**PSIV**

Aquest treball se centra en la detecció d’Helicobacter pylori, un bacteri gàstric d'alta prevalença mundial, classificat com a carcinogen de classe 1. Tot i que sovint és asimptomàtic, aquest bacteri causa inflamació de la mucosa gàstrica (gastritis) i està associat amb condicions greus com el càncer gàstric. Per tant, la seva detecció precisa és fonamental en l’àmbit mèdic per un tractament precoç.

Actualment, la tècnica de diagnòstic preferida es basa en l’anàlisi d’imatges histopatològiques amb tinció immunohistoquímica. No obstant, aquest mètode és molt exigent en temps i recursos, ja que requereix que un patòleg expert inspeccioni manualment mostres de gran mida (120.000 x 16.000 píxels) amb densitats variables del bacteri.

L’objectiu d’aquest projecte és automatitzar la detecció d’H.pylori mitjançant tècniques d’aprenentatge profund. On hem dut a terme dos enfocaments complementaris:

* **SISTEMA 1- Detecció d’anomalies amb autoencoders i adaptive thresholding**

Utilitzar codificadors per aprendre patrons latents de teixit sa de manera no supervisada, per així detectar H.pylori com una anomalia en la tinció d’imatges.

* **SISTEMA 2- Classificació amb mecanismes d’atenció**

Implementar mecanismes d’atenció per representar l’espai latent i millorar la classificació a nivell de pacient.

**DATASET:**

Aquest estudi utilitza una base de dades clínica que prové del Departament de Patologia de l’Hospital Universitari General de Catalunya – Grup Quironsalud. Aquesta base inclou un total de 245 biòpsies gàstriques, les quals han estat classificades per un patòleg segons la densitat d’Helicobacter pylori en tres categories: NEGATIVA, BAIXA DENSITAT I ALTA DENSITAT.

Dels 245 pacients inclosos, 117 pacients (47’8%) són NEGATIUS i 128 pacients (52’2%) són POSITIUS, tant densitats classificades com baixa o alta.

Com les imatges originals són de gran mida, es van retallar en pegats de

256x256 píxels per facilitar l’anàlisi. Aquestes imatges es guarden en format .png, organitzades en carpetes etiquetades amb el codi d’identificació del pacient (PatID) i la secció del teixit corresponent (PatID\_Section#).

Tenim dos sets principals:

* Cross-validation set
* Hold-out set: Inclou els 116 casos restants i serveix com a conjunt independent per verificar el rendiment/reproductibilitat dels mètodes de diagnòstic del pacient.

**Cross-Validation Set:**

Aquest set es divideix en dues categories:

* **CROPPED (retallat):**

Retalls extrets de tota la secció del teixit per entrenar els models de diagnòstic a nivell de pacient. On les seccions es classifiquen segons la densitat d’H.pylori: NEGATIVA, BAIXA i ALTA DENSITAT. La informació es recull al fitxer PatientDiagnosis.csv.

* **ANNOTATED (anotats):**

Conté pegats marcats manualment per un expert per entrenar mètodes de classificació a nivell de pacth. En total, inclou 1.211 pegats, dels quals 161 són positius. Aquest subconjunt correspon a 123 casos, dels quals 77 són positius.

Les coordenades dels pegats anotats (i, j, w, h) i la seva classificació es troben al fitxer HP\_WSI-CoordAnnotatePatches.xlsx. La presència està definida com a:

* -1 = No
* 1 = Si
* 0 = No clar

El fitxer només inclou pegats anotats. Els pegats restants es consideren negatius per defecte.

Tabla

Descripción generada automáticamente

**DATA LOADERS:**

* **LOADCROPPED()**

Carregar imatges retallades del conjunt de dades de diagnòstic a nivell de pacient. Aquestes imatges estan classificades segons la densitat d’Helicobacter pylori.

Entrades:

* ListFolders: una llista amb els noms de les carpetes que contenen les imatges dels pacients.
* nImages: el nombre d’imatges a carregar per cada pacient.

El procés és:

Llegir el fitxer PatientDiagnosis.csv que conté la classificació dels pacients en funció de la densitat. Iterar sobre les carpetes d’imatges: Carregar les imatges en format .png. i assegurar-se que les imatges tenen la mida correcta (256x256), ja que algunes no ho eren i provocaven errors. Si no compleixen aquest criteri, es descarten.

Aleshores assignar una etiqueta a cada imatge. Fent-ho utilitzant el codi del pacient (CODI) per buscar el seu diagnòstic en el fitxer CSV i convertir la classificació textual en etiquetes numèriques:

NEGATIVA → -1

BAIXA i ALTA → 1

Acabar processant les imatges pel format esperat pels models d’aprenentatge profund.

És a dir, convertir-les al format RGB, normalitzar els valors dels píxels entre 0 i 1, i

transposar la imatge: img = np.transpose(img, (2, 0, 1)).

Llavors guardar la imatge i les metadades en dues llistes separades.

* PatientsImgs: les imatges processades.
* patientsMeta: les metadades [ID pacient, ruta de la imatge, diagnòstic].
* **LOADANNOTATED()**

Carregar imatges anotades manualment per un expert per a tasques de classificació a nivell de patch.

Entrades igual que loadCropped: “ListFolders” i “nImages”. Amb un procés semblant, llegir el fitxer HP\_WSI-CoordAnnotatePatches.xlsx que conté les coordenades i etiquetes de presència d’H. pylori per a patches específics.

Aleshores filtres les dades segons la carpeta actual, extreure el Pat\_Id i el Section\_ID del nom de la carpeta i filtrar les files del fitxer Excel que coincideixin amb aquests valors.

Processar les imatges de la carpeta (format .png) i assegurar-se que es redimensionen a 256x256. Per després comparar el nom del fitxer de la imatge amb el Window\_Id del Excel, eliminant els zeros inicials per assegurar les coincidències. Llavors buscar l’etiqueta presència (presence) corresponent a l’Excel (-1=no present, 0= no clara, 1=present). Hi ha casos extra que no es troben en l’Excel, ja que corresponen tots a patches negatius.

Per acabar guardant les imatges i metadades en dues llistes com LoadCropped():

annotatedImgs i annotatedMeta (ID pacient, ruta de la imatge, presència).

**SISTEMA 1**

**METODOLOGIA SISTEMA 1:**

* Data preparation



* Patch classification

Gráfico

Descripción generada automáticamente

* Patient diagnosis

**AUTOENCODER:**

Pel primer sistema la idea principal és identificar la presència d’H.pylori detectant canvis en la tinció del teixit. On per fer-ho hem dissenyat un autoencoder que aprèn només els patrons de teixit sa. Llavors l’objectiu és que l’autoencoder pugui reconstruir amb precisió el teixit normal, però els de teixit infectat amb anomalies.

Un autoencoder és un tipus de xarxa neuronal no supervisada amb una estructura de codificador i descodificador. El codificador redueix la informació de les dades d’entrada a una representació comprimida o espai latent (menys dimensions). Per altra banda, el descodificador reconstrueix les dades originals a partir de l’espai latent. Aquest procés es guia amb l’objectiu de minimitzar l’error entre la imatge original i la reconstruïda, cosa que permet a l’autoencoder aprendre les característiques més rellevants de les dades.

En el nostre cas s’entrena l’autoencoder només amb imatges extretes de pacients sense H.pylori. Llavors aprèn a reconstruir teixits “normals” amb gran precisió. Té l’avantatge llavors que és no supervisat, no necessita dades anotades, només imatges de teixit normal per entrenar l’autoencoder. On estem treballant entrenant amb les dades cropped. Aleshores quan reconstrueixi imatges amb teixit infectat, com no ha après patrons d’infecció, la reconstrucció no serà fidel. En si els tons vermellosos, associats a la presència d’H.pylori es perden, i el color passa a tons blaus en la reconstrucció. On les diferències entre la imatge original i la reconstruïda s’utilitzen per detectar l’anomalia, és a dir, la presència d’H.pylori.

**Arquitectura de l’autoencoder**:

Variables principals del codi:

* Net\_paramsEnc: paràmetres per al codificador com el nombre de neurones en cada capa i el percentatge de “drop-out”
* Net\_paramsDec: Paràmetres per al descodificador com les configuracions de les capes i strides.
* Inputmodule\_paramsEnc: especifica els paràmetres d’entrada, com el nombre de canals d’entrada de les imatges.

Les arquitectures es configuren segons el valor de paràmetre Config per a la funció AEConfigs, que especifica la configuració desitjada (1, 2 o 3).

**CONFIG 1: senzilla, pocs nivells de compressió**

Codificador:

* 2 blocs:
  + Bloc 1:32 filtres strides [1,2]
  + Bloc 2: 64 filtres strides [1,2]
* Dropout: 20% de les connexions s’eliminaran aleatòriament durant l’entrenament

Descodificador: arquitectura inversa del codificador, 2 blocs. 64-32 filtres. 32- 3 canals d’entrada RGB.

**CONFIG 2: model pot capturar + detalls i més nivells de compressió**

Codificador:

* 4 blocs:
  + Bloc 1: 32 filtres stride 2
  + Bloc 2: 64 filtres stride 2
  + Bloc3: 128 filtres stride 2
  + Bloc 4: 256 filtres stride 2

Descodificador. Arquitectura inversa. 4 blocs. 128-64-32-3.

**CONFIG 3:** arquitectura intermèdia. Balanç entre complexitat i simplicitat.

Codificador:

* 3 blocs:
  + Bloc 1: 32 filtres stride 1
  + Bloc 2: 64 filtres stride 2
  + Bloc 3: 64 filtres stride 2

Descodificador. Arquitectura inversa. 3 blocs. 64-32-3.

**PARÀMETRES ENTRENAMENT:**

* Criterion: funció de pèrdua utilitzada per avaluar qualitat de reconstrucció=MSE
* Num\_epochs
* Learning\_rate
* Batch\_size

**CODI AUTOENCODER PROCÉS:**

* Configura l’arquitectura del model segons la configuració
* Crea i inicialitza el model d’autoencoder.

(((Assegurar que si es disposa d’una GPU, el model s’entrenarà a la GPU (cuda))))

* Utilitza l’optimitzador Adam i la funció de pèrdua MSELoss per ajustar els pesos. On Adam ajusta automàticament la velocitat d’aprenentatge per a cada paràmetre. 1e-3 és el learning rate que defineix la magnitud dels ajustaments dels pesos durant l’entrenament.
* Entrena el model iterativament durant diverses èpoques. On per cada batch, realitza una passada cap endavant i cap enrere. A més, actualitza els pesos del model per reduir l’error de reconstrucció.

(((Model.train()

I bucle d’entrenament per èpoques. I després les dades d’entrenament s’agrupen en batches (bact\_size) mitjançant el dataloader. )))

(((Hi ha una passada cap enrere amb optimizer.zero\_grad(), loss.backward() i optimizer.step() on es reinicien els gradients acumulats dels pesos, es calcula el gradient de la funció de pèrdua respecte als pesos i l’optimitzador actualitza els pesos segons els gradients calculats. )))

* Monitora la pèrdua per època.
* Després de l’entrenament transferim el model a CPU i desem els pesos entrenats del model per utilitzar-los posteriorment.

Tenir en compte: Abans de passar la imatge pel autoencoder, les hem de preparar convertint-la a RGB, normalitzar en [0,1], convertir a PIL image necessari per transformacions i transformem imatge a tensor amb transform. ((unsqueeze(0)).

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**ADAPTIVE THRESHOLDING: RECONSTRUCTION ERROR**

Després de fer l’entrenament del model i tenint els pesos guardats podem passar a la part d’implementar la detecció de H.pylori mitjançant la comparació de píxels vermellosos entre imatges originals i reconstruïdes amb l’autoencoder. Pe acabar fent un anàlisi ROC per determinar un llindar òptim.

Tal com havíem comentat abans amb l’autoencoder, les imatges reconstruïdes amb teixit infectat, passen de tenir tons vermellosos a tons blaus, on llavors aquesta diferència de pèrdua de píxels vermellosos s’utilitza per detectar l’anomalia.

En altres paraules, la presència de H.pylori en una imatge es calcula utilitzant la reconstruction error, etiquetada com a fred, entre la imatge original i les reconstruïdes. En si la fracció de píxels vermellosos que es perden entre imatges originals i reconstruïdes.

En el nostre cas si Fred és major a 1.00, hi ha hagut una pèrdua de píxels vermellosos.

Els píxels vermellosos es calculen aplicant un filtre en l’espai de color HSV. On en aquest espai els píxels amb presència de H.pylori tenen un matís en el rang [-20,20], per la qual cosa l’àrea de píxels vermellosos es defineix com el nombre de píxels en aquest rang.

Llavors en el nostre codi els passos principals han sigut la conversió d’imatges a HSV i s’extreu el canal Hue. Definim el rang de [-20,20] per a píxels vermellosos, però de diferent manera, ja que en OpenCv els valors de Hue es representen en un rang circular de 0-179, no tracta amb negatius. S’ha de tenir en compte que ñes circular, el rang és 0-179 però si es travessa el límit com podria ser 180 a 20 se continua tractant de manera correcte.

Llavors es compten els píxels vermellosos i es calcula el fred com la fracció de píxels vermellosos = reconstructed\_red\_pixels / original\_red\_pixels.

En acabar de calcular tots els freds fem una corba ROC per determinar el llindar òptim. Aquesta corba analitza els valors de Fred i els seus labels reals. Les etiquetes amb valor 0 (presència no clara) s’exclouen per evitar contaminació de la corba Roc i llavors passar a binari el -1 i 1. On 1 per a positiu (presència), 0 per a negatiu (no presència).

Llavors generem la corba ROC i calculem les distàncies al punt ideal (0,1). On el llindar òptim és aquell amb la distància més curta. Obtenim de sortida la gràfica de la corba ROC, el llindar òptim per detectar la presència d’H.pylori i la taxa de false positives i true positives.

**VALIDATION:**

Per tema validació hem implementat un pipeline complet per validar un autoencoder amb validació k-fold, en específic 10 folds, en dues tasques: classificació de finestres (patch classification) i diagnòstic a nivell de pacient (patient diagnosis).

FLUX:

1. **Càrrega del model i les dades:**

El model preentrenat es carrega amb els seus pesos desats. I les dades es carreguen en dos conjunts:

* Cropped dataset: per a diagnòstic de pacients
* Annotated dataset: per a classificació de finestres (patch)

1. **Validació amb K-fold:**

Les dades es divideixen en train/test utilitzant el mètode K-fold. Es realitzen iteracions per cada fold, entrenant el model i avaluant-lo en tasques de classificació.

1. **Classificació de finestres:**

Es calcula el valor de Fred entre imatges originals i reconstruïdes. Es genera una corba ROC per determinar el llindar òptim de classificació i es calculen mètriques com precisió, recall, F1-score, confusión matrix per avaluar el rendiment.

Es guarda els thresholds òptims dels folds en OptimalThresholdsPatchClassification.

On X= patientsimgsannotated i Y= presència que està en patientsMetaAnnotated.

Guardem la partició dels pacients train/test del fold actual en patientsFolds.

Per l’entrenament tenim les variables llistes errors i labels per emmagatzemar els reconstruction errors I els true labels.

Passem les imatges input pel autoencoder.

Després per calcular el fred també hem de processar les imatges de sortida, eliminant la dimensió de batch, convertir la imatge en HWC, convertir-la a BGR.

Llavors guardem els freds i guardem el threshold òptim del fold actual.

EN TEST: Aquí tenim una llista per guardar les etiquetes reals i prediccions de la classificació: patchClassification.

Passem la imatge pel model, processem la imatge de sortida, calculem el f\_red i classifiquem el patch segons el optimalthreshold del fold actual calculat. On si supera el optimalthreshold és 1 (té H.pylori) i sinó -1. Els valors de Fred es comparen amb el llindar òptim per assignar etiquetes positives o negatives.

Llavors en les mètriques de classificació iterem sobre les classificacions reals i predites i calculem les mètriques de validació i les guardem.

1. **Diagnòstic de pacients:**

A partir de la classificació de Finestres, es calcula el percentatge de Finestres positives per pacient. Aquest percentatge es compara amb un llindar òptim per diagnosticar la presència d’H.pylori. Es generen mètriques iguals que Classificació de Finestres.

Es guarda els thresholds òptims dels folds en OptimalThresholdsPatientsDiagnosis.

Per a cada pacient, es calcula el percentatge de Finestres classificades com a positives. Aquest percentatge representa una probabilitat de presencia d’H.pylori. Igual que en la classificació de Finestres, es determina un llindar òptim basat en la corba ROC. Llavors la probabilitat de presencia es compara amb el llindar òptim per asignar una classificació positiva o negativa al pacient. En altres paraules, Les mostres amb un percentatge de fragments/finestres positives per sobre d’aquest llindar es diagnostiquen com a positives per H. pylori.

En una llista patientDiagnosis guardem els diagnòstics dels pacients reals i predits. A més d’un diccionari per comptar els diagnòstics positius. Processem, passem pel AE, processem, calculem fred i comprovem si el pacient està dins del diccionari, sinó l’afegim. I contem els patches dels pacients que estem validant. I comprovem si el patch del pacient supera el threshold fred òptim. Al final calculem el percentatge de finestres positives dels pacients.

Per acabar, calculant les mateixes mètriques que en la classificació de Finestres.

1. **Anàlisi dels llindars:**

Es presenten estadístiques descriptives (Mitjana, desviació estàndard, coeficient de variació) dels llindars òptims per a cada tasca.

**MÈTRIQUES:**

**Mètriques de rendiment que hem tingut en compte:** precisió, recall, F1-score, confusion matrix per cada fold ((MIRAR SEPARAR per diagnostic class: positive, negative)).

**FER!!!:**

* Per resum estadístic hem fet servir average +- standard deviation per mètriques de qualitat.
* Test estadístic: Anova, t\_test o IC. Per quins mètodes aquestes IC se superposen i podem dir que són iguals. Mirar el p-valor. Probabilitat que mètode 1 sigui igual a mètode 2.

De la Debora:

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente