razonamiento para priorizar posibles causas. Este enfoque es común en áreas rurales y requiere una considerable cantidad de tiempo y recursos humanos para su implementación en grandes fincas. Además, la evaluación manual es susceptible a errores humanos como sesgos o variables cognitivas y emocionales que afectan el juicio clínico (Blancard, 2011; Thangaraj et al., 2022; Garcia, 2016).
      2. Los métodos de laboratorio**.** Incluyen técnicas inmunológicas, cultivo de patógenos y microscopía. Aunque estos métodos son precisos, son laboriosos, costosos y requieren tiempo, lo que puede no ser viable para un diagnóstico rápido y eficaz (Verma et al., 2018; Valdivieso, 2021).

### Desafíos en la detección de enfermedades

Dado los desafíos asociados con los métodos tradicionales de detección de enfermedades, como el tiempo, costo y la necesidad de expertos altamente calificados, existe una necesidad imperante de desarrollar nuevas estrategias para la detección y control de enfermedades en la planta de tomate. Los errores en la detección o el retardo en la intervención pueden resultar en pérdida de rendimiento y calidad del producto (Garcia, 2016; Valdivieso, 2021).

La ineficacia en la detección y control de enfermedades en la planta de tomate puede resultar en productos agrícolas que son huéspedes de patógenos, lo cual representa un riesgo para la salud pública. Asimismo, el uso indebido de pesticidas, a menudo utilizado como medida preventiva ante la incertidumbre en la detección de enfermedades, puede resultar en la acumulación de residuos químicos en los alimentos, comprometiendo así la seguridad alimentaria (Corrales Paredes, 2021).

## Seguridad alimentaria (food safety)

La seguridad alimentaria se centra en prevenir enfermedades transmitidas por alimentos a través de la gestión de toda la cadena de producción, desde el cultivo hasta el consumo. Las fuentes comunes de contaminación incluyen hongos, bacterias, virus, viroides, parásitos y sustancias químicas (Guennouni et al., 2022; FAO, 2022). Esta contaminación amenaza tanto la salud individual como pública y puede comprometer la calidad nutricional de los alimentos. Por lo tanto, es necesario implementar medidas de control en cada etapa de la cadena alimentaria (Uçar et al., 2016).

Es por eso por lo que en el ámbito de la producción de plantas de cultivo como el tomate la detección temprana de enfermedades cumple un rol fundamental para la seguridad alimentaria, velando por la salud tanto del cultivo como la de los consumidores. Ante esta necesidad, la tecnología de la visión artificial se presenta como un método efectivo para identificar enfermedades de forma automática, no destructiva y económica. Esta tecnología contribuye a la biovigilancia al permitir una identificación más rápida y precisa de microorganismos patógenos (Khan et al., 2021).

## Visión artificial

Los sistemas de visión artificial en la agricultura y la seguridad alimentaria ofrecen ventajas a comparación de los métodos tradicionales de detección tales como mayor velocidad de operación, consistencia, fiabilidad y adaptabilidad en entornos industriales (Khan et al., 2021). La visión artificial contribuye a la optimización sostenible de la producción agrícola mediante la incorporación de una nueva tecnología de identificación de plagas y enfermedades (J. Gómez-Camperos et al., 2021).

La visión artificial es un campo de la inteligencia artificial que tiene como objetivo permitir a sistemas computacionales procesar e interpretar datos visuales. Se utiliza algoritmos y modelos matemáticos para extraer información de imágenes y vídeos (Xu et al., 2021). El campo está en desarrollo continuo y tiene aplicaciones en áreas como detección de objetos, segmentación de imágenes, detección de rostros y conducción autónoma (Mahadevkar et al., 2022).

En el ámbito de la agronomía, la visión artificial se emplea principalmente para el diagnóstico temprano de enfermedades en plantas, específicamente mediante la identificación de síntomas en las hojas. Los avances en reconocimiento de patrones y aprendizaje automático han permitido el desarrollo de nuevos métodos diagnósticos. Esto ha generado incremento en la investigación y en la implementación de nuevos modelos y técnicas de visión artificial (Nanehkaran et al., 2020).

### Clasificación de Imágenes

Es un proceso que se enmarca en la visión artificial. Su objetivo es categorizar imágenes según su contenido a partir de un conjunto de categorías ya establecido. Es relevante en diversos campos científicos e ingenieriles para la captura y tratamiento de datos visuales (Vocaturo, 2021). Se utilizan múltiples técnicas para esta tarea, incluyendo redes neuronales artificiales, redes neuronales convolucionales, máquinas de soporte vectorial, k-nearest neighbor y algoritmos de random forest. Estas técnicas facilitan la extracción de características de las imágenes para entrenar un modelo que posteriormente pueda clasificar nuevas imágenes (Sanghvi et al., 2020).

### Machine learning y Deep learning en visión artificial

La visión artificial utiliza enfoques como Machine Learning y Deep Learning para interpretar datos visuales. El Machine Learning emplea una variedad de algoritmos como k-nearest neighbors, Bayesian classification y support vector machines para tareas de aprendizaje supervisado, no supervisado y de refuerzo (Szeliski, 2020). El Deep Learning, un subcampo del Machine Learning, usa grandes redes neuronales artificiales para resolver problemas complejos y ha mostrado mayor eficacia en diversas tareas de visión artificial (Mahadevkar et al., 2022; Szeliski, 2020). Una distinción clave entre los dos enfoques es que en el Machine Learning, las características de las imágenes se extraen manualmente, mientras que en Deep Learning, el algoritmo aprende representaciones y etapas de clasificación de manera autónoma. Desde la introducción de "AlexNet" en 2012, ha habido mejoras continuas en la clasificación y segmentación de imágenes, así como en la detección de objetos, en gran parte debido a redes más profundas y algoritmos de entrenamiento mejorados (Szeliski 2020; Shorten & Khoshgoftaar, 2019). Ambos métodos se aplican actualmente en áreas como la detección de enfermedades en cultivos (Ouhami et al., 2021).

### Visión artificial mediante Deep learning

El enfoque de deep learning en visión artificial es integral y se centra en la optimización de cada etapa del proceso de aprendizaje automático. Este enfoque requiere que la función de pérdida sea una función diferenciable de todos los parámetros involucrados, permitiendo así una búsqueda factible de los parámetros que minimizan la pérdida durante el entrenamiento. Las redes neuronales profundas proporcionan esta arquitectura de cómputo uniforme y diferenciable (Szeliski, 2020).

Este enfoque intenta imitar las redes neuronales biológicas y el interés por explorarlo ha tenido fluctuaciones desde finales de los años 50. Durante ese tiempo, varias arquitecturas de redes neuronales fueron desarrolladas, incluidas aquellas que usaban unidades estocásticas como las Máquinas de Boltzmann y las Máquinas de Boltzmann Restringidas. En la actualidad se han vuelto populares las redes neuronales profundas deterministas y discriminativas con activaciones de valor real, entrenadas mediante descenso de gradiente (Szeliski, 2020).

Comparado con otras técnicas de aprendizaje automático que generalmente dependen de varias etapas de preprocesamiento para extraer características, los enfoques de deep learning suelen estar entrenados de extremo a extremo, pasando directamente de píxeles en bruto a las salidas deseadas finales, este enfoque contribuye a la eficacia en las tareas complejas de reconocimiento visual y clasificación de imágenes (Szeliski, 2020).

Diversos modelos de aprendizaje profundo se emplean en la clasificación de imágenes en visión artificial. Entre estos, las redes neuronales convolucionales (CNN) son las más comunes y han mostrado un rendimiento destacado en tareas de clasificación de imágenes (Shah & Harpale, 2018). Modelos bayesianos de aprendizaje profundo también se han utilizado en estas tareas y son capaces de modelar tanto incertidumbres aleatorias como epistémicas (Szeliski, 2020). Los Vision Transformers (ViT) representan una arquitectura más reciente basada en transformadores, utilizada en tareas de visión artificial. Estos modelos han emergido como alternativas competitivas a las CNN, especialmente en su capacidad para manejar dependencias a largo plazo, y se han aplicado en tareas como la clasificación, segmentación de imágenes y detección de objetos (Amjoud & Amrouch, 2023).

## Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales (CNN) son una variante de las redes neuronales artificiales tradicionales (ANN, por sus siglas en inglés). Ambos tipos de redes contienen neuronas que se optimizan a través del aprendizaje. En una CNN, cada neurona recibe una entrada y realiza una operación, tal como un producto escalar seguido de una función no lineal, similar a una red neuronal tradicional. La red completa expresa una única función de puntuación perceptiva, relacionada con los pesos de la red. La última capa de la CNN contiene funciones de pérdida asociadas a las clases del problema que se está resolviendo (O’Shea & Nash, 2015).

La principal diferencia entre CNN y redes neuronales tradicionales es la especialización de las CNN en el reconocimiento de patrones dentro de imágenes. Esto se logra codificando características específicas de las imágenes en la arquitectura de red, lo que permite reducir el número de parámetros necesarios para configurar el modelo. Esta especialización permite a las CNN manejar de forma más eficiente la complejidad computacional asociada con el procesamiento de datos de imágenes, que puede ser problemática para las ANN. Por ejemplo, un conjunto de datos como MNIST, que tiene imágenes con dimensiones de 28x28 píxeles, suele ser computacionalmente manejable para las redes neuronales artificiales tradicionales. Pero si se consideran imágenes a color de mayor tamaño, como 64x64, la cantidad de pesos en una sola neurona de la primera capa oculta aumenta significativamente, lo que eleva los requisitos computacionales. Los CNN ofrecen una solución más eficiente para tratar con este tipo de datos de alta dimensionalidad (O’Shea & Nash, 2015).

Szeliski (2020) describe que, en las redes neuronales convolucionales, cada capa se organiza en mapas de características en lugar de conectar todas las unidades en una capa a todas las unidades en una capa anterior. En una capa convolucional, las sumas ponderadas se realizan solo dentro de una pequeña ventana local denominada kernel. Los pesos dentro de un kernel son idénticos para todos los píxeles de una capa y canal dado, lo que resulta en un menor número de pesos a entrenar en comparación con las capas completamente conectadas. Durante la retropropagación, las actualizaciones del peso del kernel se suman en todos los píxeles de una capa/canal. Este enfoque permite la construcción de características locales que luego se combinan de diferentes maneras para producir características más discriminativas y semánticamente significativas.

Las CNN se han convertido en una técnica dominante en el campo de la clasificación de imágenes, detección de objetos, segmentación semántica e identificación de enfermedades de las plantas (Xiao et al., 2022; Nanehkaran et al., 2020). Han mostrado avances significativos en su arquitectura y han superado a otros métodos de clasificación comunes como k-NN, SVM, Naives Bayes, Linear Regression y ANN (Sood y Singh 2021; Xiao et al, 2022). Sin embargo, aunque las CNN son eficientes para el procesamiento de imágenes, presentan ciertas limitaciones. Una de estas limitaciones, según Szeliski (2020), radica en los sesgos inductivos inherentes a las operaciones convolucionales, que asumen que los píxeles cercanos son más importantes que los píxeles distantes. Estos sesgos pueden limitar el progreso a medida que crecen los conjuntos de datos, los modelos y el poder computacional.

En este contexto, los Vision Transformers (ViTs) emergen como una solución que aborda estas limitaciones. Inicialmente desarrollados para aplicaciones en procesamiento de lenguaje natural, los ViTs adaptan la arquitectura de los transformers para tareas de visión computacional. A diferencia de las CNN, que operan con un sesgo de localidad espacial, los ViTs son capaces de operar en toda la secuencia de entrada a la vez, permitiendo una atención más globalizada de la imagen. Esto los ha llevado a superar a las CNN en diversas evaluaciones de referencia en el ámbito de la visión computacional (Szeliski, 2020; Jamil et al., 2023)."

## Vision Transformer

Las arquitecturas de Vision Transformer (ViTs) se han destacado en tareas de clasificación de imágenes, estableciendo un nuevo estado del arte en benchmarks como ImageNet (Papers With Code, 2023). Estos modelos adaptan la arquitectura de transformers, que fueron inicialmente diseñados para aplicaciones en procesamiento de lenguaje natural, a tareas de visión artificial. La metodología central consiste en dividir una imagen en parches, aplanarlos y procesarlos a través de capas de transformers para captar relaciones de largo alcance en la imagen (Alexey Dosovitskiy et al., 2020).

Según Szeliski (2020), el componente fundamental de un transformer es el mecanismo de auto-atención, que se forma a partir de aplicar atención a cada una de las N activaciones unitarias en una capa determinada de la red. La operación de auto-atención produce un conjunto de vectores de salida basados en todas las entradas. Esto se diferencia de las capas completamente conectadas y convolucionales, donde los pesos son fijos y se aplica una atención más localizada.

La atención se calcula utilizando la atención de producto escalar, en la que se realiza el producto escalar entre vectores de query y key, se escala según la raíz cuadrada de la dimensión de estos vectores y finalmente se aplica la función softmax. El bloque de transformer se compone tanto de un bloque de encoder como de un bloque de decoder, aunque ambos comparten muchos de los mismos componentes. En muchas aplicaciones, el encoder puede utilizarse sin el decoder y viceversa. Durante la fase de entrenamiento, una diferencia significativa en el bloque de decoder es el enmascaramiento de algunos de los vectores de entrada a la auto-atención. Este enmascaramiento permite el entrenamiento paralelo en tareas de predicción autorregresiva, optimizando la eficiencia computacional (Szeliski, 2020).

Los ViTs son capaces de considerar toda la secuencia de entrada simultáneamente, lo que permite una atención más globalizada de la imagen. Esto los ha llevado a superar a las CNN en diversas evaluaciones de referencia en el ámbito de la visión computacional (Szeliski, 2020; Jamil et al., 2023).

Sin embargo, los ViT tienen ciertas limitaciones, una de las principales dificultades de aplicar transformers al dominio de las imágenes radica en el tamaño de la entrada de imagen. En términos de complejidad computacional, el número de operaciones en punto flotante (FLOPs) necesarias para la auto-atención es del orden de , donde N es la longitud de la entrada y D es el número de dimensiones de cada entrada. En comparación, las FLOPs para una operación de convolución es del orden de donde K representa el tamaño del núcleo de la convolución. En este caso, la operación de convolución resulta significativamente más eficiente en términos de FLOPs que la auto-atención. (Szeliski, 2020).

Estas limitaciones hacen que los Vision Transformers requieran grandes cantidades de datos y recursos computacionales para el entrenamiento desde cero. Además, la eficiencia computacional de la atención global utilizada en los ViTs puede no ser óptima, especialmente para imágenes de alta resolución (Alexey Dosovitskiy et al., 2020; Touvron et al., 2021). Esta problemática conduce a la búsqueda de alternativas más eficientes, como la arquitectura Swin Transformer.

Diagrama

Descripción generada automáticamente

**Figura 1:** Model overview of the first visual transformer (Alexey Dosovitskiy et al., 2020)

## Swin Transformer

Swin Transformer, también conocido como Shifted Window Transformer, es una arquitectura que busca abordar ciertas limitaciones presentes en las arquitecturas de Vision Transformers (ViTs) y redes neuronales convolucionales (CNN). La arquitectura se caracteriza por dividir la imagen en parches y aplicar atención dentro de ventanas locales no solapadas, en contraste con la atención global empleada en ViTs. Este enfoque reduce la complejidad computacional a lineal respecto al tamaño de la imagen. Adicionalmente, la arquitectura utiliza una estructura jerárquica para procesar imágenes a diferentes resoluciones, lo que le permite capturar características en múltiples escalas. En pruebas de referencia, se ha encontrado que Swin Transformer supera a otras arquitecturas de ViTs y CNN en términos de rendimiento en tareas como la clasificación de imágenes en el conjunto de datos de ImageNet (Liu et al., 2021; Papers With Code, 2023). Este diseño le confiere aplicabilidad a una variedad de tareas en el ámbito de la visión artificial, incluida la clasificación de imágenes, la detección de objetos y la segmentación semántica.

### Arquitectura de Swin Transformer

Swin Transformer, aborda las limitaciones de los ViT introduciendo dos conceptos clave: ventanas desplazadas y estructura piramidal. Swin Transformer comienza dividiendo una imagen en parches no solapados mediante un módulo de división. Cada parche se considera un "token" y su característica se establece concatenando los valores RGB de los píxeles en bruto. En la aplicación de Swin Transformer, se utiliza un tamaño de parche de 4 × 4, lo que da como resultado una dimensión de característica de 48 (4 × 4 × 3). Estas características se proyectan posteriormente a una dimensión arbitraria (denotada como C) mediante una capa de embedding lineal.

En Swin Transformer, la resolución de la imagen se reduce sucesivamente, de la "Etapa 1" a la "Etapa 4". El término "Etapa" se refiere a las distintas fases de procesamiento de la imagen a través de la arquitectura. Esta estructura jerárquica es similar a la de las redes neuronales convolucionales tradicionales, como VGG y ResNet, en las que las características se procesan a distintas resoluciones (Liu et al., 2021).

### Ventanas desplazadas

En lugar de realizar una atención global sobre toda la imagen (lo que tendría una complejidad cuadrática con respecto al número de tokens), Swin Transformer introduce el concepto de atención dentro de ventanas locales no solapadas. Esto reduce significativamente la complejidad computacional. La atención basada en ventanas es escalable y computacionalmente más eficiente (Liu et al., 2021).

Imagen de la pantalla de una ventana

Descripción generada automáticamente con confianza media

**Figura 2.** El enfoque de ventana deslizante para calcular la autoatención en la arquitectura Swin Transformer

Sin embargo, para no perder la capacidad de modelar las relaciones entre parches en distintas ventanas, Swin Transformer introduce la idea de ventanas desplazadas en bloques sucesivos. Esta estrategia alterna entre dos configuraciones de particionamiento en bloques consecutivos de transformers, lo que permite que cada parche se relacione con los parches de las ventanas vecinas. También se introduce un sesgo de posición relativa para tener en cuenta las relaciones espaciales entre parches dentro de una ventana. Este sesgo mejora significativamente el rendimiento del modelo y es preferible a los embeddings de posición absoluta (Liu et al., 2021).

### Estructura piramidal

Swin Transformer procesa imágenes en una serie de resoluciones, de forma similar a una pirámide. Comienza con ventanas pequeñas en una resolución más alta y agrupa gradualmente los parches mientras reduce la resolución, permitiendo que el modelo capture características en diferentes escalas (Liu et al., 2021).

Gráfico, Diagrama

Descripción generada automáticamente

**Figura 3.** Comparación del método de estructura piramidal utilizado por Swin tranformer y el método utilizado por Vit.

### Detalles arquitectónicos de Swin Transformer

5.6.4.1 Token**:** se refiere a un parche de la imagen original. Cada parche se convierte en una representación vectorial o embedding que sirve como unidad básica de entrada para las capas posteriores del modelo.

5.6.4.2 Embedding**:** hace referencia a la representación vectorial de alta dimensión en la que se proyectan los tokens o parches de la imagen. Estas proyecciones vectoriales se utilizan para el procesamiento y la identificación de características en capas subsecuentes del modelo.

5.6.4.3 Atención**:** Este mecanismo permite al modelo asignar diferentes niveles de importancia a diversas partes de la entrada cuando está procesando información.

5.6.4.4 Bloque Swin Transformer**:** Un bloque Swin Transformer es una unidad de arquitectura en la red que contiene módulos específicos para la atención y la transformación de características. En general, cada bloque incluye un módulo de atención seguido de una capa de normalización y un perceptrón multicapa (MLP). Estos bloques se pueden apilar para formar la arquitectura completa del Swin Transformer.

5.6.4.5 MSA (Multi-Head Self-Attention)**:** un tipo de mecanismo de atención que permite al modelo prestar atención a diferentes partes de la entrada de manera simultánea. Es esencial para la capacidad del transformer de capturar relaciones a larga distancia.

Diagrama, Esquemático

Descripción generada automáticamente

**Figura 4.** A la izquierda se encuentra la arquitectura de un Swin Transformer (Swin-T) y a la derecha están dos bloques sucesivos del Swin Transformer W-MSA y SW-MSA, que son módulos de auto-atención multi-cabeza con configuraciones de ventana regulares y deslizantes, respectivamente.

5.6.4.6 MLP (Multi-Layer Perceptron)**:** Se refiere a una red neuronal completamente conectada. En el contexto de los transformers, los MLP se utilizan para transformar características después de capas de atención.

5.6.4.7 LN (LayerNorm)**:** LN, o Normalización de Capa, es una técnica de normalización que se aplica a las características en una capa específica. En Swin Transformer, LN se aplica antes de cada módulo de auto-atención multi-cabeza (MSA) y cada MLP. La normalización ayuda a estabilizar y acelerar el entrenamiento.

5.6.4.8 Conexión Residual**:** En redes profundas, las conexiones residuales ayudan a evitar el problema de desvanecimiento del gradiente al permitir que las activaciones se salten una o más capas. En Swin Transformer, se aplica una conexión residual después de cada módulo de atención y cada MLP.

5.6.4.9 Fusión de Parches**:** "Patch Merging" es una técnica utilizada en el Swin Transformer para reducir la resolución espacial del mapa de características e incrementar simultáneamente la dimensión de la característica, permitiendo que el modelo capture representaciones más abstractas a medida que se profundiza. Funciona agregando información de parches vecinos y creando un nuevo parche con una dimensión de característica más alta (Liu et al., 2021).

Después de explorar la arquitectura Swin Transformer, es necesario considerar el dataset en el que este modelo se va a entrenar y evaluar, debido a que las características propias del dataset también pueden influir en el resultado del entrenamiento del modelo.

## Dataset

Según Cappi et al. (2021) un "dataset" es una serie de realizaciones de una variable aleatoria o muestra, que es considerada representativa de alguna población si contiene características clave en proporciones similares a las de la población. En el contexto del Machine Learning, un dataset agrupa muestras utilizadas para el desarrollo de un modelo. La población aquí corresponde a todas las posibles observaciones que pueden realizarse en el dominio operacional. Los elementos de un dataset son llamados datos o muestras y son las realizaciones de las variables aleatorias de interés. Estos datos pueden incluir, por ejemplo, píxeles, imágenes o series temporales.

La representatividad del dataset es crucial durante la fase de prueba y entrenamiento para evaluar correctamente el rendimiento operacional del sistema. El dataset debe cubrir tanto como sea posible el dominio de situaciones que el sistema encontrará en operación. En el caso de procesamiento de imágenes, esto incluye varias condiciones físicas y ambientales, como objetos de interés, condiciones de exposición y condiciones ambientales, entre otros (Cappi et al., 2021).

Szeliski (2020) también destaca la importancia de probar algoritmos en datasets desafiantes y representativos y describe cómo la disponibilidad de datasets etiquetados a gran escala, como ImageNet y Microsoft COCO, revolucionó el desarrollo de algoritmos de reconocimiento de imágenes. Estos conjuntos de datos no solo proporcionan métricas confiables para el seguimiento del progreso de los algoritmos, sino que también ofrecen suficientes datos etiquetados para desarrollar soluciones completas basadas en machine learning y Deep learning.

Mohanty et al. (2016) abordan la complejidad de crear un dataset que minimice sesgos que puedan surgir durante el proceso de recopilación de datos. Subrayan la importancia de disponer de un conjunto de datos amplio y rigurosamente verificado para el desarrollo eficaz de clasificadores de imágenes en el campo de diagnóstico de enfermedades de plantas. Esta necesidad se acentúa por la ausencia histórica de datasets de alta calidad en este ámbito específico, brecha que el proyecto PlantVillage ha logrado cubrir.

## Dataset de plantVillage

El dataset PlantVillage proporciona un conjunto de datos de alta calidad para entrenar modelos de detección de enfermedades en plantas. Mohanty et al. (2016) afirman que, para el desarrollo de clasificadores de imágenes precisos para el diagnóstico de enfermedades de las plantas, es necesario un conjunto de datos grande y verificado de imágenes de plantas enfermas y saludables. El dataset de PlantVillage contiene 54,306 imágenes de hojas de plantas repartidas en 38 clases etiquetadas que abarcan 14 especies de cultivos y 26 enfermedades (Mohanty et al., 2016). Este dataset incluye una amplia gama de clases de enfermedades del tomate, lo que lo hace idóneo para el objeto de esta investigación.

### Clases asociadas a la planta del tomate en del dataset de plantVillage:

El dataset de plantVillage contiene 9 clases para enfermedades de la planta del tomate y una clase para la planta sana.

|  |  |
| --- | --- |
| **Nombre enfermedad** | **Número de imágenes** |
| Tomato Bacterial Spot | 2127 |
| Tomato Early Blight | 1000 |
| Tomato Late Blight | 1909 |
| Tomato Leaf Mold | 952 |
| Tomato Septoria leaf spot | 1771 |
| Tomato Two Spotted Spider Mite | 1676 |
| Tomato Target Spot | 1404 |
| Tomato Yellow Leaf Curl Virus | 3209 |
| Tomato Mosaic Virus | 373 |
| Tomato healthy | 1591 |
| Total | 16012 |
| Promedio | 1601,2 |

## Aumento de datos

El dataset de PlantVillage constituye una base sólida para el entrenamiento de modelos de diagnóstico de enfermedades en plantas. Sin embargo, como se menciona en la obra de Szeliski (2020), un método eficaz para minimizar el sobreajuste en tareas de clasificación de imágenes es el aumento de datos. Esta técnica consiste en perturbar las entradas y/o salidas de las muestras ya recogidas para ampliar el conjunto de entrenamiento. En el contexto de tareas de visión artificial, como la clasificación de imágenes, este enfoque resulta especialmente útil dado que obtener ejemplos etiquetados puede ser costoso y las clases de imágenes deben mantenerse constantes bajo pequeñas perturbaciones locales.

De acuerdo con el artículo de Khosla y Saini (2020), el aumento de datos se clasifica en dos categorías principales: transformaciones geométricas y sobremuestreo. Las transformaciones geométricas abarcan una variedad de técnicas que alteran la estructura y apariencia de las imágenes. Entre ellas se incluyen el volteo, que invierte la imagen; el cambio en el espacio de color, que podría convertir una imagen de RGB a escala de grises; el recorte, que elimina segmentos de la imagen; la rotación, que gira la imagen en un ángulo específico; la traslación, que desplaza la imagen en su plano; y la inyección de ruido, que añade ruido a la imagen. Aunque estas técnicas pueden incrementar la memoria y el tiempo de entrenamiento, son útiles para mitigar sesgos posicionales en los conjuntos de datos.

Por otro lado, el sobremuestreo incluye métodos como la mezcla de imágenes, donde una nueva imagen se crea superponiendo dos imágenes seleccionadas al azar del conjunto de entrenamiento. Este método puede generar nuevas muestras a partir de muestras existentes. Además, se pueden hacer modificaciones en el espacio de características de bajo nivel de las imágenes, a través de técnicas como la interpolación y la extrapolación. Las Redes Generativas Antagónicas (GANs) también se utilizan en el sobremuestreo para crear muestras sintéticas que mantienen las características del conjunto de datos original. La incorporación de estas técnicas de aumento de datos tiene el potencial de mejorar el rendimiento del modelo, especialmente cuando el conjunto de datos original es limitado o desequilibrado.

Yang et al. (2022) señalan que, aunque se ha prestado mucha atención al diseño de arquitecturas de red y al aumento de la capacidad de cálculo, se ha prestado menos atención a la generación de datos de entrenamiento cualificados mediante métodos de aumento de datos. Este descuido limita la eficiencia y la escalabilidad de las soluciones de visión artificial.

En el ámbito de la visión artificial, el aumento de datos puede añadir variabilidad al conjunto de datos de entrenamiento, lo que es especialmente valioso cuando el conjunto de datos es limitado o desequilibrado. Este proceso tiene implicaciones directas en las métricas de evaluación, ya que un modelo bien ajustado a un conjunto de datos más diverso podría mostrar un rendimiento más generalizado.

## Métricas de Evaluación

Para evaluar el rendimiento de un modelo de clasificación, Szeliski (2020) y Flach & Kull (2015) sugieren una serie de métricas, incluyendo Accuracy, Precision, Recall y F1-score. Estas métricas son especialmente relevantes para conjuntos de datos desequilibrados, que es uno de los casos más habituales en tareas de visión artificial. Antes de profundizar en dichas métricas, es fundamental clasificar las predicciones del modelo en las siguientes categorías:

* Verdaderos Positivos (TP): Casos positivos correctamente identificados.
* Verdaderos Negativos (TN): Casos negativos correctamente identificados.
* Falsos Positivos (FP): Casos negativos incorrectamente identificados como positivos.
* Falsos Negativos (FN): Casos positivos incorrectamente identificados como negativos.

A partir de estas categorizaciones, se derivan las siguientes métricas:

(1)

El Accuracy mide la proporción de predicciones correctas realizadas por el modelo sobre el total de predicciones realizadas. Sin embargo, la precisión puede ser engañosa cuando los datos están desequilibrados, es decir, cuando una clase es mucho más frecuente que las demás. En estos casos, un modelo que siempre predice la clase mayoritaria tendrá una gran precisión, pero puede no ser útil en la práctica. La “Precision” mide la proporción de verdaderos positivos (casos positivos predichos correctamente) de todos los casos predichos como positivos, es útil cuando el coste de los falsos positivos es elevado, como en el diagnóstico médico. El “Recall” mide la proporción de verdaderos positivos entre todos los casos positivos reales y es útil cuando el coste de los falsos negativos es elevado, como en la detección de fraudes. “F1-score” es la media armónica de “Precision” y” Recall”, y equilibra la compensación entre ambas, es útil cuando tanto la precisión como la recuperación son importantes, y se suele utilizar en problemas de clasificación binaria (Flach & Kull, 2015).

## Matriz de Confusión

La matriz de confusión es un recurso visual y analítico que permite descomponer las predicciones en TP, TN, FP y FN, facilitando así el cálculo de las métricas previamente discutidas (Susmaga, 2004). Según Susmaga (2004), la matriz de confusión es una matriz entera no negativa , donde y y es el número de clases. El elemento representa el número de objetos de la clase que han sido asignados a la clase durante la prueba. Los elementos en la diagonal principal indican una clasificación correcta, mientras que los elementos fuera de la diagonal representan errores del clasificador.

En este contexto, las clases no se observan en el espacio de los atributos que describen los objetos, sino en el espacio que el clasificador implica. Si el clasificador puede discriminar correctamente entre dos clases, se consideran distantes entre sí. En cambio, clases que son mal clasificadas por el clasificador se consideran cercanas. La matriz de confusión, por lo tanto, ofrece una representación efectiva de cómo el clasificador percibe las diferentes clases (Susmaga, 2004).

## Interpretabilidad mediante mapas de atención

Los mapas de atención en visión por ordenador se refieren a una técnica utilizada en modelos de aprendizaje profundo para visualizar qué partes de una imagen están siendo atendidas por el modelo. Los mapas de atención son generados por el modelo y resaltan las regiones de la imagen de entrada que son más relevantes para la salida. Esta técnica se utiliza para mejorar la interpretabilidad de los modelos de aprendizaje profundo y para comprender mejor cómo realizan las predicciones (Ulutan et al., 2020).

# MODELO MATEMÁTICO

La arquitectura Swin Transformer inicia su proceso tomando una imagen RGB de entrada de dimensiones y la transforma en una secuencia de "tokens" que se alimentan en capas subsiguientes de la red. Este proceso consta de dos pasos principales: la División en Parches y la capa de embedding Lineal. Aquí, y son las dimensiones de la altura y la anchura de la imagen, respectivamente, y 3 es el número de canales RGB.

## División de parches

La imagen de entrada se divide en parches no superpuestos. Cada parche se considera un "token" y sus características se establecen como una concatenación de los valores RGB de los píxeles en crudo cuya dimensión es 48, esto se debe a que dimensiones del parche se establecen en 4×4 y cada píxel tiene 3 canales (RGB). Matemáticamente, este proceso se puede representar como la función que extrae un parche específico de la imagen se puede definir de la siguiente manera:

Donde:

* es el parche ubicado en la fila y la columna en la disposición de parches.
* es la función que extrae el parche ubicado en de la imagen .
* es la imagen de entrada de tamaño.
* son las coordenadas iniciales del parche en la fila y la columna en la imagen.
* es una operación que toma una matriz y la convierte en un vector columna.

La dimensión de la concatenación de características del parche se calcula como:

El número total de parches P se calcula como:

Y el número total de características para todos los parches es:

## Capa de embedding lineal (Linear Embedding):

Una vez que los parches son extraídos, cada uno se transforma mediante una capa de proyección lineal. Esta proyección lineal se logra utilizando una convolución 2D con un tamaño de núcleo y un paso igual al tamaño del parche, y un número de canales de salida igual a . El resultado es aplanado y transpuesto para obtener un tensor de forma . La proyección lineal se puede expresar como:

Donde:

* es el embedding correspondiente al parche .
* es la matriz de pesos de la convolución.
* es el vector de sesgo.
* La dimensión de será , que es la dimensión del espacio de embedding.

La operación de convolución 2D utilizada en la capa de embedding lineal es equivalente a aplicar una capa totalmente conectada a cada parche de la imagen. Esto se debe al hecho de que el tamaño del núcleo de la convolución y el paso son iguales al tamaño del parche, lo que resulta en un mapeo uno a uno de cada parche a su correspondiente embedding en el espacio de características de dimensión . La dimensión en la que se proyectan los parches se establece según el tamaño de la arquitectura y puede ser:

* Swin-T: C = 96
* Swin-S: C = 96
* Swin-B: C = 128
* Swin-L: C = 192

## Bloques de Swin Transformer

Swin Transformer introduce la idea de ventanas desplazadas en bloques Swin Transformer sucesivos. Sobre los embeddings se aplican varios de estos bloques. Cada bloque contiene un módulo de atención basada en ventanas (W-MSA) seguido de un perceptrón multicapa (MLP) de dos capas con una función de activación GELU. Antes de cada instancia de W-MSA y MLP, se incorpora una capa de normalización LayerNorm (LN) y posterior a cada módulo, se añade una conexión residual. En bloques subsiguientes, el módulo W-MSA se sustituye por un módulo de atención basada en ventana desplazada (SW-MSA). La complejidad computacional se mantiene lineal al fijar el tamaño de cada ventana (7 por defecto).

### Normalización de Capa (LayerNorm)

La normalización de capas se realiza en la dimensión de las características para cada token y se aplica antes de cada módulo de autoatención y cada MLP. La fórmula matemática para la normalización de capas es:

Donde es el embedding de entrada, y son la media y la varianza a lo largo de la dimensión de las características, es un valor pequeño para evitar la división por cero, y y son parámetros entrenables. El operador denota el producto elemento a elemento.

### Autoatención Basada en Ventanas (W-MSA)

Utiliza un enfoque de atención basado en ventana, donde la atención se calcula solo dentro de cada ventana de embeddings. Para calcular la atención, se utilizan las matrices (query), (key) y (value), que se obtienen a partir de la matriz proyectada. Primero se ejecuta una operación de capa lineal que proyecta el tensor de entrada al espacio tridimensional con dimensión .

donde y son los pesos y sesgos entrenables, respectivamente. Después de la proyección, se reorganiza el tensor para separar las proyecciones y adaptarlo para la atención de múltiples cabezas.

* son las matrices de query, key y value.
* es la dimensión de los vectores y .
* es el sesgo de posición relativa.

### Multi-Layer Perceptron (MLP con GELU)

Después de la atención, los tokens pasan por un perceptrón multicapa (MLP) con dos capas lineales y una activación GELU entre ellas. La fórmula matemática de la capa MLP se puede describir como:

Aquí, y son las matrices de pesos y y son los vectores de sesgo para las dos capas lineales. GELU es la función de activación y Dropout se aplica después de la segunda capa lineal.

### Autoatención Basada en Ventanas Desplazadas (SW-MSA)

Para mitigar las limitaciones de W-MSA, SW-MSA desplaza las ventanas para introducir conexiones entre ventanas diferentes. Cuando se aplican el módulo de atención en ventanas desplazadas, se calcula de manera similar, pero con una máscara adicional para asegurar que solo se consideren los elementos dentro de la ventana desplazada.

### Conexión Residual

Las conexiones residuales se aplican después de cada módulo, después de la capa de atención (W-MSA) y después de la capa MLP, en un bloque Swin Transformer. En las ecuaciones, la conexión residual se denota como el término  *o* .

## Ventanas desplazadas en bloques sucesivos

Las siguientes ecuaciones muestran cómo se calculan las características en bloques sucesivos utilizando diferentes configuraciones de atención basadas en ventanas (Liu et al., 2021).

**,**

**,**

,

Donde y son las características de salida de los módulos (S)W-MSA y MLP, es decir tensores resultantes después de aplicar estas operaciones para el bloque , respectivamente.

## Patch Merging

En la arquitectura Swin Transformer, la resolución de la imagen se reduce en etapas sucesivas, desde la "Etapa 1" hasta la "Etapa 4". En cada etapa, el modelo procesa las características de la imagen a diferentes resoluciones para capturar información jerárquica. Para realizar este proceso, Swin Transformer emplea un método denominado "Patch Merging", este método se aplica para reducir el número de tokens (parches) y, por lo tanto, la resolución de la imagen. Se toma un grupo de parches vecinos de y se concatenan sus características en profundidad. Matemáticamente, este proceso se puede expresar de la siguiente manera:

Donde es la operación de concatenación a lo largo del eje de las características. Dado que cada parche tiene una dimensión , la dimensión del parche fusionado será 4C. Después de la concatenación, una capa lineal se aplica al parche fusionado para reducir su dimensión:

Donde y son la matriz de pesos y el vector de sesgo de la capa lineal, respectivamente. La dimensión del parche después de la capa lineal se establece a .

Esta operación reduce el número de tokens a y la dimensión de cada token a . Posteriormente, los tokens reducidos se alimentan en los bloques de Swin Transformer para transformaciones adicionales. Este proceso se puede repetir en etapas sucesivas (Etapa 3 y Etapa 4) para reducciones adicionales en la resolución y el número de tokens, permitiendo que el modelo capture características a múltiples escalas.

# METODOLOGÍA

**Fase 1:** Análisis de Requerimientos para la Implementación de Swin Transformer en la Detección de Enfermedades en Plantas de Tomate

Esta fase tiene como objetivo comprender los requerimientos esenciales para implementar con éxito la arquitectura Swin Transformers en la detección y clasificación de enfermedades en plantas de tomate. Esto proporcionará una base sólida para el diseño e implementación del sistema.

Mediante:

**Identificación de Requerimientos de Datos:** Se realizará una exhaustiva identificación de los tipos de datos (Imágenes) necesarias para el entrenamiento y validación del modelo Swin Transformer. Esto incluirá imágenes de plantas de tomate enfermas y saludables, metadatos asociados, y cualquier otro dato relevante.

**Recolección y Curación de Datos:** Se procederá a recolectar los datos identificados en la actividad anterior. Además, se llevará a cabo un proceso de limpieza y curación de los datos para garantizar su calidad y coherencia, Usaremos el dataset de PlantVillage ya que proporciona una variedad de clases de enfermedades del tomate y refleja la complejidad de la detección en el campo este modelo este cuenta con nueve clases de enfermedades distintas y una clase sana para la validación del modelo Swin Transformer. Esto proporciona una variedad de clases de enfermedades del tomate y refleja la complejidad de la detección en el campo, las imágenes de plantas de tomate enfermas y saludables.

Como se puede observar este dataset no esta balanceado correctamente ya que las cantidades de imágenes son muy dispersas tienes una cantidad significativamente mayor de muestras en una clase que en otra y esto podría generar un margen de error más alto por esto aplicaremos métodos para balancear un conjunto de datos desequilibrado:

Identificar imágenes duplicadas:

Es importante contar con un buen método para comparar imágenes, especialmente tomando en cuenta su contenido, no tanto su estructura.

Detección (y remoción) de imágenes duplicadas en un conjunto de datos. Para esto usaremos **SIFT (SCALE INVARIANT FEATURETRANSFORM)**

Nombrado así en base a que transforma datos de la imagen en características locales que se obtienen y expresan en coordenadas de la imagen invariantes a la escala. Este es un método que permite detectar puntos característicos en una imagen y luego describirlos, el algoritmo consta de dos partes claramente diferenciadas:

1) Obtención de los puntos característicos

2) Descripción de la región alrededor de cada punto de interés

La obtención de los puntos característicos o puntos de interés (keypoints), consiste en detectar aquellas regiones de la imagen en las que se producen diferencias de gradiente significativas a ambos lados de dicho punto. Lowe propuso calcular un histograma de direcciones del gradiente local alrededor del punto de interés mediante un enfoque de filtrado en cascada que utiliza algoritmos eficaces para identificar ubicaciones candidatas en detalle.

La primera etapa de la detección de los puntos clave consiste en identificar los lugares y las escalas que pueden ser el diferentes vistas del mismo objeto La detección de esos puntos característicos a diferentes escalas requiere crear un espacio-escala, detectar en él puntos de interés, y eliminar aquellos que se consideren poco estables para esto lowe propuso calcular un histograma de direcciones del gradiente local alrededor del punto de interés, además, es invariante a la rotación porque se determina la orientación dominante de los vectores del gradientes.

Las etapas para el cálculo del método son las siguientes:

**Detección de extremos en el espacio-escala** Se buscan puntos de interés en toda la imagen y en todas las escalas consideradas utilizando una diferencia de Gaussianas este ejecuta dos desenfoques Gaussianos en la imagen, con diferentes radios de desenfoque, y sustrae las dos versiones para obtener el resultado final.



**Localización precisa de puntos característicos:** Para cada uno de los puntos de interés anteriores se ajusta un modelo que permite determinar su localización y escala. Además, se seleccionan los puntos característicos, keypoints, eliminando los que están próximos a los bordes o tienen bajo contraste.

**Asignación de la orientación** A cada punto característico se le asigna una o varias orientaciones en función de las direcciones del gradiente (sinónimo de pendiente o inclinación) local. Esta orientación conjuntamente con la ubicación y la escala calculadas anteriormente permiten que el descriptor sea invariante a estas tres situaciones.

**Descripción del punto característico** Alrededor de cada punto característico se miden los gradientes locales de la imagen y se utiliza su histograma para obtener una representación de esa región que es robusta a cambios significativos en la iluminación y a pequeñas distorsiones en la forma.

La distancia a la que están ubicados los objetos afecta a su percepción. Un objeto que se encuentra lejos presenta un tamaño menor que el mismo objeto cuando está situado más próximo para obtener una descripción de los objetos que sea robusta ante estas variaciones en tamaño se utiliza el “espacio-escala”.

Koenderink y Lindeberg mostraron, tras realizar diversas asociaciones razonables, que el único kernel posible para el espacio-escala es la función gaussiana. lo tanto, el espacio escala de una imagen, I(x,y) consiste en una familia de imágenes derivadas, de L(x,y,o) que se obtienen por la convolución de una gaussiana de desviación típica σ y escala variable, G(x,y,o), con la imagen de entrada L(x,y)

Donde \* denota la operacion de convolucion en las coordenadas x,e,y, donde

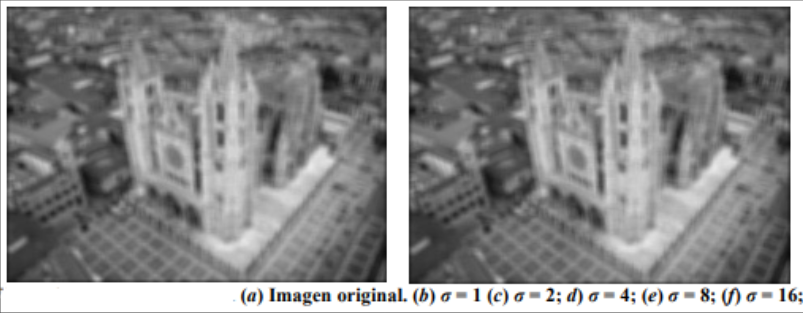
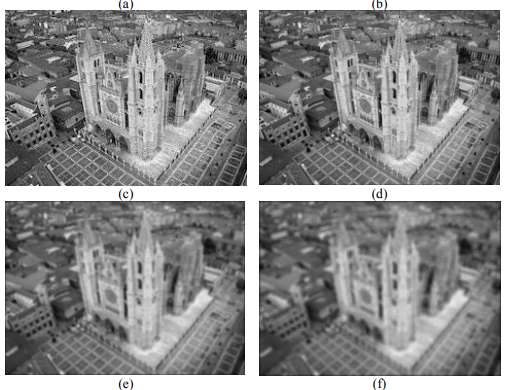


Para detectar de forma eficiente ubicaciones estables de puntos característicos, Lowe propuso utilizar los valores extremos del espacio escala de la diferencia de Gaussianas de la imagen, , que se puede calcular como la diferencia de dos escalas consecutivas separadas por un

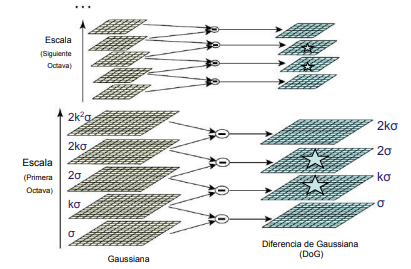
factor multiplicativo constante, k:

Este proceso es particularmente eficiente de calcular, ya que las imágenes suavizadas, , tienen que obtenerse obligatoriamente para la descripción de características en el espacio escala y, por lo tanto, , puede obtenerse simplemente restando las imágenes anteriores.

En SIFT, el espacio escala es fundamental y se extiende mediante la construcción de "octavas", que son espacios de escala reducidos obtenidos al reducir a la mitad las dimensiones de la imagen original. Esto se logra eliminando filas y columnas alternas. Cada una de estas octavas es esencial para la detección de puntos característicos y se obtiene a partir de diferencias Gaussianas (DoG) (es una técnica utilizada para detectar características clave en una imagen, como puntos de interés o bordes. Se basa en la idea de calcular la diferencia entre imágenes suavizadas con diferentes niveles de desenfoque)



Como se explicó anteriormente se generan imágenes progresivamente más suavizadas mediante la convolución con una gaussiana con desviación típica cada vez mayor. La relación entre las desviaciones típicas viene determinada por la constante , de manera que si a la primera imagen se le aplica una , a la segunda se le aplica . Se dice que el conjunto de estas n imágenes con suavizado progresivo pertenecen a la misma octava. Al dividir el tamaño de la imagen a la mitad, aparece la segunda octava, y así sucesivamente.



En la izquierda puede verse el espacio escala donde se representan dos octavas, la primera y la segunda con 5 escalas cada una de ellas. En la parte derecha (columna) del dibujo se ve cómo se obtiene el espacio escala de diferencias de gaussianas. Las estrellas indican las dos únicas imágenes, de cada octava, en las que puede buscarse máximos o mínimos, al ser las únicas que tienen una imagen vecina superior y otra inferior obteniendo como resultados estos guardan la información de las características de las imágenes.

**Definición de Requerimientos de Hardware y Software:** Se determinarán los recursos de hardware y software necesarios para la implementación de Swin Transformer. Esto incluirá la selección de la plataforma de entrenamiento, bibliotecas de software y herramientas de desarrollo.

**Análisis de Métricas de Evaluación:** Se identificarán las métricas de evaluación adecuadas para medir el rendimiento del modelo Swin Transformer en la detección de enfermedades en plantas de tomate. Esto puede incluir métricas como precisión, recall, F1-score, etc.

**Definición de Requerimientos de Interfaz de Usuario:** Si se prevé una interfaz de usuario para interactuar con el modelo, se definirán los requerimientos funcionales y de diseño de esta interfaz.

**Fase 2: Preparación de datos**

1. **Selección del dataset**: Utilizar el dataset de PlantVillage, enfocándose en las clases relacionadas con el tomate.
2. **Preprocesamiento de datos**: División del dataset en conjuntos de entrenamiento, validación y prueba
3. **Aumento de datos**: Técnicas como rotación, cambio de escala y recorte para aumentar la cantidad de datos de entrenamiento.
4. **Balanceo de clases**: Sobremuestreo de clases subrepresentadas y submuestreo de clases sobrerrepresentadas.
5. **Normalización de imágenes.**

**Fase 3: Desarrollo del modelo**

1. **Arquitectura del modelo**: Implementación de la arquitectura Swin Transformer.
2. **Configuración de parámetros**: Establecimiento de parámetros para el entrenamiento como tasa de aprendizaje, tamaño del lote y número de épocas.
3. **Entrenamiento del modelo**: Utilizar el conjunto de entrenamiento para entrenar el modelo.
4. **Validación del modelo**: Utilizar el conjunto de validación para ajustar hiperparámetros y evitar el sobreajuste.
5. **Pruebas de rendimiento**: Utilizar el conjunto de prueba para evaluar el rendimiento del modelo en la métrica Accuracy tanto para el modelo general como para cada clase utilizando una matriz de confusión.
6. **Interpretación de resultados**: Utilizar técnicas como mapas de calor de atención para entender cómo el modelo toma decisiones.

**Fase 4: Comparación y evaluación**

1. **Implementación de modelos de CNN para comparación**: Desarrollo de modelos basados en arquitecturas CNN como ResNet, DenseNet y EfficientNet.
2. **Evaluación de modelos**: Comparar el Accuracy entre el modelo basado en Swin Transformer y los modelos basados en CNN.

**Fase 5: Documentación y comunicación**

1. **Elaboración de informe técnico**: Documentar la metodología, implementación y resultados en un informe técnico.
2. **Preparación de presentación**: Crear material para presentación de resultados.
3. **Diseminación**: Publicación en revistas científicas y/o presentación en conferencias relevantes.

**Fase 6: Trabajo futuro**

1. **Identificación de limitaciones**: Establecer las limitaciones del estudio actual.
2. **Planificación de mejoras**: Proponer futuras líneas de investigación o aplicaciones prácticas del modelo desarrollado.

# REFERENCIAS

Collins, E.J.; Bowyer, C.; Tsouza, A.; Chopra, M. Tomatoes: An Extensive Review of the Associated Health Impacts of Tomatoes and Factors That Can Affect Their Cultivation. Biology 2022, 11, 239.

Agarwal, Mohit, et al. “ToLeD: Tomato Leaf Disease Detection Using Convolution Neural Network.” *Procedia Computer Science*, vol. 167, no. 293–301, 2020, pp. 293–301, https://doi.org/10.1016/j.procs.2020.03.225.

Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture, 187(106279). https://doi.org/10.1016/j.compag.2021.106279

Chen, H.-C., Widodo, A. M., Wisnujati, A., Rahaman, M., Lin, J. C.-W., Chen, L., & Weng, C.-E. (2022). AlexNet Convolutional Neural Network for Disease Detection and Classification of Tomato Leaf. Electronics, 11(6), 951. https://doi.org/10.3390/electronics11060951

Bhandari, M., Shahi, T. B., Neupane, A., & Walsh, K. B. (2023). BotanicX-AI: Identification of Tomato Leaf Diseases Using an Explanation-Driven Deep-Learning Model. Journal of Imaging, 9(2), 53. https://doi.org/10.3390/jimaging9020053

FAO. (2021). Crops and livestock products. Www.fao.org. https://www.fao.org/faostat/en/#data/QCL

Caruso, A. G., Bertacca, S., Parrella, G., Rizzo, R., Davino, S., & Panno, S. (2022). Tomato brown rugose fruit virus: A pathogen that is changing the tomato production worldwide. Annals of Applied Biology, 181(3), 258–274.

Panno, S., Davino, S., Caruso, A. G., Bertacca, S., Crnogorac, A., Mandić, A., Noris, E., & Matić, S. (2021). A Review of the Most Common and Economically Important Diseases That Undermine the Cultivation of Tomato Crop in the Mediterranean Basin. Agronomy, 11(11), 2188. https://doi.org/10.3390/agronomy11112188

Singh, V. K., Singh, A. K., & Kumar, A. (2017). Disease management of tomato through PGPB: current trends and future perspective. 3 Biotech, 7(4). https://doi.org/10.1007/s13205-017-0896-1

Zahangir Alom, Taha, T. M., Yakopcic, C. G., Westberg, S., Sidike, P., Nasrin, Mst Shamima, Esesn, V., Abdul, & Asari, V. K. (2018). The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. ArXiv E-Prints, 2. https://doi.org/10.48550/arxiv.1803.01164

Boulent, J., Foucher, S., Théau, J., & St-Charles, P.-L. (2019). Convolutional Neural Networks for the Automatic Identification of Plant Diseases. Frontiers in Plant Science, 10. https://doi.org/10.3389/fpls.2019.00941

Li, E. Y. (2020, August 1). 10 Papers You Should Read to Understand Image Classification in the Deep Learning Era. Medium. https://towardsdatascience.com/10-papers-you-should-read-to-understand-image-classification-in-the-deep-learning-era-4b9d792f45a7

Rawat, V., Singh, N., Kaur, B., & Bora, S. (2022). Tomato Leaf Disease Classification Using Supervised Learning Techniques: Contrasting Analysis. 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM), 1(2642-7354). https://doi.org/10.1109/icaccm56405.2022.10009617

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, pp. 9992-10002, doi: 10.1109/ICCV48922.2021.00986.

Boulent, J., Foucher, S., Théau, J., & St-Charles, P. L. (2019). Convolutional neural networks for the automatic identification of plant diseases. Frontiers in plant science, 10, 941.

Gadade, H. D., & Kirange, D. D. (2020). Machine learning approach towards tomato leaf disease classification. International Journal of Advanced Trends in Computer Science and Engineering, 9(1), 490-495.

Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture, 187, 106279.

Uppada, R., & Kumar, D. R. (2023). Computer-aided fusion-based neural network in application to categorize tomato plants.Springer-Verlag London Ltd., part of Springer Nature 2023, 1-9.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021, July). Training data-efficient image transformers & distillation through attention. In International conference on machine learning (pp. 10347-10357). PMLR.

Flores, C. (2012). Enfermedades de tomate: Guía de consulta (S. Buono & S. Giorgioni, Eds.; 1st ed.). Ediciones Instituto Nacional De Tecnología Agropecuaria De Argentina INTA.

Bernal, R. (2010). Enfermedades de tomate (lycopersicum esculentum mill.) en invernadero en las zonas de salto y bella unión. Serie Técnica. INIA. Montevideo, Editorial Hemisferio Sur SRL, 181, 1–71.

Thangaraj, R., Anandamurugan, S., Pandiyan, P., & Kaliappan, Vishnu Kumar. (2022). Artificial intelligence in tomato leaf disease detection: a comprehensive review and discussion. *Journal of Plant Diseases and Protection*, *129*, 3.

Garcia, J. (2016). A review on the main challenges in automatic plant disease identification based on visible range images. Biosystems Engineering, 144, 52–60.

Valdivieso. (2021). Estudio comparativo de las técnicas de inteligencia artificial para el diagnóstico de enfermedades en la agricultura.

Blancard, D. (2011). Enfermedades del tomate. Mundi-Prensa Libros.

Verma, S., Chug, A., & Singh, A. P. (2018). Prediction models for identification and diagnosis of tomato plant diseases. IEEE, 1557–1563.

Corrales Paredes, J. (2021). *Uso de depredadores, parasitoides y entomopatógenos para el control biológico de plagas y enfermedades de hortalizas con énfasis en tomate (Solanum lycopersicum l.)* (p. 105) [MSc Thesis].

Gómez-Camperos, J., Jaramillo, H., & Guerrero-Gómez, G. (2022). Técnicas de procesamiento digital de imágenes para detección de plagas y enfermedades en cultivos: una revisión. Ingeniería Y Competitividad, 24, 1.

Guennouni, M., Admou, B., Bourrhouat, A., El Khoudri, Noureddine, Zkhiri, W., Talha, I., Hazime, R., & Hilali, A. (2022). Knowledge and practices of food safety among health care professionals and handlers working in the kitchen of a moroccan university hospital. Journal of Food Protection, 85, 4.

Uçar, A., Yilmaz, M. V., & Çakiroglu, Funda Pinar. (2016). Food safety–problems and solutions. Significance, Prevention and Control of Food Related Diseases, 3.

Nanehkaran, Y., Zhang, D., Chen, J., Tian, Y., & Al-Nabhan, N. (2020). Recognition of plant leaf diseases based on computer vision. In Journal of Ambient Intelligence and Humanized Computing (pp. 1–18). Springer.

Khan, R., Kumar, S., Dhingra, N., & Bhati, N. (2021). The use of different image recognition techniques in food safety: a study. Journal of Food Quality, 2021, 1–10.

Mahadevkar, S. V., Khemani, B., Patil, S., Kotecha, K., Vora, D. R., Abraham, A., & Gabralla, L. A. (2022). A Review on Machine Learning Styles in Computer Vision—Techniques and Future Directions. IEEE Access, 10, 107293–107329. https://doi.org/10.1109/access.2022.3209825

Sood, S., & Singh, H. (2021). Computer vision and machine learning based approaches for food security: A review. Multimedia Tools and Applications, 80, 18.

Jamil, S., Md. Jalil Piran, & Kwon, O.-J. (2023). A Comprehensive Survey of Transformers for Computer Vision. Drones, 7(5), 287–287. https://doi.org/10.3390/drones7050287

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6(1). https://doi.org/10.1186/s40537-019-0197-0

Xiao, Z., Wang, J., Han, L., Guo, S., & Cui, Q. (2022). Application of machine vision system in food detection. *Frontiers in Nutrition*, *9*, 888245.

Papers With Code. (2023). *Papers with Code - ImageNet Benchmark (Image Classification)*. Paperswithcode.com. https://paperswithcode.com/sota/image-classification-on-imagenet. (Accessed on September 2023).

Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A.-M., & Wang, X. (2021). Computer vision techniques in construction: a critical review. *Archives of Computational Methods in Engineering*, *28*, 3383–3397.

Ouhami, M., Hafiane, A., Es-Saady, Y., El Hajji, Mohamed, & Canals, R. (2021). Computer vision, IoT and data fusion for crop disease detection using machine learning: A survey and ongoing research. *Remote Sensing*, *13*, 13.

Szeliski, R. (2020). *Computer Vision: Algorithms and applications.* (2nd ed.). Springer Nature.

Vocaturo, E. (2021). Image classification techniques. In *Handbook of Research on Disease Prediction through Data Analytics and Machine Learning* (pp. 22–49). IGI Global.

Sanghvi, K., Aralkar, A., Sanghvi, S., & Saha, I. (2020). A survey on image classification techniques. In *Available at SSRN 3754116*.

Shah, U., & Harpale, A. (2018). A review of deep learning models for computer vision. In *2018 IEEE Punecon* (pp. 1–6). IEEE.

Amjoud, A. B., & Amrouch, M. (2023). Object detection using deep learning, CNN and vision transformers: A review. *IEEE Access*, *11*(35479-35516).

Cappi, C., Chapdelaine, C., Gardes, L., Jenn, E., Lefevre, B., Picard, S., & Soumarmon, T. (2021). Dataset definition standard (dds). In *arXiv preprint arXiv:2101.03020*.

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, *7*. https://doi.org/10.3389/fpls.2016.01419

Khosla, C., & Saini, B. S. (2020). Enhancing performance of deep learning models with different data augmentation techniques: A survey. *IEEE*, 79–85.

Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., & Shen, F. (2022). Image data augmentation for deep learning: A survey. In *arXiv preprint arXiv:2204.08610*.

Dyk, V., & Meng, X.-L. (2001). The art of data augmentation. *Journal of Computational and Graphical Statistics*, *10*, 1.

Flach, P. A., & Kull, M. (2015, December). Precision-recall-gain curves: PR analysis done right. *NIPS*. https://api.semanticscholar.org/CorpusID:937625

Susmaga, R. (2004). Confusion matrix visualization. In A. Kłopotek, Mieczysław, T. Wierzchoń, Sławomir, & K. Trojanowski (Eds.), *Springer, Berlin, Heidelberg* (pp. 107–116). Springer Berlin Heidelberg.

Ulutan, O., Rallapalli, S., Srivatsa, M., Torres, C., & Manjunath, B. (2020). Actor conditioned attention maps for video action detection. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 527–536.