

Reconocimiento de cultivos, procesamiento y parametrización de riego

Amato Luciano, Borda Jonathan, Lancuba Emiliano, Gomez Nicolas, Olivera
Florenia.

¹Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florenio Varela 1903 - San Justo, Argentina
luchiiamato@gmail.com, jonathanmatiasborda@gmail.com, lancuba.emiliano@gmail.com,
nneegomez@gmail.com, o.florenia123@gmail.com

Resumen. En la actualidad, existen diversas metodologías y algoritmos conocidos de detección de objetos y reconocimiento de imágenes. A pesar del avance y evolución de estos con la llegada de las nuevas tecnologías, aun no se consigue un algoritmo que se adapte completamente a todos los puntos deseables de un sistema o producto (Costo, tiempo y alcance). En este documento, particularmente, se busca demostrar teóricamente la implementación del algoritmo SIFT de reconocimiento de imágenes con el objetivo de identificar cultivos particulares, procesarlos y mejorar la precisión en el riego sobre el cultivo identificado. Adicionalmente, se busca embeber el proyecto en un ecosistema IOT el cual beneficiara y brindara soporte al modulo de reconocimiento, aumentando así, las prestaciones que SmartGarden puede ofrecer al usuario. Motivado a la implementación de esta estructura nos encontraremos con la necesidad de cubrir diversas aristas y realizar modificaciones a gran escala en la estructura actual del proyecto, las cuales serán abordadas en posteriores secciones.

Palabras claves: SIFT, reconocimiento de imágenes, IOT, MIMD, MPI.

1 Introducción

Actualmente SmartGarden se compone de un sistema embebido y una aplicación móvil. El sistema embebido consiste en un dispositivo de riego autónomo, encargado de obtener valores de su entorno, realizar cálculos internos y establecer la intensidad de agua involucrada en un riego. La función de auto aprendizaje es la principal ventaja con respecto a la competencia: el sistema aprende de su entorno, y mejora con el transcurso del tiempo, la precisión de la intensidad de agua destinada al riego, y, por consiguiente, la calidad del mismo. Además, posee la tecnología apropiada para disponer de una comunicación por medio del protocolo Bluetooth, que, utilizando una aplicación como medio, permite establecer conexiones con *smartphones* que posean sistema operativo Android (API mayor o igual a 19 -Android 4.4-), permitiendo el intercambio de información entre estos. Esta aplicación posee funcionalidades que

abarcan tanto consulta de datos/estado como implantación de tareas manuales al sistema embebido, es decir, es posible consultar datos registrados, y -por medio del uso de sensores del dispositivo móvil- establecer directrices al dispositivo, tales como, iniciar registro de entorno, iniciar riego, detener riego, entre otros.

El objetivo del desarrollo e implementación del módulo de reconocimiento de cultivos es otorgar una nueva funcionalidad orientada a profundizar y mejorar la inteligencia del modelo. Mediante este, se lograran identificar objetos particulares, los cuales variaran i) el nivel de intensidad del agua destinada al riego ii) el tiempo de riego sobre los objetos. Estas modificaciones en los parámetros de riego serán procesadas en los momentos previos al pautado según la función de aprendizaje automático de SmartGarden, mejorando así la calidad y el aprovechamiento de los recursos por parte del dispositivo. En palabras más simples, este nuevo módulo permitiría detectar un cultivo o conjunto de cultivos (árbol, arbusto, flores, hierba, etc) los cuales requieran mayor o menor magnitud de agua por tasa de tiempo.

Actualmente existen aplicaciones y sistemas que obtienen provecho de tecnologías de reconocimiento de imágenes, generalmente, utilizadas para detectar objetos o incluso rostros. Algunos ejemplos generales de uso: La aplicación de Ebay que permite buscar artículos usando la cámara, una red neuronal que convierte fotos en tonos negros en imágenes brillantes o la inteligencia artificial de Facebook, que interpreta las fotografías que suben los usuarios para adecuar la publicidad dirigida a cada cuenta.

2 Desarrollo

A la hora de implementar este módulo, se debe comenzar con las modificaciones lógicas e infraestructurales- a realizar en el sistema embebido. El módulo añade una cámara al dispositivo (*endpoint*) la cual será la encargada de obtener los datos del entorno, previo al riego, capturando imágenes del mismo. Luego, mediante la conexión con la aplicación móvil el dispositivo SmartGarden se encargará de enviar estos datos de entorno a la misma para su tratamiento.

La aplicación recibirá las visuales requeridas, y realizará una serie de validaciones, tales como, nitidez de imágenes o cantidad de capturas tomadas, entre otras. Finalmente, enviará el paquete de información a un *datacenter* de servidores externos, pertenecientes a SmartGarden, a través de ethernet. Estos servidores externos son los que, implementando SIFT junto a otras tecnologías, obtendrán los datos necesarios de intensidad y tiempo de riego sobre los cultivos, interactuando con una base de datos Firebase, la cual se retroalimentará de cada nuevo reconocimiento realizado por los SmartGarden. Como ya se ha mencionado en el documento, SIFT será la técnica utilizada para identificar un cultivo como tal. Entre sus ventajas principales -y la principal desde el punto de vista de SmartGarden- se encuentra la gran performance que sostiene ante efectos lumínicos altos, tales como, alta y baja luminosidad, sombras, etc. Esta es una característica importante, debido a que los *endpoints* se encuentran expuestos a condiciones luminosas cambiantes. Por otra parte, entre las desventajas de SIFT, encontramos que el tiempo de procesamiento suele dispararse a niveles muy elevados. Este algoritmo se puede explicar en cuatro pasos:

- **Detección de extremos en el espacio-escala:** Se busca dentro de la imagen puntos los cuales no dependan de la rotación, escalado y traslación de la imagen.
- **Localización exacta de puntos clave:** Se deben eliminar aquellos que no sean estables a cambios de iluminación y ruido.
- **Asignación de orientación:** Mediante la asignación de una orientación a cada punto de la imagen basada en características locales de la misma, los puntos clave pueden ser descriptos relativos a estas orientaciones y de esta manera lograr características invariantes a las rotaciones.
- **Descriptor de puntos clave:** Determinar para cada punto clave un descriptor (ubicado en la Firebase) relativamente invariante a cambios de iluminación y afinidades.

Motivados en agilizar este proceso de detección SIFT, se implementara MPI con el objetivo de realizar procesamiento distribuido entre todos los servidores del *datacenter* previamente mencionado. Desde SmartGarden creemos que si bien MPI otorga un gran salto en cuanto a performance, equiparando así las perdidas sufridas por la ejecución de SIFT, no sera suficiente para que se logre responder a las solicitudes en el tiempo esperado. Es por esto que, además, se optara por implementar el paradigma MIMD (*Multiple Instruction Multiple Data*) complementando a MPI, y logrando así, una mayor eficiencia en las detecciones de puntos clave y descriptores. Cabe destacar que al construir un ecosistema de IOT las solicitudes por tasa de tiempo tendrán un valor elevado y deberán ser respondidas con la en un tiempo establecido. En resumen, MPI nos permitirá distribuir la carga de los datos entre los servidores y MIMD, a agilizar el procesamiento en cada servidor. Si bien sabemos que SIFT requiere tiempo de procesamiento alto, utilizando la combinación de MPI + MIMD obtendremos un equilibrio necesario para obtener los tiempos de respuesta esperados. *Nota: se debe implementar la arquitectura requerida en los servidores del datacenter para poder utilizar MIMD.*

3 Explicación del algoritmo

El siguiente pseudocódigo buscara mostrar la implementación del algoritmo SIFT, junto con el uso de MPI + MIMD para la distribución de carga. No se abordarán las implementaciones del sistema embebido. No se abordarán las implementaciones de la aplicación móvil a excepción del desarrollo del módulo de recepción de información desde los servidores externos.

Servidores Externos – SIFT, MPI+MIMD.

Recibir información de Mobile y envió
<pre>#include "/usr/local/include/mpi.h" #include "/usr/local/include/mimd.h" images[] record; int i=0; string usr=root, pass=root; //Usuario y pass para todos los servidores que conforman la red. string[] hosts, V;</pre>

```

get_hosts_network(hosts); //Obtengo los hosts de la red de servidores.
record[] = receive_images(emisor); //Recibo capturas de entorno de mobile a través de
internet.
WHILE ( i < record[].length){
    image_read(img);
    MIMD_define(); //Definición de parámetros de MIMD.
    MIMD_send_directives (hosts); //Envío de parámetros de MIMD a servidores.
    involucrados
    MPI_init(hosts, usr, pass); //Comienzo sector de paralización.
    MPI_Comm_images(MPI_COMM_WORLD, &record); //paralización de
procesamiento de vector de imágenes.
    //Se realiza comunicación entre servidores con primitivas MPI_send() y MPI_rcv();
    Get_orientation(x);
    Create_dog(y);
    ...//mas información para construir descriptor
    V[i] = Get_points(x, y, ...); //Obtengo los puntos clave
    match_database(DB, FIREBASE_USER, FIREBASE_KEY, V[i]); //Busco el descriptor
    en la base de datos
    MPI_Finalize(); //Termino paralización
    i++; //sigue con la siguiente imagen
}
EnvioSolicitudDeInfoLista(emisor); //Se le avisa al emisor de la solicitud que la info esta lista
If(emisor acepta)
    EnvioDatosAAplicacionMobile(emisor);
}

```

Aplicación Mobile – Recepción de información.

Recibir informacion de embebido y envio

```

RecibirValidarEntorno{
    ...
    Video = RecibirDatosEmbebido();
    A = VerificarPesoDeVideo(video);
    B = VerificarCalidadDeVideo(video);
    C = VerificarTiempoDVideo(video);
    ...
    If (A==1 Y B==1 Y C==1)
        enviarImagenesValidadasASVsExterno(video);
        crear_hilo_pendiente_de_rta(respuesta) //cuando la señal de respuesta esté lista para ser
transmitida desde alguno de los servidores, el hilo encargado devolverá un booleano.
    else
        solicitarNuevaTomaDeVideoAEmbebido();
        evento_rta_1(respuesta); //Cuando el hilo devuelva un “true”, un evento se disparará e
iniciara la rutina de recepción y envío de información al embebido.
    ...
}

```

4 Pruebas que pueden realizarse

Para probar y verificar la funcionalidad del módulo basta con colocar el dispositivo regador en la ubicación donde desee que riegue, y activar el modulo mediante la

aplicación. Una vez hecho esto observaremos, cuando comience el riego, la moderación del flujo y tiempo de riego en determinado sector. Particularmente esto será observado claramente donde existan cultivos que requieran mayor flujo de agua (un árbol) o requieran menor flujo de agua (Un cactus).

5 Conclusiones

Hemos recorrido las modificaciones a realizar en el proyecto SmartGarden si se busca añadir una funcionalidad compatible con el reconocimiento de objetos. Se ha observado el alto costo de desarrollo de la misma además de los esfuerzos por obtener eficiencia y respuestas en tiempos estipulados.

Las lecciones que se dejan asentadas luego de este análisis se observan en el entendimiento de la complejidad que un sistema de inteligencia avanzado atrae, junto con las metodologías complejas utilizadas para ello. Se deja a entrever también, que existen campos para continuar avanzando sobre una base: Los algoritmos de reconocimiento no son del todo precisos y muchos de ellos ofrecen ventajas que otros no.

Luego de recorrer este camino, muchas ideas en torno a este tipo de inteligencia comienzan a surgir, por ejemplo, llevar este módulo de reconocimiento de objetos más allá y utilizar un reconocimiento facial para, por medio de gestos, la aplicación detecte los mismos y se realicen acciones que impacten en el *endpoint*.

6 Referencias

- Ebrahim Karami, S. P. (2017). Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. Memorial University, Canada: Faculty of Engineering and Applied Sciences.
- Güzel, M. S. (2015). A Hybrid Feature Extractor using Fast Hessian Detector and SIFT. Ankara: ISSN 2227-7080.
- Hima Bindu Ch, D. K. (2019). SIFT and It's Variants: An Overview. Amity University Rajasthan, Jaipur, India: SUSCOM-2019.
- Marc-Andre Hermanns, K. M. (2018). Introduction to the MPI_T Events Interface. Tools Working Group.
- Young, H. G. (s.f.). MIMD Interpretation on a GPU. Kentucky: University of Kentucky, Electrical and Computer Engineering.