Plan de Pruebas

Para

Segmentación Semi-Automática de Tejido Adiposo en el Espacio Parafaríngeo por Medio de Aprendizaje Profundo

Por

David Alejandro Castillo Chíquiza
Oscar David Falla Pulido
Juan Sebastián Ruiz Bulla

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BOGOTÁ, D.C.
2022

Tabla de Contenidos

Lista de tablas

Capítulo 1

Introducción

1.1. Objetivo General

Definir y estructurar la fase de pruebas que se desarrollará para garantizar la completitud funcional y la alta calidad en el sistema propuesto.

1.2. Estrategia de Pruebas

Reunir y comprender los recursos disponibles para desarrollar el sistema propuesto con el fin de optimizarlos, asegurando la calidad y funcionalidad del sistema. Estos recursos comprenden: el hardware, el software, las características de las imágenes y la arquitectura del modelo de redes neuronales convolucionales. Teniendo en cuenta estos recursos, se debe encontrar la configuración óptima que permite tener resultados de calidad y un funcionamiento completo del sistema.

1.3. Alcance

Contar con un sistema capaz de segmentar el tejido adiposo en la región parafaringea dada una imagen por resonancia magnética, utilizando un modelo de redes neuronales convolucionales pre-entrenado. Asimismo, a través del sistema, el entrenamiento de un nuevo modelo para segmentar la región de interés que el usuario desee en imágenes por resonancia magnética.

Capítulo 2

Entregables

2.1. Documentación a Entregar

Al final del plan de pruebas se deben entregar los siguientes documentos para demostrar el éxito del plan y la completitud del sistema, junto con la calidad de sus funcionalidades:

Casos de pruebas

En la tabla 2.1 se expone una plantilla a seguir para documentar los casos de prueba de las características del sistema.

	Intento n
No. Imágenes	Número de imágenes para entrenamiento y prueba
Dimensiones	Dimensiones de las imágenes
Canales	Canal de las imágenes
Pesos para clases	Pesos dados para las clases en entrenamiento
Arch Unet	Descripción de la arquitectura Unet utilizada
Optimizador	Optimizador dado para el entrenamiento
Tasa de aprendizaje	Tasa de aprendizaje dada para el entrenamiento
Épocas	Épocas dadas para el entrenamiento
Dice	Índice de Dice obtenido
Accuracy	Precisión obtenida
Pérdida	Pérdida obtenida
Tiempo	Tiempo gastado por el entrenamiento
GPU	Características de la GPU utilizada
RAM	Características de la RAM utilizada
Recurso	Características del recurso utilizado

Tabla 2.1: Plantilla para casos de prueba

Resultados de las pruebas

Conjunto de gráficas generadas por una librería de Python (matplotlib) donde se exponen las métricas obtenidas por el modelo, así como las imágenes obtenidas por esta librería de la predicción hecha por el modelo junto con la imagen original y la segmentación original de la imagen.

Capítulo 3

Características a ser Probadas

Para garantizar la calidad y el funcionamiento del sistema se deben probar los recursos en los cuales este será desarrollado. Estos recursos comprenden: el hardware, el software y los requirimientos que debe cumplir. En ese sentido, en la siguiente lista se explican las características se va a probar:

- **Recursos computacionales:** Se prueban características como la RAM, la tarjeta gráfica y la VRAM de los computadores, o recursos en la nube como Colab, en donde va a funcionar el algoritmo teniendo en cuenta sus mínimos para que el sistema funcione. Esto quiere decir: RAM mínima de 24GB, tarjeta gráfica de fabricante NVIDIA (compatibles con CUDA) y VRAM mínima de 16GB.

  Estas características se prueban usando el monitor del sistema operativo, o de Colab, para dar seguimiento a su consumo.

- **Configuración de recursos:** Se prueban características como: el sistema operativo donde se desarrollará el sistema, configuración de los drivers para la tarjeta gráfica, configuración de CUDA Toolkit, cuDNN SDK y tensorRT. En el caso de Colab solo la configuración de ejecución del entorno es necesaria probarla.

- **Carga de datos:** Se prueba que el sistema pueda cargar:

  - El modelo pre-entrenado o el modelo dado por el usuario para segmentar un conjunto de imágenes por resonancia magnética.
  - Las imágenes por resonancia magnética ya sea para segmentar estas imágenes o para entrenar un nuevo modelo.

- **Preprocesamiento de imagénes:** Se prueba que el sistema pueda preprocesar el conjunto de imágenes por resonancia magnética dado por el usuario para que las funcionalidades del sistema (segmentar y entrenar) puedan ser satisfactorias.
- **Cantidad de imagénes:** Se prueba la cantidad del conjunto de imágenes por resonancia magnética dado por el usuario ingresado en el sistema ya sea para entrenar un nuevo modelo o para segmentar este conjunto.
- **Modelo:** Se prueban características como los hiperparámetros de la red neuronal convolucional así como su arquitectura. Por ejemplo: cuántas capas va a tener la red, la profundidad, las neuronas en cada capa, la función de activación en esas neuronas, las épocas que va a estar entrenando, la tasa de aprendizaje, etc.
- **Entrenamiento:** Se prueba que el sistema pueda entrenar un nuevo modelo para la región de interés en un conjunto de imágenes por resonancia magnética dado por el usuario sin ningún tipo de fallo.
- **Segmentación:** Se prueba que el sistema pueda segmentar la región de interés en un conjunto de imágenes por resonancia magnética dado por el usuario con un índice de DICE de al menos el 85 %.

  El índice de DICE, o también conocido como índice de superposición, es la métrica más utilizada para validar segmentaciones médicas en 3D. Esta métrica refleja el acuerdo tanto en tamaño como en localización y se basa en la comparación de la presición de los pixeles entre la segmentación obtenida y la validación. En la ecuación 3.1 se observa que el índice de DICE se define como dos veces el área de superposición, dividido por el número de pixeles total de ambas segmentaciones.

$$DICE = \frac{2|S_g \cap S_t|}{|S_g| + |S_t|} \tag{3.1}$$

- **Tiempo de segmentación:** Se prueba el tiempo que toma la segmentación hecha por el sistema en un conjunto de imágenes por resonancia magnética dado por el usuario, teniendo en cuenta que este tiempo debe ser menor a la segmentación manual que hace un experto médico.
- **Persistencia de datos:** Se prueba que el sistema pueda guardar:
  - El modelo entrenado por el usuario dado un conjunto de imágenes por resonancia magnética en un archivo nuevo.

- Las segmentaciones hechas a un conjunto de imágenes por resonancia magnética en un archivo nuevo sin modificar los archivos originales.

Capítulo 4

Criterios

En este capítulo se exponen los criterios bajo los cuales se ejecutará el plan de pruebas, estos comprenden: Los criterios de aprobación, fallo, suspensión y reanudación.

4.1. Criterios de Aprobación y Fallo

Para poder dar por terminado el plan de pruebas se requiere principalmente que el sistema propuesto tenga un desempeño de alta calidad y su funcionamiento sea completo. Teniendo en cuenta la lista expuesta en el capítulo 3 se encuentra que:

- **Recursos computacionales:**
  - **Criterio de aprobación:** Optimización del consumo de los recursos computacionales teniendo en cuenta sus mínimos para que el sistema funcione.
  - **Criterio de fallo:** Cambio o mejora de recursos computacionales. Por ejemplo: mayor capacidad de RAM y VRAM.
- **Configuración de recursos:**
  - **Criterio de aprobación:** Entrenamiento exitoso del modelo en la funcionalidad de entrenamiento con GPU.
  - **Criterio de fallo:** Cambio de configuración en los recursos. Por ejemplo: versiones de CUDA, cuDNN o tensorRT.

- **Carga de datos:**
  - **Criterio de aprobación:** Mensaje del sistema expresando que la carga de datos (modelo y/o imágenes) ha sido exitosa.
  - **Criterio de fallo:** Modificación en el código fuente para resolver los problemas de la carga de datos (modelo y/o imágenes).
- **Preprocesamiento de imagénes:**
  - **Criterio de aprobación:** Mensaje del sistema expresando que el preprocesamiento de imagénes ha sido exitoso.
  - **Criterio de fallo:** Modificación en el código fuente para resolver los problemas del preprocesamiento de imagénes.
- **Cantidad de imagénes:**
  - **Criterio de aprobación:** Funcionalidad completa del sistema (entrenamiento y segmentación) con éxito.
  - **Criterio de fallo:** Cambio en las imágenes. Por ejemplo: mayor o menor cantidad de imágenes.
- **Modelo:**
  - **Criterio de aprobación:** Compilación exitosa de la arquitectura.
  - **Criterio de fallo:** Modificación en el código fuente para resolver los problemas de la arquitectura de la red neuronal.
- **Entrenamiento:**
  - **Criterio de aprobación:** Entrenamiento completo del modelo utilizando el conjunto imágenes por resonancia magnética dado por el usuario teniendo un índice de DICE de al menos 85 %.
  - **Criterio de fallo:** Modificación en el código fuente para resolver los problemas del entrenamiento del modelo.
- **Segmentación:**
  - **Criterio de aprobación:** Nivel de segmentación, por parte del algoritmo, donde se demuestre que la región de interés ha sido segmentada con éxito –este criterio es apreciativo y depende de la experticia del experto médico u observador–.

  - **Criterio de fallo:** Re-entrenamiento del modelo ajustando hiperparámetros, arquitectura de la red y/o cantidad de imágenes teniendo en cuenta la calidad de las mismas.

- **Tiempo de segmentación:**

  - **Criterio de aprobación:** El tiempo de segmentación debe ser menor a la segmentación manual que hace un experto médico.
  - **Criterio de fallo:** Re-entrenamiento del modelo ajustando hiperparámetros, arquitectura de la red y/o cantidad de imágenes teniendo en cuenta la calidad de las mismas.

- **Persistencia de datos:**

  - **Criterio de aprobación:** Mensaje del sistema expresando que la persistencia de datos ha sido exitosa.
  - **Criterio de fallo:** Modificación en el código fuente para resolver los problemas de la persistencia de datos.

4.2. Criterios de Suspensión y Reanudación

El plan de pruebas se suspendería en dado caso que no haya acceso a los recursos mínimos de hardware que el sistema necesita para ser probado, no hayan imágenes nuevas o mejoras en las imágenes actuales para entrenar el algoritmo nuevamente, o en caso de que no hayan mejoras en el desempeño del modelo según sus métricas.

De ese modo, el plan de pruebas se puede volver a reanudar principalmente cuando hayan imágenes nuevas o mejoras en las imágenes actuales para conseguir mejores métricas y mayor calidad en la segmentación.

Capítulo 5

Tareas de las Pruebas

Para poder completar el plan de pruebas de forma satisfactoria se necesitan documentar las tareas que se van a seguir; exponiendo quién va a hacer cuál tarea, la descripción de la tarea, sus fechas de inicio y final y la duración que toma. En la tabla 5.1 se presenta una organización de las tareas a realizar para obtener al final un sistema de calidad con completa funcionalidad.

Nombre	Descripción	Fecha inicio	Fecha final	Duración total	Responsable
Configurar computadores	Configuración de las herramientas y recursos necesarios para desarrollar el sistema.	22-08-2022	02-10-2022	1 mes y 1 semana	David Castillo
Carga de datos	Pruebas de la carga de datos para las funcionalidades del sistema propuesto.	06-10-2022	15-10-2022	1 semana	Todos
Preprocesar imágenes	Pruebas al preprocesamiento del conjunto de las imágenes dadas por el usuario ya sea para segmentar o entrenar.	16-10-2022	17-10-2022	1 día	Oscar Falla
Probar modelo	Pruebas de la arquitectura de red neuronal convolucional propuesta (número de capas, número de neuronas, funciones de activación, etc).	17-10-2022	19-10-2022	2 días	Sebastián Ruiz
Entrenar modelo	Pruebas al entrenamiento del modelo implementado ajustando sus hiperparámetros (número de épocas, tasa de aprendizaje, optimizador).	19-10-2022	28-10-2022	1 semana y 2 días	Todos
Entrenar modelo BraTS	Entrenamiento del modelo implementado con el dataset de BraTS 2020.	29-10-2022	31-10-2022	2 días	Oscar Falla
Persistencia de Datos	Pruebas de la persistencia del modelo en la funcionalidad de entrenamiento, así como las segmentaciones hechas en la funcionalidad de segmentación.	01-11-2022	01-11-2022	Menos de 1 día	Sebastián Ruiz
Probar CLI	Pruebas de las funcionalidades de la CLI para el sistema propuesto.	02-11-2022	05-11-2022	3 días	David Castillo

Tabla 5.1: Organización de tareas a realizar

Capítulo 6

Necesidades Ambientales

6.1. Hardware

Computadores o recursos en la nube como Colab que cuenten con RAM mínima de 24GB y VRAM mínima de 16GB y tarjeta gráfica de fabricante NVIDIA (compatibles con CUDA). Esto con el objetivo de poder entrenar el modelo sin ningún problema y optimizando el tiempo.

6.2. SUT (Sistema Bajo Pruebas)

El funcionamiento del sistema completo. Esto es: la funcionalidad de segmentar el tejido adiposo en la región parafaríngea dada una imagen por resonancia magnética utilizando un modelo pre-entrenado y la funcionalidad de entrenar un nuevo modelo para segmentar una nueva región de interés en un conjunto de imágenes por resonancia magnética dado por el usuario.

6.3. Test-Ware

La configuración de los computadores o recursos en la nube como Colab. Esta configuración comprende: instalación del sistema operativo Linux distribución basada en Debian, configuración de los drivers para la tarjeta gráfica, instalación y configuración de CUDA

Toolkit, cuDNN SDK y tensorRT para que los modelos implementados se ejecutaran con GPU, instalación de gestor de paquetes de Python (pip), instalación de librerias para trabajar con aprendizaje profundo (tensorflow, keras, sklearn, etc). En el caso de Colab solo la configuración de ejecución del entorno es necesaria. Esto es: utilizar GPU y RAM amplia para el entorno de ejecución.

Capítulo 7

Riesgos

Un buen funcionamiento del sistema requiere cumplir los requisitos mínimos expuestos en el capítulo 6 y satisfacer las características a ser probadas del sistema en el capítulo 3. El incumplimiento de estos puede generar riesgos y un mal desempeño del sistema. En la tabla 7.1 se presentan algunos riesgos que puede tener el sistema, las consecuencias, un nivel de su impacto y su acción preventiva:

Nombre	Descripción	Consecuencia	Impacto	Acciones preventivas
Recursos computacionales	No tener los recursos necesarios (capacidad RAM, capacidad VRAM, etc).	La funcionalidad de entrenamiento no podría ejecutarse. No existirían recursos para el entrenamiento del modelo.	Muy alto	Obtener recursos que satisfagan las necesidades mínimas, ya sea en la nube como Colab o uso de otros computadores.
Configuración recursos	La no configuración de los Drivers, al igual que CUDA Toolkit, cuDNN SDK y tensorRT.	Mayores tiempos de entrenamiento (se entrena con CPU) o fallos de entrenamiento por falta de memoria.	Alto	Configuración exitosa de los recursos para soportar el entrenamiento del modelo.
Tipo de imágenes	Uso de imágenes médicas diferentes a imágenes por resonancia magnética.	El sistema no funcionaría dado que sus funcionalidades son exclusivas para tipo de imágenes por resonancia magnética.	Muy alto	Restringir el uso de imágenes médicas a imágenes por resonancia magnética.
Carga de datos	Fallo o no implementación del código para cargar datos.	El sistema no podría cargar el modelo pre-entrenado o dado por el usuario y el conjunto de imágenes por resonancia magnética dado por el usuario.	Muy alto	Implementación del código fuente exitosa para la carga de datos.
Preprocesamiento de imágenes	Fallo o no implementación del código para preprocesar las imágenes.	El desempeño de las funcionalidades del sistema sería bajo dado que el resultado de estas (segmentaciones o modelo) no tendría buena calidad.	Alto	Implementación del código fuente exitosa para el preprocesamiento de imágenes.
Cantidad de imágenes	Cantidad insuficiente de imágenes.	El desempeño de la funcionalidad de entrenamiento sería bajo dado que no se cuenta con la cantidad de imágenes suficientes para resultar con un modelo de calidad.	Medio	Contar con una cantidad de imágenes suficientes para el entrenamiento de un nuevo modelo.
Calidad de segmentaciones	Baja calidad del conjunto de datos utilizado en entrenamiento.	El modelo resultante del entrenamiento tendría resultados insatisfactorios en el índice de DICE.	Alto	Obtener imágenes de buena calidad o probar otros conjuntos de datos de mayor calidad.
Persistencia de datos	Fallo o no implementación del código para cargar datos.	El sistema no podría guardar el modelo entrenado por el usuario y el conjunto de segmentaciones de las imágenes por resonancia magnética.	Muy alto	Implementación del código fuente exitosa para la persistencia de datos.

Tabla 7.1: Riesgos