Deteccion, procesamiento y reconocimiento de objetos

Amato Luciano Salvador, Borda Jonathan Marias, Lancuba Emiliano, Gomez Nicolas Eduardo, Olivera Florencia

[1]Universidad Nacional de La Matanza,
Departamento de Ingeniería e Investigaciones Tecnológicas,
Florencio Varela 1903 - San Justo, Argentina
luchiiamato@gmail.com, jonathanmatiasborda@gmail.com, lancuba.emiliano@gmail.com, nneegomez@gmail.com, o.florencia123@gmail.com

Resumen. En la actualidad, existen distintas metodologias y algoritmos conocidos y funcionales para la deteccion de objetos y reconocimiento de imágenes. Cabe destacar que mas alla del avance y evolucion que tuvo este campo con el avance de las tecnologias en los ultimos años, aun no existe un algoritmo que se adapte completamente a todos los campos requeridos (Costo, tiempo y alcance). Particularmente, su buscara demostrar teoricamente la implementacion del algoritmo SIFT de reconocimiento de imágenes con el objetivo de mejorar la precision y las prestaciones que "SmartGarden" puede ofrecer. Para la implementacion de esta estructura nos encontraremos con la necesidad de cubrir diversas aristas y realizar distintos cambios en la actual estructura del proyecto. Ademas de proponer y argumentar la eleccion del algoritmo SIFT, se buscara embeber el proyecto en un ecosistema IOT la cual beneficiara y brindara soporte al modulo.

Palabras claves: SIFT, reconocimiento de imágenes, arduino, mobile, android.

1 Introducción

La investigacion plasmada en este documento busca mostrar la forma elegida de implementar una tecnologia de reconocimiento de imágenes combinadas con el añadido de construir una estructura IOT, con la cual se buscara añadir una funcionalidad robusta al modulo, y, por consiguiente, al proyecto en general. Esto, ademas, abrira las puertas para incluir otras funcionalidades no especificadas en el presente. Nos adentraremos en la implementacion de una tecnologia asi como tambien en los cambios arquitectonicos que se deben realizar tanto en el endpoint como en su macroinfraestructura general del proyecto. Esto es, los cambios en la regadera (*endpoint*) y los cambios a realizar para establecer un contexto IOT (Multiples nodos conectados con el fin de aprender y obtener datos para la toma de decisiones). Queda fuera de enfoque de este documento el tratamiento del codigo e implementacion tecnica de los mecanismos necesarios para que el sistema de riego capte las imágenes de su entorno. Se abordaran las soluciones que utilizara la aplicaion para el

tratamiento de estos datos captados por el sistema de riego, para retroalimentar al mismo con los datos pertinentes luego del analisis de reconocimiento.
Actualmente se posee un sistema de contexto estanco, cuya principal ventaja con respecto a la competencia es la de brindar un sistema de autoaprendizaje, por medio del cual el sistema aprende de su entorno y mejora con el tiempo la precision y previsibilidad de los parametros de riego. Además, posee la tecnologia apropiada para establecer una comunicación por medio del protocolo Bluetooth donde, utilizando una aplicación como medio, establece conexiones con dispositivos inteligentes (los cuales posean una version de API de Android mayor o igual a 19 -Android 4.4-), permitiendo el intercambio de informacion entre estos.
Cabe destacar que el objetivo de la implementacion del modulo de reconocimiento de imágenes es otorgar una nueva funcionalidad orientada a profundizar y mejorar la inteligencia del modelo. Se lograran identificar objetos particulares, los cuales variaran tanto el flujo de agua de la regadera como el tiempo de riego sobre los objetos. Estas modificaciones en los flujos seran calculadas previo al riego establecido según su funcion de aprendizaje automatico,y ayudaran a elegir el mejor flujo de riego en cada momento del mismo. En palabras mas simples, este nuevo modulo permitiria detectar un objeto o conjunto de objetos (Cultivo, persona, animal, etc.) el cual necesite mas o menos flujo de agua por taza de tiempo, de manera tal que obtendriamos un control mas alto de las posibilidades de riego de nuestro dispositivo.
Existen aplicaciones hoy en dia implementadas que utilizan el reconocimiento de imágenes para obtener provecho: Algunos smartphones cuentan con esta tecnologia para detectar objetos o incluso rostros. Algunos ejemplos generales de uso de este mecanismo son:

- La aplicación de **Ebay** que permite buscar artículos usando la cámara.
- Una **red neuronal** que convierte fotos en tonos negros en imágenes brillantes.
- La inteligencia artificial de **Facebook**, que interpreta las fotografías que suben los usuarios para adecuar la publicidad dirigida a cada cuenta.

2 Desarrollo

Para montar este modulo se deben comenzar con los cambio a realizar en el arduino: Tanto logicos como de su infraestructura. El modelo añade una camara al regador (endpoint) la cual tomara datos previo al riego y mediante la conexion con la aplicacion enviara estos a la misma para su procesamiento.
La aplicación recibira las visuales requeridas y enviara esto a una granja de servidores externos a traves de internet. Estos servidores externos son los que, implementando los algoritmos de reconocimiento, obtendra los datos necesarios de flujo y tiempo de riego en los objetos, interactuando con una base de datos que se retroalimentara con cada nuevo reconocimiento realizado. En palabras simples: Una camara captura imagenes del entorno, las cuales se envian a la aplicacion mobile, estas seran

validadas (Tiempo de video, velocidad, etc) y se enviaran a traves de internet a un area de servidores externos de SmartGarden. Luego, estos, mediante algoritmos de paralelizacion de procesamiento (el cual se abordara en la siguiente seccion), y mediante algoritmos de reconocimiento y la ya mencionada base de datos, obtendra los parametros de riego necesarios para esa deteccion. Estos parametros seran enviados en camino inverso (A traves de internet a la aplicación y luego desde la aplicación se le dara la forma necesaria para que el sistema embebido pueda interpretar esos valores).
Como ya se ha mencionado en el paper, SIFT sera la tecnica utilizada para lograr realizar un conjunto de detecciones que nos ayuden a identificar una imagen como tal. SIFT es una tecnica publicada por David Lowe en 1999 y pese a sus 20 años de antiguedad es una de las mas utilizadas. Entre sus ventajas principales y la principal para los autores de este documento se encuentra la gran performance que sostiene ante efectos luminicos, una caracteristica muy necesaria para el contexto de nuestro producto. Por el contrario, el tiempo de procesamiento y consumo de bateria del mismo se disparan a niveles elevados. No se tiene por objetivo explicar el funcionamiento completo del algoritmo SIFT pero se debe realizar una aproximacion del mismo para comprender su funcionamiento. Este algoritmo se realiza en cuatro pasos:

- Detección de extremos en el espacio-escala: Mediante un algoritmo matemático, 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: El anterior proceso encuentra diversos candidatos de los cuales se deben eliminar aquellos que no sean estables a cambios de iluminación y ruido.
- Asignacion de orientacion: 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: Una vez realizado todos los procedimientos previos y teniendo un conjunto de puntos claves, el siguiente paso es determinar para cada punto clave un descriptor (ubicado en la base de datos) relativamente invariante a cambios de iluminación y afinidades, basado en el entorno del mismo.

Lo primero a destacar es que para agilizar este proceso utilizaremos una granja de servidores implementando OpenMP para reelizar procesamiento distribuido. Una vez aclarado esto, para realizar estas detecciones de puntos clave y descriptores en un gran volumen de datos, en los servidores utilizaremos el paradigma SIMD (Single Instruction Multiple Data), el cual ayudara a obtener una considerable reducción del tiempo de procesamiento sumado tambien a la implementacion de CUDA (Compute Unified Device Architecture) la cual nos permitira utilizar la GPU para realizar estos procedimientos matematicos a una velocidad aun mayor (Al ser un ecosistema IOT las solicitudes por taza de tiempo tendran un valor eleveado y se debera responder con la mayor agilidad posible). Si bien sabemos que SIFT requiere tiempo de procesamiento alto, utilizando la combinacion de SIMD + CUDA obtendremos un equilibrio necesario para obtener los tiempos de respuesta esperados.

3 Explicación del algoritmo.

El siguiente pseudocódigo ejemplificará el uso de CUDA con el algoritmo SIFT utilizando OpenMP para la distribución de carga entre servidores. No se abordarán las implementaciones tanto del arduino como de la aplicación Android para el movimiento y adaptación de los datos.

Servidores Externos – Pseudocodigo – Implementación de CUDA, SIFT, OPENMP

Recibir información de embebido y envió

```
#Incluir librerías de CUDA y OPENMP

video_rec = Recibo_video_de_entorno_de_mobile_a_traves_de_internet(emisor)
imagenes[] = cortarVideo(video_rec, 3) //3 es el número de segundos.
#pragma omp parallel default(shared) private(imágenes[],hilos)//Comienzo paralelizacion
SiftData datosSift;
InitSiftData(datosSift, 25000, true, true);// Se reserva memoria para funciones de SIFT
   WHILE (i< imagenes[].length){
      Leer_Imagen(img)
      CudaImage img; //Crear_imagen_paralelizada
      img.Allocate(1280, 960, 1280, false, NULL, (float*) limg.data);
      img.Download();

      Defnicion_de_parámetros_de_SIFT

      Get_orientation(x);
      Create_dog(y);
      …//mas informacion para construir descriptor
      …
      V = Get_descriptor(SQL_SERVER_USER, SQL_SERVER_KEY, x,y,…);
      Match_database();
      i++;//sigue con la siguiente imagen
   END_WHILE
!$OMP END PARALLEL DO//Termino paralelizacion

EnvioDatosAAplicacionMobile(emisor);

}
```

Aplicación Mobile – Pseudocodigo – Envio y recepción de informacion

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)
    enviarImagenesValidadasASVExterno(video);
  else
    solicitarNuevaTomaDeVideoAEmbebido();
  …
  …
}
```

Recibir informacion de servidores externos

```
EnviarReconocimiento  {
  …
  …
  Ext  = recibirInfoExterno();
  Parámetros[] = establecerParametrosDeRiego(Ext.objetosReconocidos)
  enviarDatosEmbebido(parámetros[])
  …
  …
}
```

Sistema Embebido – Pseudocodigo – Envió y recepción de datos de entorno

Capturar información y enviar a aplicación

```
Void  loop() {
  …
  …
  EncenderCamara();
  GrabarEntorno();
  If(Bluetooth_disponible)
    enviarEntornoAAplicacion();
  …
  …
}
```

Recibir información y regular riego

```
Void  loop() {
  …
  …
  recibirInfoReconocimientoObjetos();
  establecerRiego();
  …
  …
}
```

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 notado donde haya objetos que se entienda pueden requerir mayor flujo de agua (un árbol de roble) o requieran menor o incluso nada de agua (Un cactus o una persona que pudo colocarse en el camino del riego).

5 Conclusiones

Hemos recorrido los cambios a realizar en el proyecto SmartGarden si se busca añadir una funcionalidad compatible con el reconocimiento de objetos. Se ha observado el costo de la misma lo cual no solo influye en tiempo de desarrollo sino también comienzan a asomar variables importantes que antes no debían ser observadas con detenimiento: Performance, consumo de batería, conexiones externas, tiempo de respuesta, complejidad de implementación, etc.

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. Es importante añadir, si bien el núcleo de este módulo consta de un desarrollo de reconocimiento de objetos, existe toda una infraestructura detrás y cambios en el producto original.

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.

Flores, P. (2011). *Algoritmo SIFT: fundamento teorico*. Montevideo: Instituto de Ingenieria Electrica.

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.

Jyothi Krishna V S, S. B. (2018). *Compiler Enhanced Scheduling for OpenMP for*. Indian Institute of Technology Madras, Chennai, India.