

UNIVERSIDAD DE COSTA RICA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS DE LA
COMPUTACIÓN E INFORMÁTICA

CI-0163 Análisis de grandes volúmenes de datos

Prof. Dr. Allan Berrocal

Investigación Aplicada

Elaborado por:

Ayales León Eduardo B90833

Quesada Luis Carlos B65580

Ureña Torres Jose David B88044

2022

Aplicación web para clasificación de plantas utilizando modelo basado en VGG16

Introducción

Actualmente existe mucho avance en el área de las ciencias de datos, sin embargo, en muchas ocasiones no se le da prioridad a la usabilidad de los sistemas desarrollados o descubrimientos encontrados. De manera que los usuarios no técnicos, es decir, sin conocimientos en ciencias de datos o programación de computadoras, no pueden hacer uso de ese conocimiento. Este trabajo plantea aportar en el área de la visualización de la información y la interactividad con usuarios finales.

En este trabajo de investigación consiste en un prototipo de aplicación para permitir a los usuarios realizar tareas de clasificación de cinco tipos de plantas a partir de imágenes.

Problema o pregunta de investigación

Crear un prototipo de aplicación web con interfaz gráfica que facilite la interacción con el modelo de clasificación de plantas con un usuario no técnico.

Descripción

Se pretende implementar una aplicación web que permita a los usuarios subir imágenes y clasificarlas dentro de cinco categorías. Esta interfaz permitirá subir una imagen y al presionar un botón se le mostrará un resultado de clasificación dentro de alguna de cinco categorías de plantas.

El modelo que se utilizará será una VGG16 [1] modificada a partir de técnicas de transferencia de aprendizaje esto con el fin de utilizar características previamente entrenadas, además obteniendo una reducción considerable de tiempo de entrenamiento.

La arquitectura web consistirá en un endpoint que devuelva la página web donde se pueda subir una imagen y enviarla al servidor. El servidor consultará al modelo previamente cargado y se procesará la salida de este para expresarlo de una manera entendible para el usuario no técnico.

Justificación

En el mundo de la ciencia de datos, existen muchos modelos para resolver un mismo tipo de problemas, por lo que no siempre es claro cuál es el indicado. De lo anterior, nace la idea de comparar el rendimiento de dos modelos de clasificación: pero no solo dando énfasis al

aspecto técnico o teórico, sino también a la visualización de resultados para el usuario no técnico y generar valor para los interesados.

La Inteligencia Artificial puede ser una herramienta de apoyo en el ámbito laboral. Sin embargo, se debe trabajar aspectos de Interacción Humano Computador para que el usuario pueda utilizarlo como en su flujo de trabajo.

No es necesario que los usuarios entiendan en profundidad conceptos como modelos, redes neuronales o modificación de hiperparámetros. Para ellos debería mostrarse como una herramienta de trabajo, entretenimiento o educativa donde la interacción entre ellos y la inteligencia artificial sea amena y sencilla, sin carga cognitiva innecesaria.

Objetivos

Principal:

- Desarrollar una prototipo de aplicación web para facilitar el uso de modelos de aprendizaje máquina para una tarea de clasificación de plantas.

Específicos:

1. Construir modelo basado en la arquitectura VGG16, mediante transferencia de conocimiento, para la clasificación de plantas.
2. Medir la efectividad del modelo para la clasificación de plantas utilizando las métricas accuracy, F1 Score, precision y recall.
3. Desarrollar un prototipo de aplicación web para permitir a un usuario no técnico interactuar con el modelo.
4. Aplicar al menos 3 principios de diseño de Jacob Nielsen para mejorar la usabilidad del prototipo.

Metodología

El dataset que se utilizará es un dataset de flores [5] que contiene 5 clases, cada una con más de 700 imágenes de cada clase. Este dataset se obtuvo de Kaggle [5].

Para el ambiente de trabajo se utilizará Jupyter Notebook [4]. En cuanto a equipo de cómputo se utilizará una PC normal.

Se planea utilizar un modelo base construido a partir de una red VGG16 utilizando una técnica de transferencia de aprendizaje. De manera paralela se construirá una aplicación web que facilitará el uso de los modelos de clasificación para un público no técnico.

Las redes neuronales VGG son modelos basados en redes neuronales convolucionales, estos modelos son comúnmente utilizados para la clasificación de imágenes.

El modelo VGG16 fue creado para clasificar entre 1000 diferentes clases por lo que tiene muy buenas capacidades para extraer características de una imagen, existen múltiples implementaciones de este modelo que ya se encuentran pre entrenadas, sin embargo para el caso de esta investigación solo se busca clasificar 5 clases que no necesariamente se encuentran dentro de las utilizadas para entrenar el modelo base, sin embargo es posible tomar ventaja del poder de extracción de características pre entrenado del modelo VGG16 utilizando técnicas de transferencia de aprendizaje.

La biblioteca de aprendizaje mecánico PyTorch incluye dentro de sus modelos pre entrenados una versión de VGG16, este modelo puede ser modificado en su sección de clasificación con el fin de aprovechar las capas convolucionales para extraer características que se encuentran pre entrenadas.

Para esta investigación se aplicará técnicas de transferencia de aprendizaje sobre un modelo VGG16 creado con la biblioteca PyTorch, esto se realizará congelando las capas de extracción de características para evitar su reentrenamiento y truncando su capa de clasificación, cambiando la capa de salida por una capa densa de 5 neuronas de salida, esto de acuerdo a la cantidad de clases que se clasificará en este caso.

Para la parte de la interfaz de usuario:

Para esto se implementará una pantalla de inicio donde se le dé la bienvenida al usuario y se le invite a utilizar la aplicación. Esta pantalla permitirá al usuario cargar una imagen y al presionar un botón se le mostrará el resultado de la clasificación. Adicionalmente habrá una pantalla de cómo funciona que le mostrará una pequeña explicación, sin entrar en mucha complejidad, de cómo funciona este proyecto. Finalmente, una pantalla con la información de los participantes y la razón de ser del proyecto. De manera que si un usuario entra, pueda saber de quién es la página y por qué existe.

En cuanto a las herramientas se utilizará React para el frontend, el cual es un framework de JavaScript ampliamente utilizado en la actualidad.

Para la parte del servidor:

Se usará Flask para la parte de predecir la clase de las plantas. Para esto, dentro de un archivo .py se creará una ruta POST donde se recibe la imagen a clasificar desde la interfaz web. Esta luego es convertida a una imagen de la librería PIL (Python Imaging Library). Finalmente, se hará una predicción de esta usando una clase donde fue previamente cargado el modelo entrenado.

Delimitación

Para la aplicación web se se establecen las siguientes limitaciones:

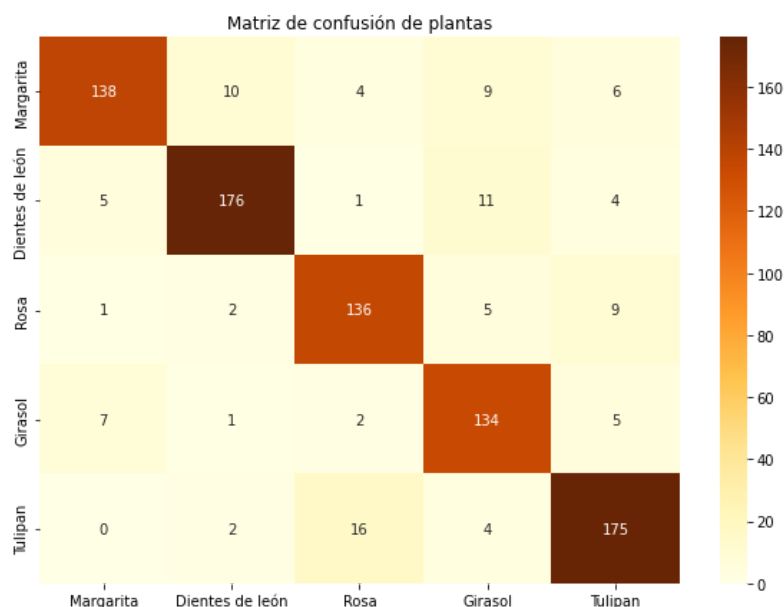
- Se limitarán los tipos de plantas a únicamente margaritas, rosas, dientes de león, tulipanes y girasoles.
- Solo se podrá clasificar una imagen a la vez.
- El endpoint es público, no requiere autenticación.

Desarrollo

Aplicando técnicas de Transferred Learning sobre un modelo VGG16 se eliminó la capa final del modelo pre entrenado, cambiando esta por una capa densa de salida 5 ya que esta es la cantidad de clases a predecir en este proyecto, además como se busca obtener las probabilidades para cada una de las imágenes, se le añade una capa de salida softmax que nos permite obtener la densidad de la probabilidad.

El rendimiento del modelo se mide en 4 distintas métricas, *accuracy*, *precision*, *recall* y *F1 Score*, se utilizaron estas métricas ya que son las usuales para medir el rendimiento y calidad de un modelo, estas métricas pueden ser obtenidas fácilmente con una matriz de confusión, ya que estas son derivables de los datos que la misma nos provee.

Se realizó mediciones sobre un conjunto de evaluación ya que se deseaba observar el comportamiento del modelo sobre datos que no se hubieran analizado con anterioridad, a partir de la clasificación dada sobre este conjunto se construye una matriz de confusión que se puede observar a continuación.



Utilizando la figura anterior se pueden derivar las siguientes métricas, se puede notar que el modelo alcanza un 0.87 de accuracy, esto nos indica que el modelo puede en la mayoría de casos, clasificar de manera correcta la flor cargada. También se puede observar como el modelo presenta un 0.87 de precisión, lo que indica que por cada 100 predicciones de una clase, 87 de éstas serán clasificadas de manera correcta. El valor para recall es también de 0.87 lo que indica que un alto porcentaje de las predicciones fue correcto y finalmente podemos ver que la métrica de F1 Score es de 0.87, siendo esta un promedio entre precision y recall permite ver que el modelo en términos generales es bueno.

Metricas del modelo	
Accuracy:	0.8794
Precision:	0.8784
Recall:	0.8792
F1 Score:	0.8794

Posteriormente se construyó un API Rest con Flask con un endpoint para realizar peticiones. Estas peticiones se realizan por medio de HTTP con un método POST. El método de encoding utilizado fue form data, esto para enviar con facilidad la imagen desde el cliente del sistema. Dentro del servidor se toma esta imagen y se procesa para ser procesada por el modelo predictivo. Una vez el modelo finaliza la predicción se retorna la respuesta al usuario.

En cuanto a la interfaz de usuario se construyó un sitio web con React.js. Adicionalmente se usó React Bootstrap el cual es una implementación de la librería Bootstrap para React. Se usó una paleta de colores basada en verdes, esto porque el verde se asocia a la naturaleza.

En cuanto a la funcionalidad se agregó tres secciones del sitio. El primero es la sección de “Inicio” donde se encuentra la funcionalidad principal con un cargador de imágenes y las instrucciones del sitio. El usuario puede cargar una imagen y al presionar un botón el sitio le mostrará el resultado de su predicción. En la sección de instrucciones se agregaron imágenes de ejemplo para que el usuario pueda practicar.

La segunda sección se llama “¿Cómo funciona?”. El objetivo de esta sección es mostrar contenido relacionado al algoritmo utilizado y qué significa aprendizaje de máquina, pero sin recurrir a tecnicismos muy complejos. Esto dado que no es el objetivo del sitio.

Finalmente, hay una sección de participantes donde se puede ver la foto del perfil de Github de los autores de este trabajo de investigación.

La investigación se enfocó en los siguientes principios de usabilidad de Jacob Nielsen:

- Diseño estético y minimalista: Se logró una interfaz con solo los elementos necesarios para su funcionamiento, pero con una apariencia y colores de la web moderna. Además, como se está usando una librería de componentes reactivos, la página no tiene necesidad de recargar para mostrar el estado del sistema.
- Ayuda y documentación: Se añadió una sección de instrucciones para que el usuario pueda ver cómo interactuar con el sitio. Además, la sección de “¿Cómo funciona?” invita al usuario a conocer más al respecto si está interesado.
- Consistencia y estándares: La consistencia y sistema de diseño se mantiene a lo largo de las tres secciones del sitio previamente descritas.

Discusión

Se ha visto en la sección de desarrollo que el modelo obtuvo buenos resultados para la clasificación de plantas, como se mencionaba anteriormente este modelo se construye utilizando técnicas de transferencia de aprendizaje, el modelo base VGG16 es un modelo que alcanza métricas muy altas por lo que no es raro que al utilizar sus capacidades de extracción de características sea posible entrenar un buen clasificador para 5 categorías.

El modelo se adapta bastante, puede clasificar imágenes que no ha visto previamente con bastante certeza. Además, reacciona bien ante cierta cantidad de ruido en la imagen, por ejemplo cuando hay personas o animales en las fotos junto con las plantas. Se realizaron pruebas para clasificar una imagen con múltiples clases de plantas a la vez, es posible clasificar únicamente a una de estas sin embargo al tener múltiples posibilidades la densidad de la probabilidad para la clase final será menor a una clasificación con una sola clase de flor.

Dado la gran cantidad de tecnologías web existentes, elegir una u otra no fue una tarea sencilla: por un lado, tecnologías como Flask ofrecen renderizado del lado del servidor mediante un sistema de plantillas. Este ofrece sencillez a la hora de usarlo, por lo que es una buena opción si el contenido no debe actualizarse constantemente como lo sería un blog con publicaciones.

En contraste, existen marcos de desarrollo los cuales se basan en ser reactivos, esto ofrece una experiencia más interactiva al usuario, eliminando la necesidad de recargar la página para ver actualizaciones. Entre estos se encuentran React, Angular, VueJS, entre otros. Para esta investigación, se decidió analizar la opción de usar React dado que es muy flexible y ofrece múltiples herramientas para facilitar la programación de las aplicaciones web. Por defecto, utiliza client side rendering el cual permite tener mayor control de las animaciones y los cambios de estado de la aplicación. Además no necesita recargar la página constantemente ya que permite hacer peticiones asíncronas con mucha facilidad y flexibilidad.

Como la experiencia de usuario es algo esencial en cualquier aplicación web, y en especial para una dirigida a un grupo de usuarios no técnicos, es necesario que esta sea intuitiva a la hora de ser utilizada. Para esto, se decidió que era necesario basarse en alguna guía de HCI conocida en la industria como lo son los principios de Jakob Nielsen. Se intentaron aplicar 3 principios: el diseño estético y minimalista, la ayuda y la consistencia. Es necesario hacer énfasis en la palabra *intentar*, dado que no existió la posibilidad de probar si, en efecto, se cumplieron estos 3 principios, por factores como falta de tiempo para realizar pruebas con usuarios reales y falta de expertos en el área.

En cuanto a la interfaz se logró que fuera minimalista en cuanto a los elementos a mostrar. Los resultados de las clasificaciones se muestran de manera interactiva y con ciertas animaciones para que la experiencia de usuario sea buena.

Referencias consultadas

- [1] Team, K., 2022. *Keras documentation: VGG16 and VGG19*. [online] Keras.io. Available at: <<https://keras.io/api/applications/vgg/>> [Accessed 16 June 2022].
- [2] Robots.ox.ac.uk. 2022. *Visual Geometry Group - University of Oxford*. [online] Available at: <<https://www.robots.ox.ac.uk/~vgg/data/flowers/17/>>
- [3] Team, K., 2022. *Keras: the Python deep learning API*. [online] Keras.io. Available at: <<https://keras.io/>>
- [4] Jupyter.org. 2022. *Project Jupyter*. [online] Available at: <<https://jupyter.org/>>
- [5] Kaggle.com. 2022. *Flowers Recognition*. [online] Available at: <<https://www.kaggle.com/datasets/alxmamaev/flowers-recognition>> [Accessed 16 June 2022].