

Böngészőalapú szoftverrendszerek fejlesztése orvosi képfeldolgozó és radiológiai leletezési felhasználásra

PhD tézis

Dr. Harmouche Ahmed Mouhamad

Klinikai Idegtudományok Doktori Iskola (D221)

Idegtudományi képzés (B-3/2014)

A doktori iskola vezetője: Prof. Dr. Komoly Sámuel

Programvezető: Prof. Dr. Bogner Péter

Témavezető: Dr. Tóth Arnold

Pécsi Tudományegyetem Általános Orvostudományi Kar

Orvosi Képző Klinika



Pécs, 2023

Rövidítések jegyzéke

AI – Artificial Intelligence

BET – Brain Extraction Tool

CSS – Cascading Style Sheets

CT – Computerized Tomography

DICOM – Digital Imaging and Communications in Medicine

DOM – Document Object Model

FLIRT – FMRIB's Linear Image Registration Tool

FSL – FMRIB Software Library

GUI – Graphical User Interface

HIS – Hospital Information System

HTML – HyperText Markup Language

HTML5 – HyperText Markup Language 5

LLM – Large Language Model

LLVM – Low Level Virtual Machine

MGH – Massachusetts General Hospital

MINC – Medical Imaging NetCDF

MIT – Massachusetts Institute of Technology

MRI – Magnetic Resonance Imaging

NifTI – Neuroimaging Informatics Technology Initiative

RIS – Radiology Information System

RTG – Röntgen

SPA – Single Page Application

TPU – Tensor Processing Unit

URL – Uniform Resource Locator

WebGL – Web Graphics Library

XSS – Cross Site Scripting

Bevezetés

A 21. század digitális fejlesztései modernizálták az orvoslást, és jelentős technológiai előrelépéseket tettek lehetővé. A század két talán legnagyobb vívmánya a web és a mesterséges intelligencia (Artificial Intelligence, AI), melyek közül mindkettő hamar felforgatta az orvoslást. Ezen belül is kiemelkedik a radiológia, hisz ez egy olyan terület, ahol minden olyan technológia hamar hasznosulni tud, ami az orvosi képek (RTG, CT, MRI) kiértékelését segíti.

A kórházi informatikai rendszerek (Hospital Information System, HIS), és a radiológiai informatikai rendszerek (Radiology Information System, RIS) javarészt webes alapúak, még ha nem is feltétlenül az összes komponensük. Míg kezdetben a radiológiai munkafolyamatoknak azon részén jelentek meg a böngészőalapú megoldások, amik nem igényelnek nagy számítási kapacitást (főként adminisztratív jellegű tevékenységek), úgy később a web fejlődésével, nevezetesen a HyperText Markup Language 5 (HTML5), a Web Graphics Library (WebGL) és a WebAssembly megjelenésével már nagy számításigényű képfeldolgozó és képelemző algoritmusok böngészőben való futtatása is lehetővé vált. Mivel a webböngészők szinte minden digitális platformon elérhetőek (asztali számítógépek, okostelefonok, tabletek), a webes programok platformfüggetlenek. A platformfüggetlenség mellett a böngészős alkalmazások telepítést és karbantartást sem igényelnek, hisz a program futásához szükséges minden függőség és előfeltétel a weboldalra látogatáskor letöltésre kerül a böngészőbe. Az alkalmazások manuális frissítésére sincs szükség ily módon, hisz a program új verziója azonnal elérhető lesz az oldal megnyitásakor. A webes alkalmazások ezen tulajdonságai nagyban hozzájárultak az orvosi informatikai térhódításuknak. A korábban említett radiológiai felhasználási területek mellett a leletezésre, ezen belül is a strukturált, sablon alapú leletezésre is készültek webes megoldások.

A továbbiakban két fontos radiológiai munkafolyamat, a képfeldolgozás és a leletezés informatikai, ezen belül is webes vonatkozását szeretnénk bemutatni.

Az orvosi képfeldolgozás informatikai alapjai

Az orvosi képfeldolgozás magában foglal minden olyan technikát, melyek az orvosi képek javítására, elemzésére és feldolgozására irányulnak. Az orvosi képalkotó eszközök és a RIS közötti kommunikáció a Digital Imaging and Communications In Medicine (DICOM) szabvány szerint történik. A DICOM egyszerre definiál egy kommunikációs protokolt, és egy fájlformátumot is. A különböző modalitások által generált felvételek rendszerint DICOM formátumban kerülnek (általában „.dcm” kiterjesztéssel) mentésre a készülékek adatbázisában, és ebben a formátumban kerülnek továbbadásra is egy következő DICOM csomópontnak. Azok a modalitások, amelyek térben összefüggő felvételeket készítenek (CT, MRI), úgynevezett szeletekben generálják, mentik le, és küldik tovább a képeket. Ennek az az előnye, hogy a felvétel kiértékelése megkezdődhet anélkül, hogy a teljes térfogat betöltésre kerülne, hisz a szeletek külön-külön címezhetőek és letölthetőek. Hátránya azonban, hogy a felvételek elemzése és feldolgozása körülményesebbé válik. Volumetrikus felvételek képfeldolgozása szempontjából előnyösebb egy olyan fájlformátum, ami egy fájlban írja le az egész térfogatot, és a felvételt, mint háromdimenziós adathalmazt kezeli.

Idegtudományi képfeldolgozás

Az idegtudományi képfeldolgozás az orvosi képfeldolgozásnak azon halmaza, amely az agy és a központi idegrendszer struktúráinak és működésének vizualizálására és elemzésére fókuszál. Az MRI különösen fontos ezen a területen, hisz az agyról nagy felbontású képeket tud készíteni anélkül, hogy a betegeket ionizáló sugárzásnak tenné ki. Míg a DICOM egy az orvosi képalkotás szinte minden területén gold standard fájlformátumnak és szabványnak mondható, a Neuroimaging Informatics Technology Initiative (NIfTI) formátum az Analyze utódjaként kifejezetten idegtudományi felhasználásra lett kifejlesztve. A cél egy olyan formátum létrehozása volt, ami megkönnyíti a neurológiai volumetrikus adatok elemzését és feldolgozását, áthidalva a DICOM limitációit képelemzés szempontjából. Az idegtudományi képfeldolgozás két fontos algoritmus a agy kiemelés (brain extraction) és a lineáris regisztráció, melyeket legtöbbször az FMRIB Software Library (FSL) szoftvercsomag programjaival hajtanak végre. Az FSL egy nyíltforráskódú szoftvercsomag, melynek szoftverei a neurológiai képelemzés teljes spektrumát lefedik. Lineáris regisztrációra a FMRIB's Linear Image Registration Tool (FLIRT), agy kiemelésre a Brain Extraction Tool

(BET) programok használhatóak. Bár Windows, Linux és MacOS platformokra is elérhető a szoftvercsomag, leginkább Linux környezetben használják. Mivel sem a telepítése, sem a használata nem triviális legalább alapfokú programozói ismeretek nélkül, klinikai használata nem jellemző, azonban a kutatásban előszeretettel használják.

A nyíltforráskódú idegtudományi képfeldolgozó alkalmazások közül kiemelendő még a Slicer3D. Míg az FSL főként parancssori, úgy a Slicer3D inkább grafikus felhasználói felületen (Graphical User Interface, GUI) keresztüli használatra fókuszál, így programozói vagy mélyebb informatikai jártasság nélkül is könnyen használható. A fentebb említett lineáris regisztrációt és agy kiemelést a Slicer3D is támogatja, azonban nem beépített, hanem harmadik féltől származó bővítményeken keresztül.

Webes szoftverek az idegtudományi képfeldolgozásban

Ahogy azt korábban említettük, a HTML5, a WebGL és a WebAssembly technológiáknak köszönhetően nagy számítási igényű algoritmusok böngészőben való futtatása is lehetséges.

A HTML5 a HyperText Markup Language (HTML) leírónyelv legújabb verziója, mely a következőkben részletezett innovatív technológiákat hozta be a böngészők világába.

A WebGL grafikus könyvtár lehetővé teszi, hogy a böngésző két- vagy háromdimenziós képek megjelenítéséhez a számítógép grafikus feldolgozó egységének (Graphics Processing Unit, GPU) hardveres gyorsítását használja, ezáltal szignifikánsan lecsökkentve egy adott kép renderelésének idejét a központi feldolgozó egység (Central Processing Unit, CPU) szoftveres rajzolási idejéhez képest. Ez annak köszönhető, hogy míg a CPU egyszerre csak egy képpont kirajzolását tudja végrehajtani, úgy a GPU ezt párhuzamosítva végzi, egy időben akár több millió képpont feldolgozását is lehetővé téve.

A WebAssembly egy bináris formátum, mely olyan gépi utasításokat tartalmaz, amelyet egy a böngészőben futó virtuális gép képes futtatni. Szemben a böngésző „natív” programnyelvével, a JavaScripttel, mely dinamikusan típusos, a WebAssembly statikusan típusos, így a program típushelyessége még fordításidőben leellenőrzésre kerül, ezzel teret adva számos optimalizálási lehetőségnek, melyek gyorsabb futási időt eredményeznek.

A neurológiai adatok mérete gigabyte-os nagyságrendben mozog, így azok feldolgozása és megjelenítése a nagy számítási igényű kategóriába esik.

A BrainBrowser egy böngészőalapú szoftver, mely a fentebb ismertetett technológiákat használja volumetrikus neurológiai adatok kétdimenziós szeletenkénti (slice-by-slice), és

háromdimenziós felületi megjelenítésre. A szoftver nyíltforráskódú, és mind a moduláris architektúrája, mind pedig a licenszelése (GNU Affero General Public License v3.0) lehetővé teszi a módosítást, és bővítést.

Radiológiai leletezés

A radiológiai leletezés elengedhetetlen az orvosi képzési folyamatban. A radiológiai leleten keresztül ad választ a radiológus a beutaló klinikus kérdésére, tehát egy fontos közvetítő szerepe van a klinikus és a radiológus közötti kommunikációban. A leletben közölt információnak pontosnak, tömörnek és világosnak kell lennie. Fontos szempont továbbá, hogy a radiológus mennyi idő alatt tudja elkészíteni a leletet, vagyis, hogy a klinikus milyen gyorsan kap választ a kérdésére, és a beteg milyen gyorsan kapja meg a megfelelő diagnózist, és végül a kezelést.

Két fő irány létezik a radiológiai leletezésben:

- **Diktálás alapú leletezés:** Ezzel a módszerrel a radiológus egy mikrofon segítségével szóban ismerteti a képen látott elváltozásokat. A hanghullámokat egy elemző szoftver valós időben szabadszöveggé konvertálja, és beírja egy szövegdobozba, amit aztán az orvos módosíthat, formázhat. Bár ez a módszer hatékony és rugalmas, jelentős variabilitást eredményezhet a leletek tartalmában és formátumában a különböző szóhasználat, megfogalmazás és leletezési stílus miatt.

- **Sablon alapú strukturált leletezés:** Ezzel a módszerrel a radiológusoknak egy előre meghatározott sablont kell kitölteniük, hasonlóan egy Google Forms kérdőív kitöltéséhez. A strukturált leletek növelik a konzisztenciát a rugalmasság rovására.

Webes szoftverek a strukturált leletezésben

Számos szoftveres megoldás áll rendelkezésre strukturált leletezésére, melyek közül talán a legismertebb a Radiological Society of North America (RSNA) által kifejlesztett RadReport nevű webes platform, mely lényegében egy nagy sablon adatbázis, ahová felhasználók tudnak strukturált leletezési template-eket beküldeni, melyeket az oldalon tudnak elkészíteni a T-Rex Template nevű webes alkalmazással. A sablonok modalitás, dátum és számos egyéb paraméter szerint szűrhetőek. Bár a program (a sablon böngésző és a sablonkészítő is) ingyenesen használható, harmadik fél általi módosítása nem lehetséges, hisz a szoftver nem nyílt forráskódú.

Több cég is fejlesztett strukturált leletező megoldásokat (Smart Reporting: <https://www.smart-reporting.com>, RadioReport: <https://radioreport.com/>), azonban ezek nem érhetőek el ingyen, és még a kipróbálásukat is kérvényezni kell egy űrlapon keresztül, majd csak a kérvény jóváhagyása után lehet tesztüzemben kipróbálni az alkalmazásokat.

Mesterséges neurális hálózatok a radiológiában

Ahogy a bevezető első szakaszában említettük, a web mellett a másik nagy technológiai áttörést a mesterséges intelligencia, azon belül is a mesterséges neurális hálók hozták a radiológiában. A GPU-k és a tenzor feldolgozó egységek (Tensor Processing Unit, TPU) terén elért rohamos hardveres és szoftveres fejlődés lehetővé teszi az akár több tízmilliárdos paraméter számú neurális hálók tanítását és futtatását. Ez azt jelenti, hogy a mostani AI modellek rendkívül komplex összefüggéseket képesek megtanulni, legyen szó kétdimenziós, háromdimenziós adathalmazról, vagy akár szabadszövegről.

- **Konvolúciós neurális hálózatok:** A konvolúciós neurális hálózatok (Convolutional Neural Network, CNN) olyan modellek, melyek által végrehajtott fő művelet a konvolúció. A konvolúció a bemeneti adatra történő különféle szűrők (filterek) alkalmazásaként értelmezhető. A szűrők használatával a bemeneti adat leegyszerűsíthető: a modell képes az élek, sarkok ily módon történő kijelölésére, majd a szűrt képet tovább adva a háló későbbi rétegei formákat, objektumokat tudnak felismerni. A CNN-eket gyakran alkalmazzák különféle szegmentálási (pl. CT felvételen szervek teljes kijelölése) és objektum felismerési

feladatokra (pl. RTG felvételen pneumónia detektálása) az orvosi képzésben és képfeldolgozásban.

- **Nagy nyelvi modellek:** A nagy nyelvi modellek (Large Language Model, LLM) olyan nagy (akár több százmilliárdos) paraméterszámú neurális hálózatok, melyek nyelvi feladatok (mint például fordítás, szöveg generálás, beszélgetés, információ kiszűrés, hosszabb szöveg összefoglalása) elvégzésére nagy hatékonysággal betaníthatóak. Jellemző rájuk, hogy úgynevezett „few-shot learner” képességük van, ami azt jelenti, hogy egy adott feladattípusból elegendő néhány példát látniuk, hogy általánosítani tudjanak, és megtanulják azt. A hatalmas paraméterszám miatt az LLM-ek betanítása feletébb nagy energiaigényű, és költséges. A jelenleg igen népszerű, az OpenAI cég által fejlesztett GPT4 LLM betanítása közel 100 millió dollárba került. A hatalmas költségek miatt általában az LLM-eket egyszer tanítják be, és úgynevezett „prompt” -ok (súgás) által a felhasználók tudják finomhangolni a hálót különféle nyelvi feladatok elvégzésére. Elérhetőek nyílt forráskódú LLM-ek is, mint például a Meta által fejlesztett LLaMA, egy 65 milliárd paraméterrel rendelkező LLM.

A nagy nyelvi modellek fentebb ismertetett tulajdonságai a radiológiai leletezésben is hasznosíthatóak lehetnek.

Célok

Ezen dolgozat fő célja a web-alapú orvosi szoftverrendszerek alkalmazhatóságának vizsgálata orvosi képfeldolgozó és radiológiai leletezési területen. Célunk egy teljes egészében böngészőben futó neurológiai képelemző szoftver, a WebMRI létrehozása, amely a nyílt forráskódú BrainBrowser-re épül. A BrainBrowser térfogati megjelenítési képességei és az FSL képfeldolgozási algoritmusainak ötvözésén keresztül egy olyan platformot hozunk létre, amely képes agy kivonást és lineáris regisztrációt futtatni dedikált web szerver vagy bővítmények nélkül, melyet az FSL BET és FLIRT eszközöknek a WebAssembly-re történő portolása tesz lehetővé. A portolt szoftvereket összehasonlítjuk a natív FSL programokkal.

Egy másik web-alapú orvosi szoftvert is elkészítünk, az XReport-ot, amely egy nyíltforráskódú, ingyenesen használható strukturált leletezési platform. Szoftverünket összevetjük egy másik ingyenesen elérhető platformmal, az RSNA RadReport-jával.

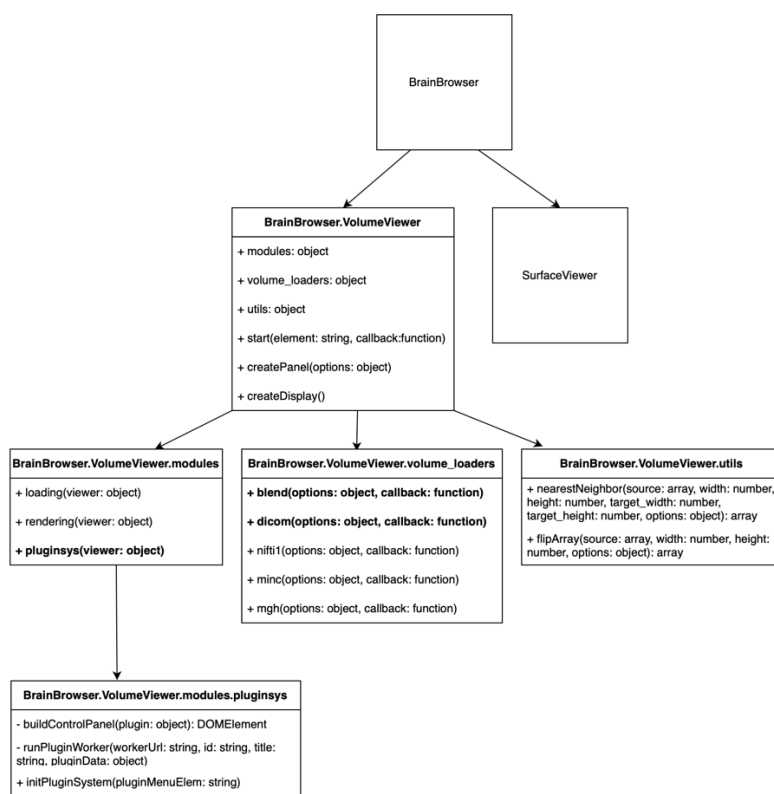
Webes orvosi szoftvereinkkel szeretnénk előmozdítani a böngésző alapú idegtudományi képfeldolgozást és a strukturált leletezést, valamint népszerűsíteni a nyílt forráskódú orvosi szoftvereket. Mindkét programunk forráskódját elérhetővé tesszük a Github-on.

Anyagok és módszerek

WebMRI

A WebMRI fejlesztése három fő részre bontható:

- Az FSL BET és a FLIRT WebAssembly-re portolása.
- A BrainBrowser módosítása, és egy plugin rendszerrel történő kibővítése, hogy a portolt eszközöket a BrainBrowseren belül tudjuk futtatni (1. ábra).
- Egy demo webalkalmazás fejlesztése, mely a WebMRI-t használva egy egyszerű felhasználó felületen keresztül teszi lehetővé az agy kiemelés és a lineáris regisztráció böngészőben való futtatását.



1. ábra – Az ábra a BrainBrowser szoftverarchitektúráját mutatja be, vastagon jelölve azokat a komponenseket („blend”: két térfogat összemosisát teszi lehetővé, „dicom”: DICOM térfogatok betöltését segíti, „pluginys”: plugin rendszer belépési pontja), amelyek a WebMRI fejlesztésével kerültek be.

FSL BET és FLIRT portolása WebAssembly-re

A FSL egy modulárisan felépített szoftvercsomag, melyben a BET és FLIRT mint különálló parancssori programok vannak definiálva. Az FSL nagyrészt C és C++ programnyelven íródott a teljesítmény maximalizálása végett. Hogy ezeket a komponenseket portolhassuk, módosítanunk kellett a programkód egyes részeit, illetve azokat a szkripteket (Makefile), amelyek a programok fordítását végzik. A portolás automatizálásához az Emscripten nevű fordítóprogramot használtuk, mely a Low Level Virtual Machine (LLVM) infrastruktúra alapján készült, és LLVM bitcode-ot képes WebAssembly-re fordítani. Az Emscripten nagyban megkönnyítette a portolási folyamatot, hisz az eredeti C/C++ kódbázist csak minimálisan kellett módosítani. Hogy a portolt programok a felhasználói felület reszponzivitását ne blokkolják, azokat úgynevezett háttérszálon futattuk, Web Worker-ök alkalmazásával.

DICOM támogatása

A BrainBrowser többféle neurológiai fájlformátum (NIFTI, MINC, MGH) kezelést is lehetővé teszi, a DICOM-ot azonban nem támogatja. DICOM szeletek betöltéséhez azokat előbb egy külső programmal konvertálni kell egy a BrainBrowser által támogatott formátumra. A BrainBrowser-t kibővítették egy DICOM térfogatbetöltővel, mely egy szintén Emscripten-nel WebAssembly-re portolt C++ alapú szoftver, a [dcm2niix](https://github.com/rordenlab/dcm2niix) (<https://github.com/rordenlab/dcm2niix>) segítségével történik.

Plugin rendszer

A BrainBrowser-t kibővítettük egy plugin rendszerrel, melynek köszönhetően a portolt szoftvereinket (BET, FLIRT, DICOM térfogatbetöltő) könnyedén tudjuk integrálni. Minden plugint egy JavaScript Object Notation (JSON) file ír le. Ez a file definiálja a bővítmény által támogatott paramétereket, azok típusát, és nevét, melyekből a WebMRI GUI generáló rendszere kirajzolja a plugin felhasználói felületét, és futtatja a pluginhoz regisztrált programot.

XReport

Az XReport webalkalmazás fejlesztése során két programozási nyelvet használtunk: JavaScriptet és TypeScriptet. A projekt két részre osztható, a könyvtárra és az alkalmazásra. A könyvtár egy önálló, újrahasznosítható JavaScript szoftvercsomag (más programokba is integrálható), mely a sablonkészítéshez és a leletezéshez szükséges fő funkciókat foglalja magában. Az alkalmazás egy Single Page Application (SPA), mely a könyvtárat használva jeleníti meg a sablonokat, a sablonkészítőt és a leletező felületeket.

A könyvtár

Az XReport könyvtár belépési pontja a „makeWidget” függvényhívás, mely Uniform Resource Locator (URL) megadása nélkül egy új sablon készítését, URL megadásával pedig egy már létező sablon betöltését teszi lehetővé. A könyvtárban specifikus Document Object Model (DOM) elemek vagy DOM elemek összetételei használhatóak a sablon megalkotásához.

Létrehoztunk egy FormScript nevű doménspecifikus nyelvet (Domain Specific Language, DSL), hogy a sablonjaink dinamikus viselkedése támogatott legyen Cross Site Scripting (XSS) támadásoktól védett módon. Ha engedélyoznénk, hogy a felhasználók tetszőleges JavaScript kódot illesszenek a template-ekhez, az biztonsági kockázatot jelentene. A FormScript garantálja, hogy csak az implementált műveletek hajthatók végre a sablonokban, és semmi más. A sablonhoz rendelt FormScript kódban leírt parancsoknak és feltételeknek megfelelően a mezők eltűnése, megjelenése, kitöltése és egyéb tulajdonságainak változása automatizálható.

Az alkalmazás

Az XReport alkalmazást az Angular SPA, és a Bootstrap Cascading Style Sheets (CSS) keretrendszereket használva készítettük el, és keverten JavaScript és TypeScript programnyelveken írtuk. Az oldal üzemeltetéséhez a Google Firebase ingyenes webtárhely szolgáltatását használtuk.

LLM alapú strukturált leletezés

A GPT4 LLM-mel végzett kísérletünk során egy szabadszöveges leletrészlet automatikus strukturálását végeztük el az alábbi prompt-ot megadva:

We will provide a free text radiological report, and a structured radiological report template. Convert the free text format to the template format.

Free text:

A polypoid, mucinous mass is visible 3mm from the mesorectal fascia. The mass infiltrates the external sphincter. The length of the mass is 50mm. There are 5 enlarged lymph nodes, 2 of them less than 5 mm in size, and 3 of them larger than 9mm. There are 2 tumor deposits inside the mesorectal fascia.

Template:

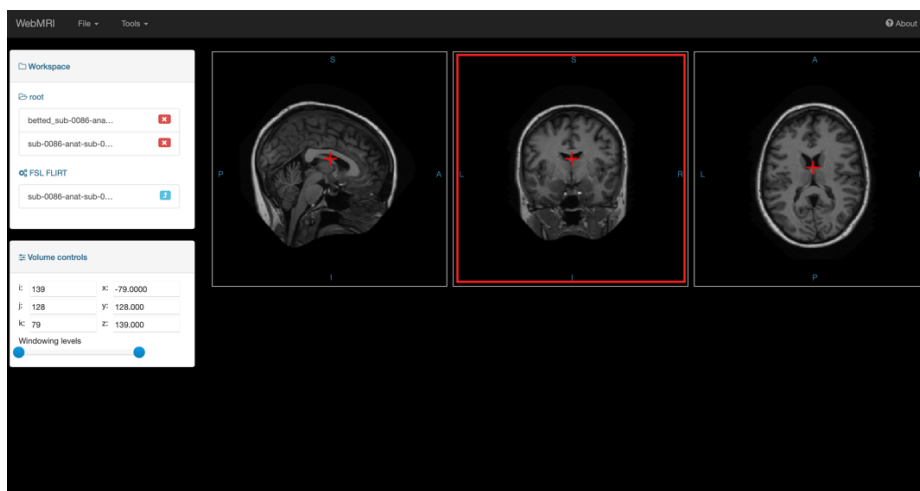
```
{
„morphology”: [„polypoid”, „ulcerating”, „circumferential”, „semicircumferential”],
„selectedMorphology”: „”,
„appearance”: [„mucinous”, „non-mucinous”], „selectedAppearance”: „”,
„lengthOfTumor”: { „value”: „”, „unit”: „mm” },
„sphincterInvolvement”: [„none”, „internal sphincter is involced”, „intersphincteric space
is involved”, „external sphincter is involved”], „selectedSphincterInvolvement”: „”,
„distanceTumorMesorectalFascia”: { „value”: „”, „unit”: „mm” }
„lymphNodeInvolvement”: False,
„numEnlargedLymphNodes”: 0,
„numLymphNodesLessThan5mm”: 0,
„numLymphNodesGreaterThan9mm”: 0,
„tumorDepositInsideMesorectalFascia”: False,
„numTumorDeposites: 0
}
```

Eredmények

WebMRI

A WebMRI demo alkalmazásunk kezdőképernyője (2. ábra) jobbról a betöltött térfogatot mutatja sagitális, koronális és axiális metszetekben, balról pedig a betöltött, és a pluginok által generált térfogatok listanézete látható. A listanézet alatti GUI elemekben a kép ablakolását és egyéb módosításokat lehetővé tevő csúszkák és inputok jelennek meg.

A fenti menüsorban három menüelem látható: „File”, „Tools” és „About”. A „File” menüponton keresztül tállózhatóak be a feldolgozni kívánt fájlok, a „Tools” menüben a programba betöltött pluginok érhetőek el, az „About” menüpontra kattintva pedig a szoftver használatáról és továbbfejlesztésének lehetőségeiről információt nyújtó weboldal látogatható meg.



2. ábra – A képen a WebMRI felhasználói felülete látható. Fent a menüsor jelenik meg, balra fent a menüsor alatt egy ablak („Workspace”), ami a betöltött és a pluginok által generált fájlokat mutatja, alatta („Volume controls”) a betöltött kép manipulálását és navigációt lehetővé tevő elemek látszanak, jobbról pedig a betöltött térfogat sagitális, koronális és axiális metszetekben.

A portolt pluginok futási idejének összevetése a natív verziókkal

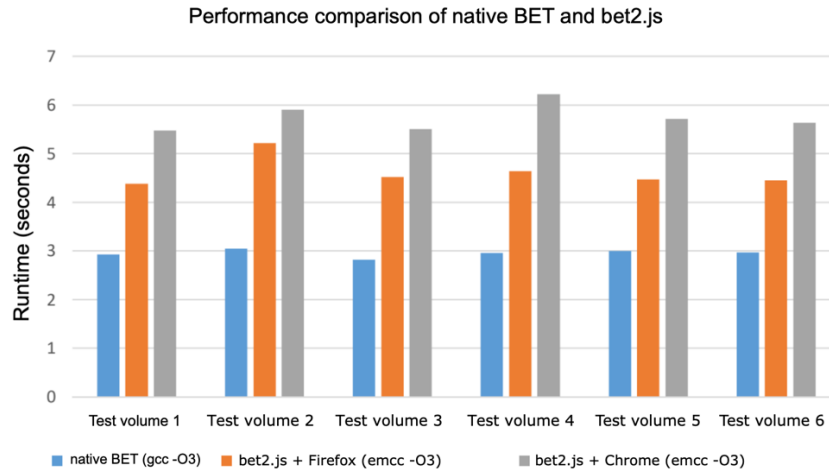
Összehasonlítottuk a WebAssembly-re portolt bet2.js és flirt.js teljesítményét a natív FSL algoritmusokkal futási idő szempontjából. Mindkét programot a legmagasabb -O3 optimalizálási szinten fordítottuk. A natív programokat Windows 10-es Linux alrendszeren futtattuk, míg a portolt verziókat Google Chrome és Mozilla Firefox böngészőkben. A teszteléshez használt számítógép specifikációi a következők:

- CPU: Intel® Core™ i5-3230 @ 2.60Ghz
- RAM: 6.00 GB
- Rendszer típusa: 64-bites operációs rendszer, x64-alapú processzor

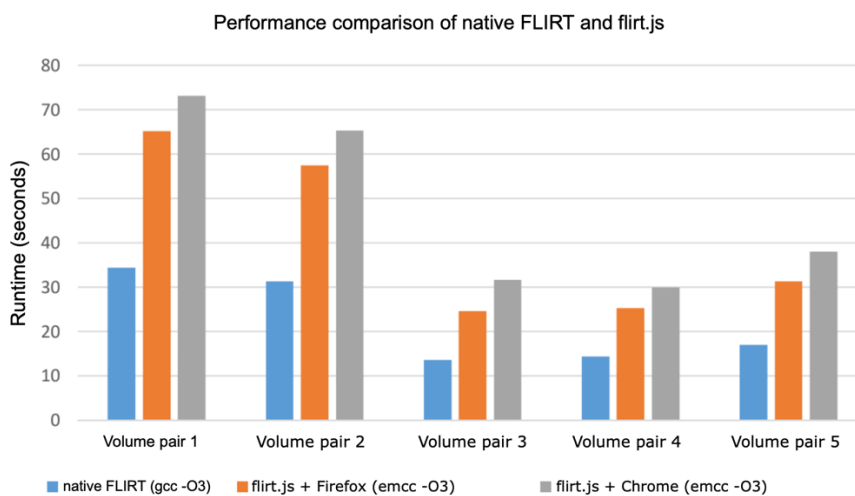
A bet2.js-t és a BET-et 6 MR térfogaton futattuk le oly módon, hogy mindegyiket 5x processzálta mind a natív, mind a portolt szoftver, ezzel kiküszöbölve a hideg indításból („cold start”) adódó varianciát. A natív BET átlagos futási ideje 2.96 másodperc volt. A bet2.js Google Chrome böngészőben átlagosan 5.75 másodperc, míg Mozilla Firefox-ban 4.62 másodperc alatt végzett a feldolgozással (3. ábra).

A flirt.js és a natív FLIRT összehasonlításához 10 MR térfogatot használtunk, melyek 5 párt alkottak. Mindegyik térfogatpár egyik eleme egy T1 súlyozott referencia, másik eleme pedig egy SWI térfogat volt, melyet az algoritmus a referenciához regisztrált. A T1-SWI párok mindig ugyanazon alanytól származtak. A feldolgozott fájlok NIFTI formátumúak voltak, és előfeldolgozási lépésként egy kiemelést alkalmaztunk rajtuk. A hideg indítási variancia kiküszöbölésére az előbbiekben ismertetett módon jártunk el.

A natív FLIRT átlagosan 22.12 másodperc alatt végezte el a regisztrációt. A WebAssembly verzió Firefox-ban futtatva átlagosan 40.79 másodperc alatt, Chrome-ban 47.64 másodperc alatt végzett (4. ábra).



3. ábra – A natív BET és a bet2.js programok futási idejének összehasonlítása. A függőleges tengelyen a futási időt mutatjuk másodpercekben kifejezve. A 6 oszlop csoport a 6 teszt térfogat feldolgozási idejeit mutatja: a kék oszlop a natív, a narancssárga a Firefox-ban, és a szürke a Google Chrome-ban mért futási időt ábrázolja.

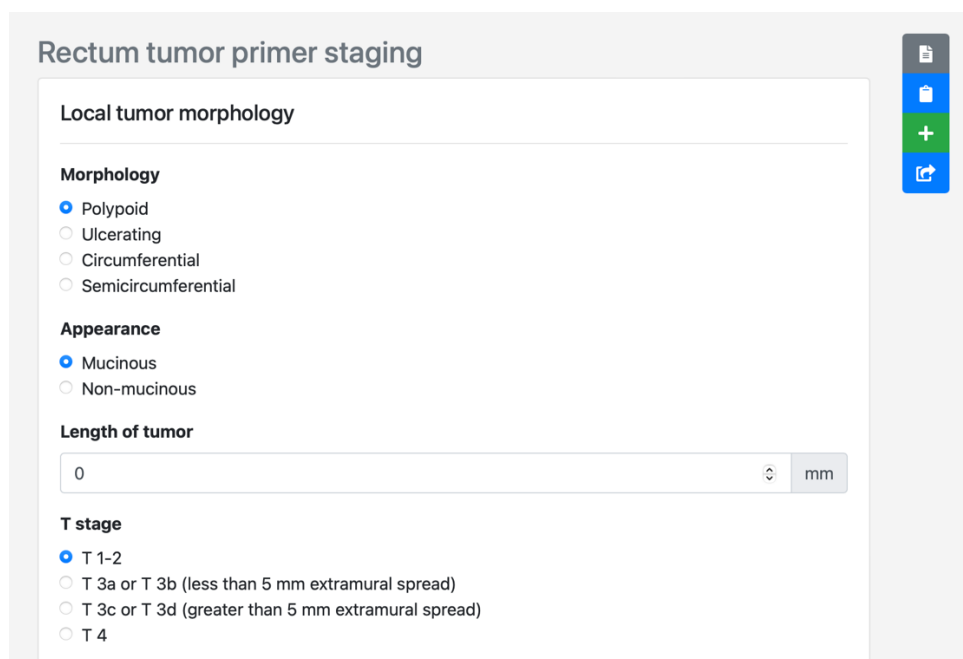


4. ábra – A natív FLIRT és a flirt.js programok futási idejének összehasonlítása. A függőleges tengelyen a futási időt mutatjuk másodpercekben kifejezve. Az 5 oszlop csoport az 5 térfogatpár feldolgozási idejeit mutatja: a kék oszlop a natív, a narancssárga a Firefox-ban, és a szürke a Google Chrome-ban mért futási időt ábrázolja.

XReport

Elkészítettük az XReport alkalmazást, melynek forráskódját Github-on Massachusetts Institute of Technology (MIT) liszensz alatt tettük közzé.

A program sablonkészítő és leletező modulja az 5. ábrán látható. Felül a sablon neve, alatta pedig maga a template tartalma látható. A sablon mellett jobb oldalt 4 gomb jelenik meg: a felső dokumentum ikon előhozza azt a nézetet, melyben a generált leletet jeleníti meg a szoftver. Fentről a második ikon a lelet tartalmát vágólapra másolja, így könnyen beilleszthető más szoftverek szövegdobozába. A zöld háttéren megjelenő fehér pluszjel egy új leletezést indít, a legalsó ikonra kattintva pedig a sablon osztható meg (nem a generált lelet, hanem a sablonhoz tartozó link kerül vágólapra).



The screenshot shows a web-based form titled "Rectum tumor primer staging". The form is divided into several sections:

- Local tumor morphology**
 - Morphology**
 - Polypoid
 - Ulcerating
 - Circumferential
 - Semicircumferential
 - Appearance**
 - Mucinous
 - Non-mucinous
 - Length of tumor**
 - Input field: 0
 - Unit: mm
 - T stage**
 - T 1-2
 - T 3a or T 3b (less than 5 mm extramural spread)
 - T 3c or T 3d (greater than 5 mm extramural spread)
 - T 4

On the right side of the form, there is a vertical toolbar with four icons: a document icon (top), a trash icon, a plus sign, and a share icon.

5. ábra – Egy rektum tumor primer staging sablon az XReport leletező felületén megjelenítve.

Template készítési módban a leletezési nézet egyes elemei megváltoznak. A jobb oldali gombcsoport 4 gombja helyett csak 2 lesz látható: sablon mentése, és sablon elvetése. Ezen felül a sablon legalján középen egy plusz gombot rajzol ki a program, amelyre kattintva egy újabb sor lesz illeszthető a sablonba. A sorok fölé húzva a kurzort lehetőség nyílik a sor duplikálására vagy törlésére, az egyes mezőket kijelölve pedig az adott elem szerkesztési lehetőségei fognak megjelenni.

A sablonokhoz rendelhető FormScript kódrészleteket egy szövegdobozban lehet szerkeszteni, mely a lelet jobb felső sarkában megjelenő fogaskerék ikonra kattintva hívható elő.

LLM alapú strukturált leletezés

A módszerek részben bemutatott prompt-ra a GPT4 a következő strukturált leletet generálta:

```
{
  "morphology": ["polypoid", "ulcerating", "circumferential", "semicircumferential"],
  "selectedMorphology": "polypoid",
  "appearance": ["mucinous", "non-mucinous"],
  "selectedAppearance": "mucinous",
  "lengthOfTumor": { "value": 50, "unit": "mm" },
  "sphincterInvolvement": ["none", "internal sphincter is involced", "intersphincteric space is involved", "external sphincter is involved"],
  "selectedSphincterInvolvement": "external sphincter is involved",
  "distanceTumorMesorectalFascia": { "value": 3, "unit": "mm" },
  "lymphNodeInvolvement": true,
  "numEnlargedLymphNodes": 5,
  "numLymphNodesLessThan5mm": 2,
  "numLymphNodesGreaterThan9mm": 3,
  "tumorDepositInsideMesorectalFascia": true,
  "numTumorDeposites": 2
}
```

Megfigyelhető, hogy az információkinyerés a promptban megadott JSON struktúra alapján történt. A „selectedMorphology” és a „selectedSphincterInvolvement” mezőkbe a megfelelő listákból kerültek kiválasztásra a helyes értékek. A mértékegység kinyerése is sikeres volt a „distanceTumorMesorectalFascia” és a „lengthOfTumor” mezőkbe. A nyirokcsomók és egyéb elváltozások számára és méretére vonatkozó adatok extrakcióját is helyesen hajtotta végre az LLM, ami a „numEnlargedLymphNodes”, „numLymphNodesLessThan5mm”, „numLymphNodesGreaterThan9mm” és a „numTumorDeposites” példáján jól látható.

Diszkusszió

WebMRI

A WebMRI-vel létrehoztunk egy bővíthető, platformfüggetlen, nyílt forráskódú neurológiai képfeldolgozó platformot, amely agy kiemelés és lineáris regisztráció elvégzését is támogatja, és bármely modern webböngészőben futtatható külső bővítmények vagy webserverek nélkül. Az agy kiemeléshez a BET, a lineáris regisztrációhoz pedig a FLIRT FSL szoftvercsomagot használtuk. Emscripten segítségével a natív programokat WebAssembly-re fordítottuk. A BrainBrowse-be egy új plugin rendszert fejlesztettünk, mely lehetővé teszi, hogy a portolt eszközeinket futtathassuk. Létrehoztunk egy demó alkalmazást is, amely a WebMRI-t használva egy komplett neurológiai képfeldolgozó munkafolyamat elvégzését implementálja, a DICOM szeletek betöltéstől a lineáris regisztrációig.

Szoftverrendszerünkkel demonstráltuk, hogy nem csak a neurológiai vizualizáció, hanem a képfeldolgozás is lehetséges modern böngészőkben. Szemben a Slicer 3D-vel, vagy más nem böngésző alapú programokkal, a megoldásunk nem igényel telepítést. Szoftverünk korlátai közé tartozik a böngészők limitált erőforrásai miatti teljesítményromlás a natív futáshoz képest. A jövőben további neurológiai képfeldolgozó algoritmusok portolását tervezzük és célunk a rendszerünk integrációjának egyszerűsítése klinikai PACS környezetekbe.

XReport

Az XReport-tal létrehoztunk egy ingyenes, platformfüggetlen, nyílt forráskódú strukturált leletezési szoftverrendszert radiológusok számára. Szoftverünk lehetővé teszi, hogy a felhasználók könnyen tudjanak új lelet sablonokat létrehozni, és leletezni. Programunk tervezésénél a modularitásra törekedtünk, így a rendszer magját egy külön könyvtárba szerveztük ki, melynek köszönhetően a leletezési és sablonkészítési funkciók egyszerűen integrálhatóak bármely más webes alkalmazásba. Ezen felül egy FormScript nevű egyszerű DSL-t is megalkottunk a sablonok dinamikus működéséhez. Reszponzív sablonjaink könnyen integrálhatóak bármely HIS rendszerbe. Az RSNA által fejlesztett ingyenes szolgáltatáshoz képest a mi megoldásunk egyszerűbb és felhasználóbarátabb, főként a dinamikus szkriptelhetőségnek, és a rezponzív felhasználói felületnek köszönhetően. A mi megoldásunk nyíltforráskódú, így bárki által könnyedén továbbfejleszhető, szemben a

RadReporttal, ami zárt. Az XReportban fellelhető sablonok száma elenyésző a RadReport könyvtárához képest, azonban ez a platformunk terjedésével változhat.

A GPT4-gyel végzett kísérletünk mutatja, hogy az LLM-ek nyelvi feldolgozási képességei rendkívül előrehaladottak, és a szabadszöveges leletezés és a strukturált leletezés közötti kapcsolatot jelenthetik.

Új tudományos eredmények összefoglalása

1, Kifejlesztettük a WebMRI-t, egy webalapú, modularis idegtudományi képfeldolgozó platformot. A FSL BET és FLIRT (agy kivonás és lineáris regisztráció) képfeldolgozó algoritmusokat átportoltuk WebAssembly-re, hogy böngészői környezetben is futtathatók legyenek. Tudomásunk szerint nem létezik más webes port ezekhez az FSL eszközökhöz. Összehasonlítottuk az eredeti és az átportolt FSL eszközök futási idejét, és arra jutottunk, hogy mindennapi használat során a mi verzióink nem számottevően lassabbak az eredeti programoknál. A DICOM formátumot is támogatjuk a WebMRI-vel, így kiküszöbölve a külső program általi DICOM-NIfTI konverziós lépés szükségességét. Fejlesztettünk egy plugin rendszert, amely lehetővé teszi más programozók számára, hogy új algoritmusokat hozzanak létre vagy meglévőket portoljanak, és beépítsék azokat a WebMRI platformunkba.

2, Kifejlesztettük az XReport-ot, egy ingyenes és nyílt forráskódú, webalapú strukturált leletezési platformot radiológusok számára, amely támogatja mind a sablonkészítést, mind a leletezést felhasználóbarát módon. Fejlesztettünk egy LLM-alapú megoldást automatikus strukturált leletezési sablon kitöltéshez szabad szöveges leletekből, prompt-engineering technikák segítségével.

Köszönetnyilvánítás

Szeretném őszinte hálámat és köszönetemet kifejezni programvezetőmnek, Prof. Dr. Bogner Péternek az évek során nyújtott támogatásáért és útmutatásáért.

Nagyon hálás vagyok témavezetőmnek, Dr. Tóth Arnoldnak a támogatásért és szakmai kiválóságáért. Segített a WebMRI koncepcionálásában, és az FSL eszközök használatában szerzett tapasztalatai révén értékes visszajelzéseket és iránymutatásokat adott a szoftverünk fejlesztése és tesztelése során.

Hálás vagyok Dr. Kövér Ferencnek az XReport fejlesztésében való részvételéért. Segített népszerűsíteni az XReportot a Pécsi Diagnosztikai Központ kollégái körében, és több sablont is készített a programban. Visszajelzése és útmutatása nagyon értékes volt az XReport fejlesztésében és finomhangolásában.

Hálás vagyok Dr. Szukits Sándornak az XReport fejlesztésében való közreműködéséért. Más strukturált leletező platformok használatában való jártassága sokat segített programunk elkészítésében.

Hálás vagyok a PTE ÁOK Orvosi Képző Klinikai és a Pécsi Diagnosztikai Központ munkatársainak. Visszajelzéseik sokat segítettek, hogy az XReport jobb és megbízhatóbb legyen.

Végül, de nem utolsó sorban szeretném megköszönni feleségemnek, Margarétának és fiaimnak, Ádámnak, Sebestyénnek és Bendegúznak a szeretetüket és támogatásukat. Hálás vagyok az egész családomnak, akik támogattak PhD-s éveim alatt.

Publikációs jegyzék

Tudományos közlemények száma: 3

Kumulatív impakt faktor: 3,268

A doktori disszertációm alapjául szolgáló közlemények száma: 2 (**IF: 3,268**)

1. **A. Harmouche**, F. Kövér, S. Szukits, T. Dóczi, P. Bogner, A. Tóth, “XReport: An online structured reporting platform for radiologists,” *SoftwareX*, vol. 17, p. 100993, Jan. 2022, doi: 10.1016/j.softx.2022.100993.

Kvartilis: Q2, Impakt faktor: **2.868** (2023)

2. **A. Harmouche**, F. Kövér, S. Szukits, T. Dóczi, P. Bogner, A. Tóth, “WebMRI: Brain extraction and linear registration in the web browser,” *Imaging*, vol. 15, no. 1, pp. 31–36, Jun. 2023, doi: 10.1556/1647.2023.00111.

Kvartilis: Q4, Impakt faktor: **0.4** (2023)

Egyéb közlemények

G. Jandó, E. Mikó-Baráth, A. Czigler, **A. Harmouche**, I. Szabó, L. Závor, and D. P. Piñero, “Amblyopia screening with the dynamic random dot stereotest,” *Ophthalmology Times Europe*, vol. 16, no. 7, Sept. 2022