

Detecció d'*Helicobacter pylori* en WSI mitjançant enfocament per *patches*: System 1 (Autoencoder) i System 2 (MIL + Attention)

Assignatura: Mètodes Avançats de Processament de Senyal, Imatge i Vídeo (MAPSIV)

Marc Cases

Álvaro Bello

Adrián Fuster

Namanmahi Kumar

Universitat Autònoma de Barcelona (UAB)

Grau en Enginyeria de Dades – 4t curs

30 de novembre de 2025

Resum

La detecció de *Helicobacter pylori* en biòpsies digitalitzades es basa sovint en imatges de molt alta resolució (Whole Slide Images, WSI), fet que obliga a treballar de manera eficient mitjançant *patches*. En aquest treball desenvolupem i avaluem dos sistemes complementaris.

En el **System 1** entrenem un *AutoEncoderCNN* exclusivament amb *patches* de pacients sans per aprendre un model de normalitat; la presència d'anomalies s'estima a partir de l'*error de reconstrucció*, incorporant discrepàncies cromàtiques (Hue/HSV) i un llindar adaptatiu ajustat amb validació creuada a nivell de pacient.

En el **System 2** proposem un *pipeline* amb supervisió feble basat en *embeddings* i *Multiple Instance Learning* (MIL) amb un mòdul d'*attention* per predir l'etiqueta del pacient sense requerir anotacions per *patch*.

Reportem resultats amb corbes ROC/AUC i mètriques de classificació tant a nivell de *patch* com de pacient, evitant *data leakage* mitjançant particionat estratificat per pacient.

1 Introducció

La histopatologia digital permet analitzar biòpsies mitjançant imatges de molt alta resolució coneぐides com a Whole Slide Images (WSI). Aquestes imatges contenen una gran quantitat d'informació (i també variabilitat visual de tinció, textura i artefactes), de manera que una pràctica habitual és dividir-les en regions petites o *patches* per poder entrenar models d'aprenentatge profund de forma eficient. En el cas de la detecció de *Helicobacter pylori* (*H. pylori*), la decisió final és intrínsecament **a nivell de pacient** (positiu/negatiu), tot i que l'evidència visual pot aparèixer de manera localitzada en zones concretes del teixit.

En aquest treball abordem el problema amb dos sistemes: (i) un enfocament no supervisat de **detecció d'anomalies** (System 1), on un autoencoder entrenat només amb *patches* sans produeix un *score* basat en l'error de reconstrucció; i (ii) un enfocament de **supervisió feble** (System 2) basat en *Multiple Instance Learning* (MIL) amb *attention*, que integra informació de múltiples *patches* per predir l'etiqueta del pacient sense necessitat d'etiquetes exhaustives per *patch*.

Per garantir una avaluació realista, evitem el *data leakage* utilitzant particions i validació creuada a nivell de pacient (i no a nivell de *patch*), i reportem corbes ROC/AUC i mètriques estàndard (exactitud, sensibilitat i especificitat) tant a nivell de *patch* com a nivell de pacient.

2 Dades

El dataset emprat està format per imatges de histopatologia digital organitzades a nivell de pacient. Cada pacient té una o diverses biòpsies en format WSI, de les quals s'extreuen *patches* RGB que s'utilitzen com a entrada dels models.

2.1 Fonts d'informació i etiquetes

Disposem de dues fonts principals d'informació:

- **Fitxer de diagnòstic de pacients** (`PatientDiagnosis.csv`): conté un total de 309 pacients, etiquetats segons la columna `DENSITAT` amb tres categories: 151 pacients `NEGATIVA` (sense `H. pylori`), 72 pacients amb densitat `BAIXA` i 86 pacients amb densitat `ALTA`.
- **Fitxer d'anotacions de *patches*** (`HP_WSI-CoordAllAnnotatedPatches.xlsx`): recull coordenades i etiquetes per *patches* extrets de WSI. La columna `Presence` pot prendre els valors -1 , 0 o 1 . En el nostre treball considerem només els *patches* amb `Presence` $\in \{-1, 1\}$, i descartem aquells amb `Presence` $= 0$ (casos “skip”).

Després de filtrar per `Presence` $\in \{-1, 1\}$, obtenim un total de 2691 *patches* anotats, dels quals 1475 són positius (`Presence = 1`) i 1216 són negatius (`Presence = -1`). Aquests *patches* corresponen a 154 pacients diferents.

2.2 Relació entre anotacions i diagnòstic de pacient

La informació de *patches* anotats es creua amb el diagnòstic de pacient mitjançant els identificadors `Pat_ID` (en l'Excel) i `CODI` (en el `PatientDiagnosis.csv`). Entre els 154 pacients que apareixen en l'Excel d'anotacions, 76 tenen diagnòstic `NEGATIVA` i 78 presenten algun grau de presència de `H. pylori` (`BAIXA` o `ALTA`). Això reflecteix que les anotacions inclouen tant zones sanes com zones amb presència del bacteri.

2.3 Resum quantitatiu

Taula 1: Resum del conjunt anotat utilitzat en l'avaluació del System 1.

Conjunt	# Pacients	# Patches
Annotated \cap Excel, $\text{Presence} \in \{-1, 1\}$	154	2691
Patches positius ($\text{Presence} = 1$)	44 ¹	1475
Patches negatius ($\text{Presence} = -1$)	151 ¹	1216

3 Metodologia

En aquesta secció descrivim com hem construït el System 1, que és el mòdul encarregat de detectar anomalies a partir de la reconstrucció d'un autoencoder convolucional (AE). La idea és que l'AE només veu imatges de pacients sans durant l'entrenament, i per tant aprèn la seva estructura i colors típics. Quan rep un *patch* que conté `H. pylori`, la reconstrucció és menys precisa i l'error és més alt.

¹Un mateix pacient pot tenir *patches* positius i negatius; per això la suma de pacients positius i negatius supera el total de 154 pacients únics.

3.1 Preprocessat de les imatges

Totes les imatges s’escalen a una mida fixa de 256×256 píxels i es passen a format RGB normalitzat entre 0 i 1. Aquest preprocessat és bàsic però suficient per entrenar el nostre model, i facilita el treball amb *batches* i la memòria de la GPU.

Els *patches* es carreguen com tensors PyTorch en format *CHW*, que és l’estructura habitual en models convolucionals.

3.2 Arquitectura de l’Autoencoder (Config 3)

Per al System 1 hem utilitzat una de les arquitectures d’autoencoder convolucional proposades per la professora Débora Gil. En concret, teníem tres configuracions possibles, amb diferents profunditats i amplades de blocs convolucionals. Després de revisar-les i fer diverses proves inicials, vam optar per la “Config 3” perquè és la que ens donava un millor equilibri entre capacitat de representació i risc de sobreajustament.

La Config 3 utilitza els següents blocs al **encoder**:

$$32 \rightarrow 64 \rightarrow 64$$

Això significa que, a mesura que la imatge passa pel model, es redueix la resolució i augmenta la quantitat de canals, comprimint la informació en un *bottleneck* relativament compacte. Aquesta configuració és prou expressiva per capturar patrons de teixit dels pacients sans, però al mateix temps és força lleugera comparada amb les altres configuracions proposades.

L’estructura global de l’AE és la següent:

- **Encoder:** seqüència de convolucions i estrides que redueixen progressivament la resolució.
- **Bottleneck:** representa la informació essencial del *patch*. Si la imatge és “normal”, aquest espai latent és suficient per reconstruir-la. Si conté anomalies (H. pylori), part de la informació es perd.
- **Decoder:** fa la inversió del procés anterior i intenta reconstruir la imatge original.

Aquesta arquitectura segueix l’enfocament de “model de normalitat”: l’autoencoder només veu imatges de pacients sans durant l’entrenament i, per tant, aprèn com han de ser els *patches* sans. Quan rep imatges amb H. pylori, la reconstrucció és pitjor i l’error augmenta, cosa que ens permet detectar anomalies.

A més, la professora ens va proporcionar tres configuracions d’autoencoder possibles. La Config 1 era la més senzilla i superficial, amb pocs canals i poca capacitat per captar detalls fins del teixit. La Config 2, en canvi, era la més profunda i amb més canals, però en les nostres proves inicials vam veure que tendia a sobreajustar-se i que requeria més memòria i temps d’entrenament.

La Config 3 representa un punt intermig entre les altres dues: té prou profunditat per capturar patrons rellevants del teixit, però no és tan pesada com la Config 2. Per això vam acabar escollint la Config 3 com a opció equilibrada, ja que s’adapta bé al nostre cas d’ús i permet entrenar l’autoencoder de forma estable i eficient.

3.3 Funció de pèrdua (Loss)

La funció de pèrdua combina dos termes:

1. **MSE en RGB:** error de reconstrucció de la imatge en l’espai habitual de color.

2. Error en Hue (HSV): comparem el to (Hue) entre la imatge original i la reconstruïda.

El canal Hue és important perquè el bacteri H. pylori acostuma a aparèixer en zones amb tonalitats vermelloses. Afegir aquesta pèrdua ens ajuda a penalitzar més les diferències en aquestes parts.

La pèrdua final és:

$$\mathcal{L} = \alpha \cdot \text{MSE}_{RGB}(x, \hat{x}) + \beta \cdot \text{MSE}_{HSV/H}(HSV(x), HSV(\hat{x}))$$

En el nostre cas fem servir $\alpha = 1.0$ i $\beta = 0.2$. El valor $\beta = 0.2$ ens ha funcionat bé perquè dóna pes al color, però sense dominar completament la pèrdua ni fer que el model s'acabi centrant només en el Hue. També apliquem distància circular al Hue per evitar problemes amb el salt entre 0° i 360° .

3.4 Details de l'entrenament

Per entrenar l'autoencoder hem utilitzat:

- **Optimitzador:** Adam amb `lr = 1e-3`.
- **Batch size:** 32.
- **Èpoques:** 10 (valor suficient per convergir sense sobreentrenar).
- **Divisió train/val:** 90% entrenament, 10% validació.

A cada època guardem el model amb millor `val_loss`, de manera que evitem sobreajustament. L'entrenament es realitza al clúster de la UAB mitjançant *Slurm*, llançant un script `sbatch` que reserva una GPU i monitoritza la sortida a fitxers `.out` i `.err`.

3.5 Avaluació amb K-Fold per pacients

Per avaluar l'AE no té sentit fer una simple divisió train-test a partir de *patches*, perquè múltiples *patches* provenen del mateix pacient. Per evitar *data leakage*, fem un **K-Fold** a nivell de pacient.

El procés per cada *fold* és:

1. Estratifiquem els pacients per tenir positius i negatius a cada fold.
2. Entrenem un llindar τ a partir dels errors dels pacients del fold d'entrenament (mètode de Youden o millor F1).
3. Apliquem aquest llindar al fold de test i calculem:
 - AUC i acc. a nivell de **patch**.
 - AUC i acc. a nivell de **pacient**.
4. L'agregació per pacient la fem amb el p95 de l'error dels seus *patches*, que ens ha funcionat millor que la mitjana o el màxim.

Finalment, calclem la corba ROC mitjana de tots els folds i la seva desviació estàndard, cosa que ens dóna una estimació més robusta de com generalitza el model a pacients que no ha vist mai.

4 Resultats del System 1

4.1 Reconstrucció de patches sans

A la Figura 1 es mostren alguns *patches* sans originals. A la Figura 2 es veu que l'autoencoder els reconstrueix de manera fidel, mantenint la textura, els nuclis i els tons blavosos característics del teixit sa.

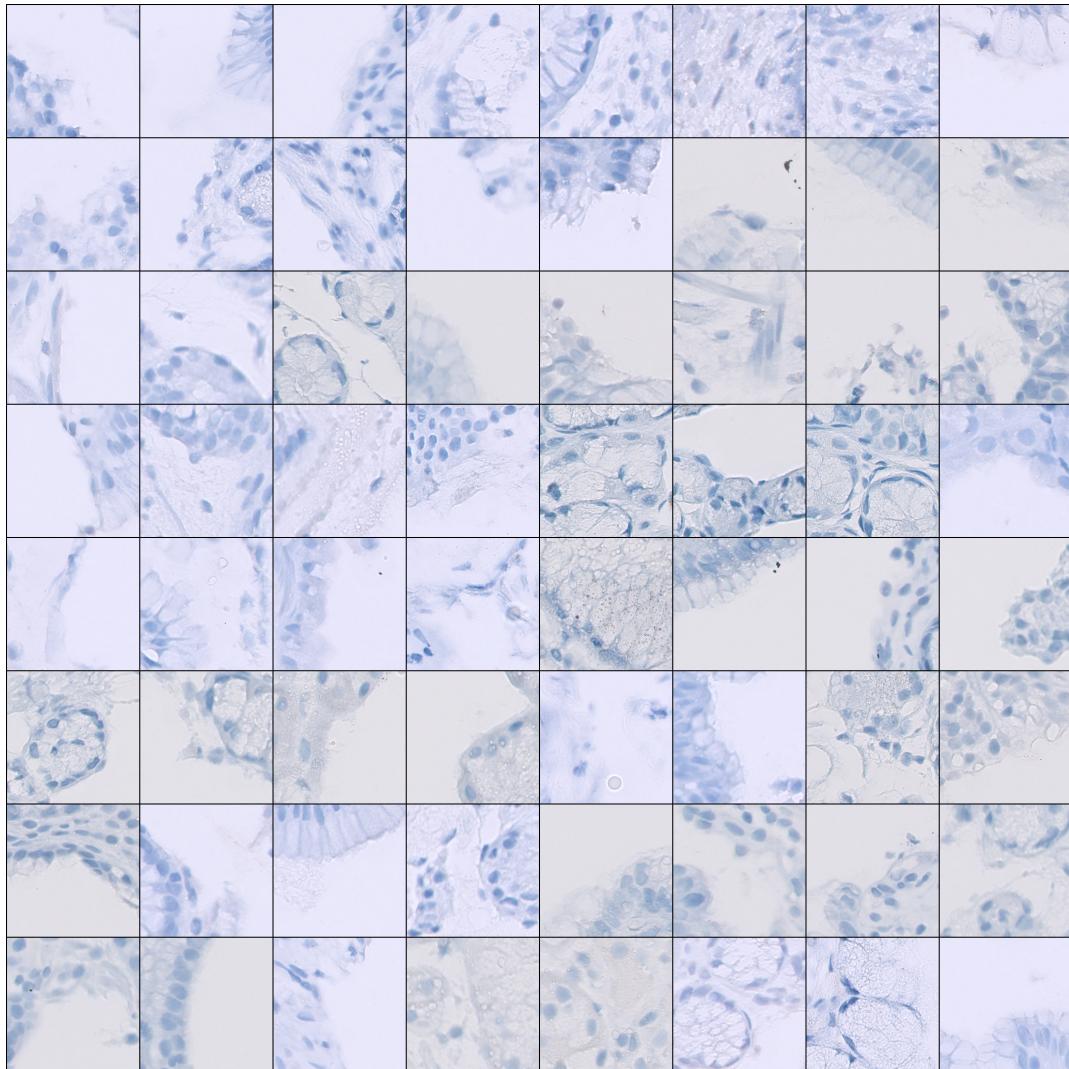


Figura 1: Patches sans originals (conjunt d'entrenament).

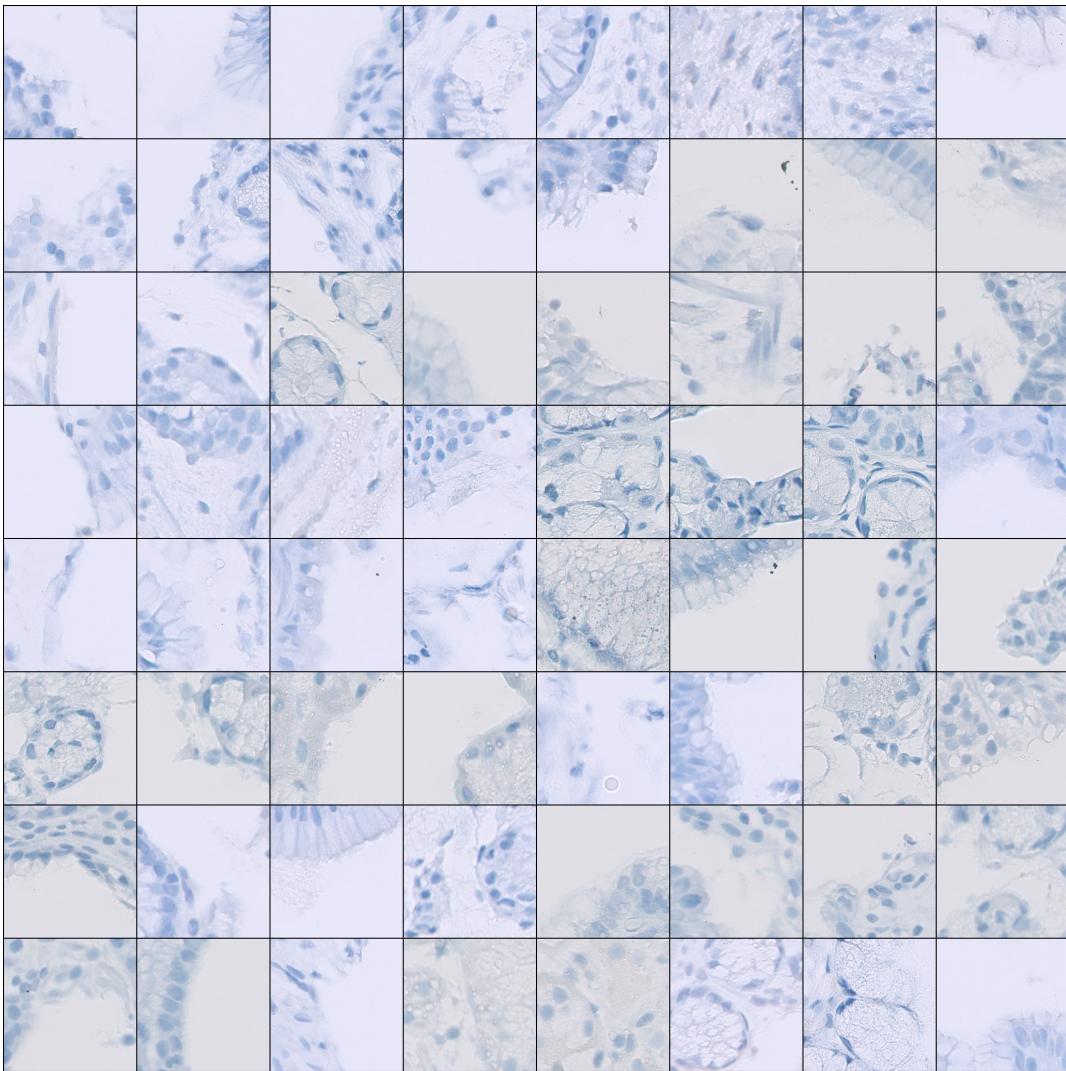


Figura 2: Reconstruccions de patches sans generades per l'autoencoder.

4.2 Reconstrucció de patches no sans

Quan el model processa *patches* de pacients positius (mostres amb *H. pylori*), tendeix a “esborrar” o suavitzar les zones marronoses característiques. Això genera un error de reconstrucció notablement més alt i indica que el model utilitza el patró de color com a senyal d’anomalia.

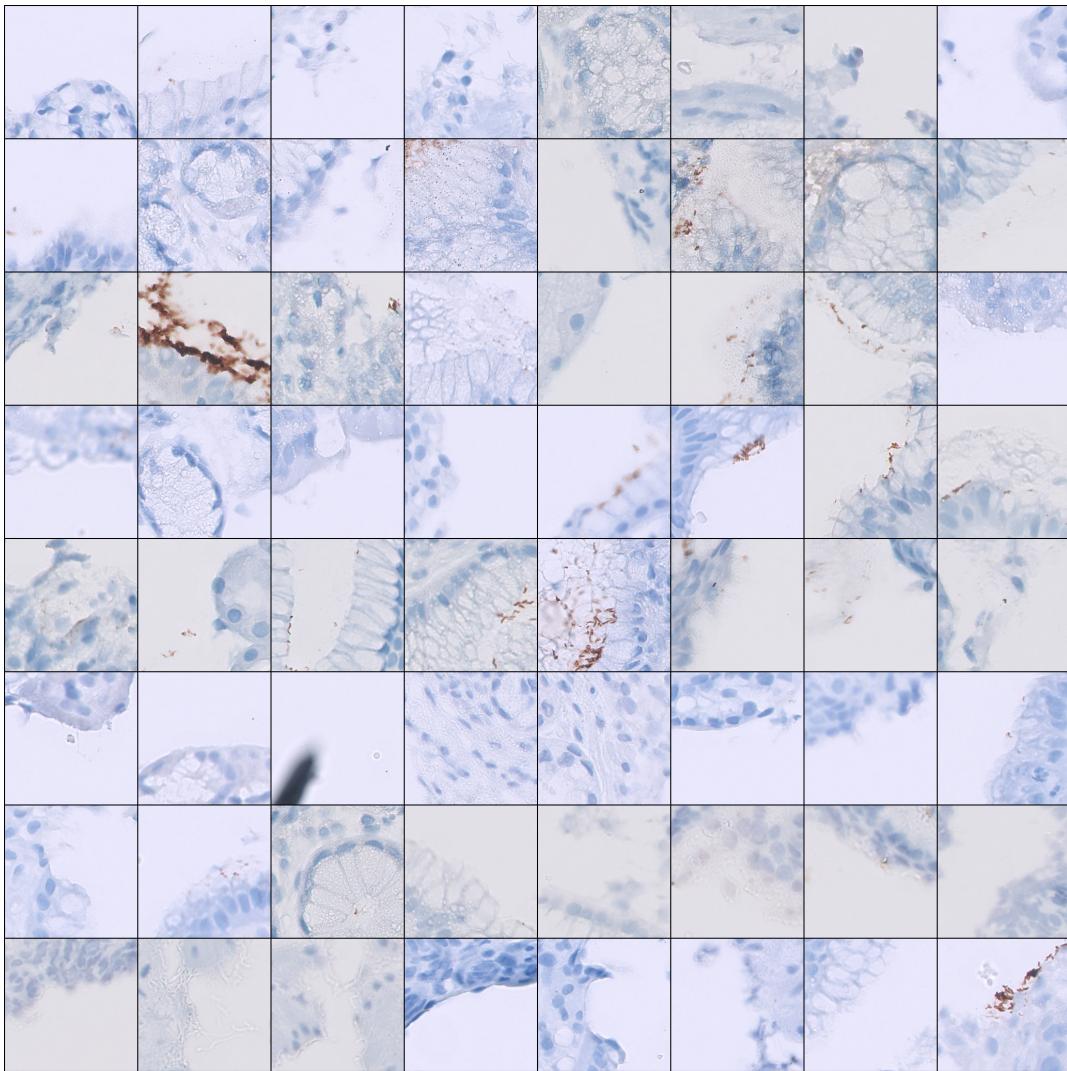


Figura 3: Patches no sans originals (presència d'*H. pylori*).

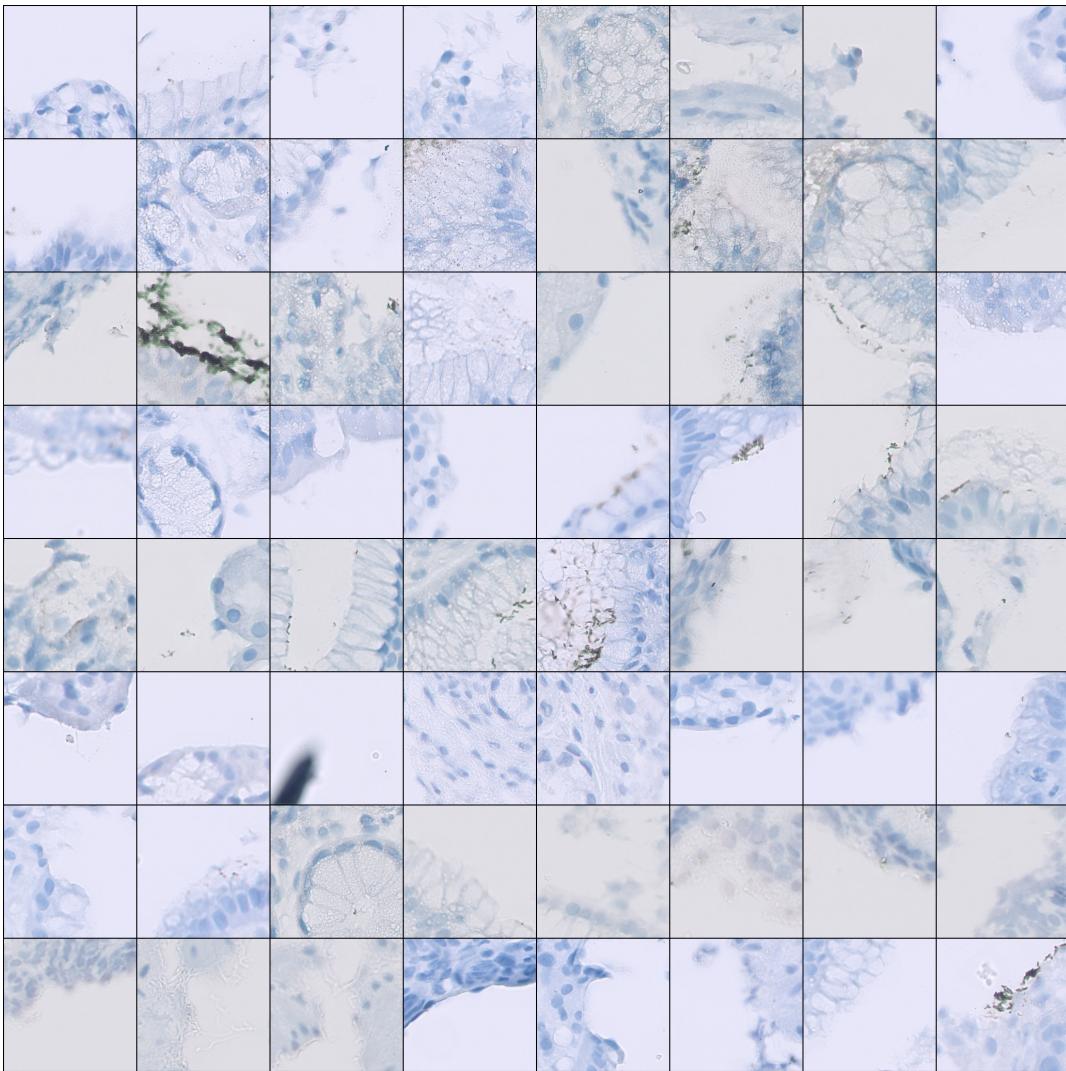


Figura 4: Reconstruccions de patches no sans. Les zones marronoses no es reconstrueixen correctament.

4.3 Avaluació a nivell de patch

Fem validació creuada estratificada per pacients (10 folds). Cada fold calcula el llindar òptim τ a partir del conjunt de *train*, utilitzant la mètrica *Hue Red Weighted*. La Figura 5 mostra la corba ROC agregada.

S'obté:

- AUC mitjà: **0.963 ± 0.032**
- Llindar global aproximat: $\tau = 0.174$
- Comportament estable entre folds.

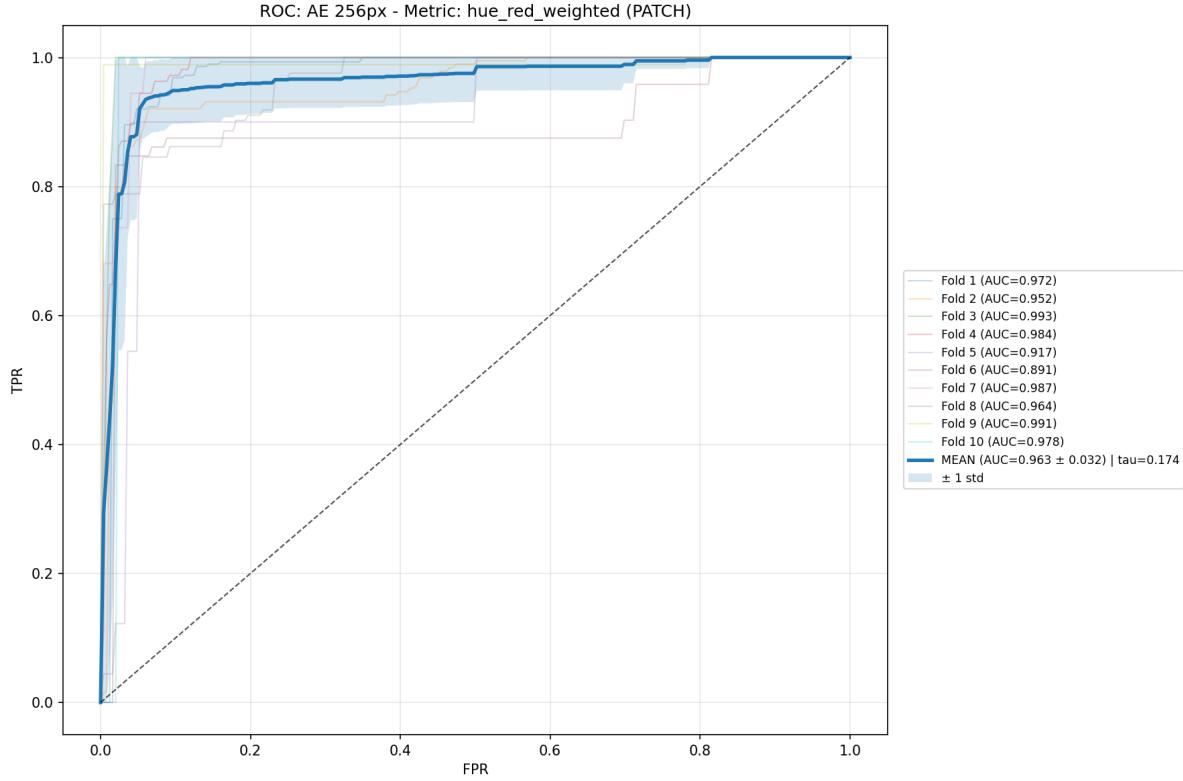


Figura 5: ROC a nivell de *patch* utilitzant la mètrica *Hue Red Weighted*.

4.4 Avaluació a nivell de pacient

Agreguem els errors dels *patches* d'un mateix pacient mitjançant el percentil 95 (p95). Això permet detectar pacients positius encara que només hi hagi uns pocs *patches* anòmals. La Figura 6 mostra les ROC per pacient.

Resultats:

- AUC mitjà: **0.977 ± 0.037**
- Molts folds amb AUC ≈ 1.0
- Llindar global per pacient: $\tau \approx 0.4257$

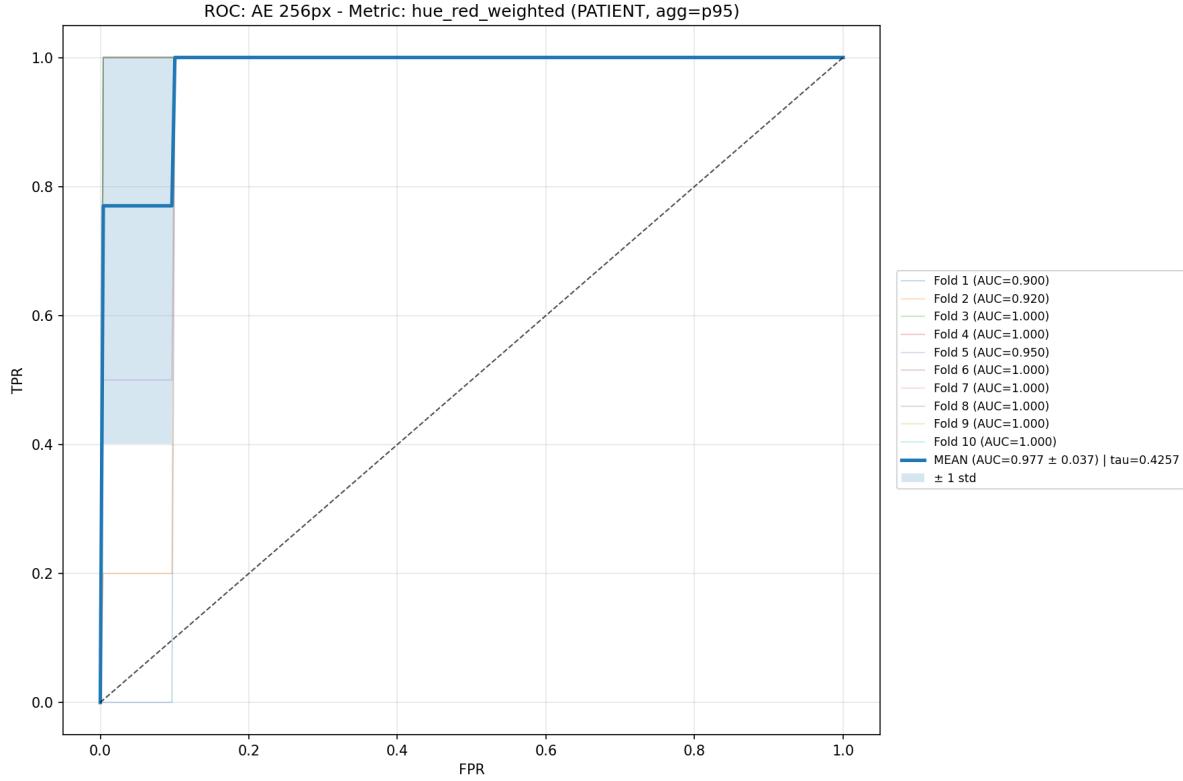


Figura 6: ROC a nivell de pacient utilitzant agregació p95.

5 System 2: MIL amb mòdul d'Attention

El segon sistema aborda la classificació de pacients mitjançant un enfocament de supervisió feble. En lloc d'etiquetar cada *patch* individual, utilitzem un model basat en *Multiple Instance Learning* (MIL) amb un mòdul d'*attention*, que permet agrupar tots els *patches* d'un pacient en una única predicció global.

Cada pacient es representa com un conjunt (*bag*) de *patches*. Primer extraiem *embeddings* convolucionals i després els passem pel mòdul MIL d'*attention*, que assigna un pes rellevant a cada *patch*. Els *patches* més informatius (per exemple, aquells amb regions marronoses compatibles amb *H. pylori*) reben més pes en el càlcul final.

5.1 Arquitectura general

L'arquitectura del System 2 segueix tres passos:

1. **Extracció d'embeddings:** cada *patch* es passa per un extractor convolucional (ResNet18 reduïda), obtenint vectors de característica.
2. **Mòdul d'Attention-MIL:** combina els *embeddings* ponderant-los segons la seva importància.
3. **Classificador final:** rep el vector agragat i retorna la probabilitat que el pacient sigui positiu.

Aquest tipus d'aproximació és especialment útil perquè permet treballar amb WSI sense necessitar anotacions exhaustives, i deixa que el model aprengui quins *patches* són útils.

5.2 Validació creuada per patient

Igual que al System 1, utilitzem una validació creuada estratificada de **10 folds**, sempre partint pacients sencers per evitar *data leakage*. A cada fold entrenem el model complet i calculem corbes ROC/AUC sobre pacients no vistos. Els resultats mostren un rendiment molt alt i estable.

5.3 Matriu de confusió global

La Figura 7 mostra la matriu de confusió agregada utilitzant un llindar de 0.5 per a la classificació final.

- **Sensibilitat alta:** el model detecta bé els pacients positius.
- **Especificitat alta:** també s'equivoquen molt pocs negatius.

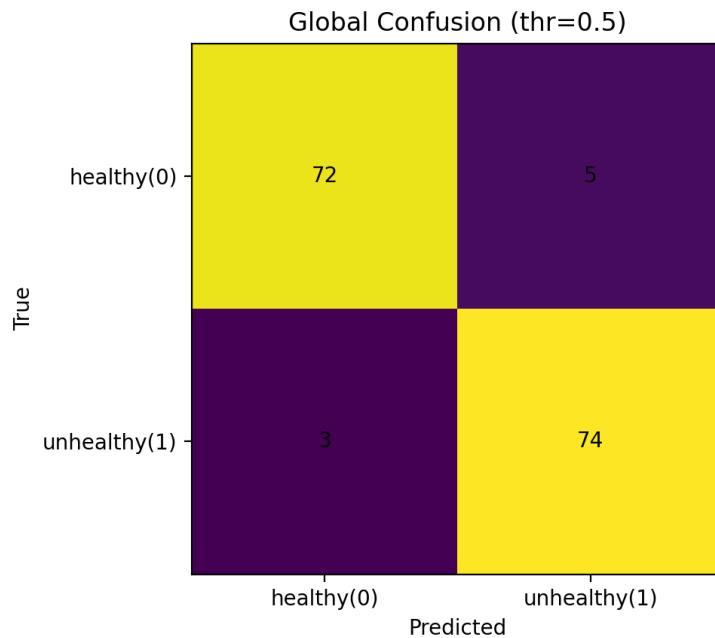


Figura 7: Matriu de confusió global del System 2 utilitzant $thr = 0.5$.

5.4 Corbes ROC per fold

La Figura 8 mostra les corbes ROC obtingudes en cada fold i la mitjana. El model és molt estable, amb una AUC mitjana propera a 1.0.

- AUC mitjana: **0.990 \pm 0.008**
- Diversos folds amb AUC ≈ 1.0
- Variabilitat molt baixa entre particions.

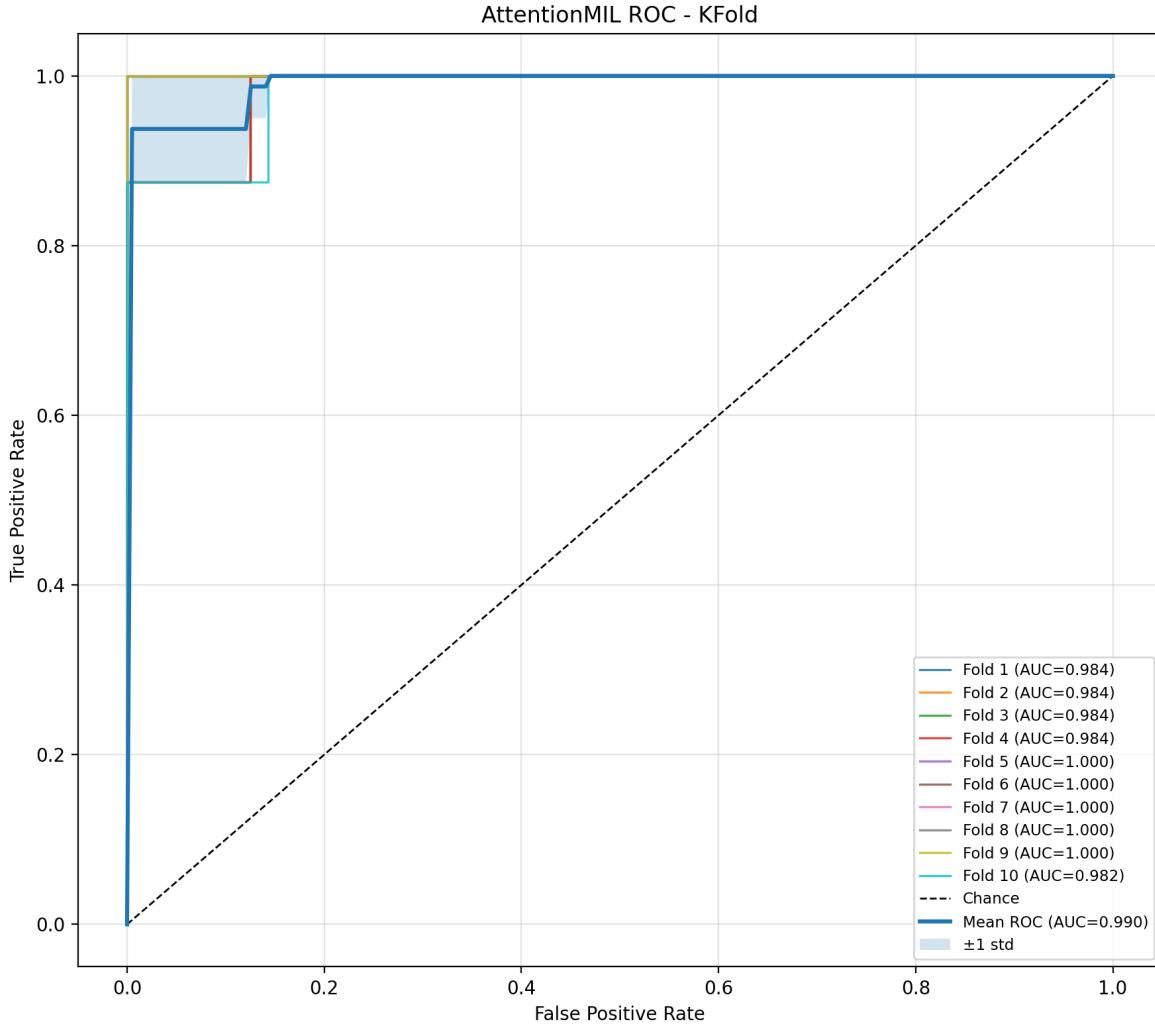


Figura 8: Corbes ROC per fold del System 2 (MIL + Attention).

5.5 Discussió del System 2

El System 2 aconsegueix un rendiment excel·lent, superior al System 1, principalment perquè:

- Treballa directament amb supervisió de pacient, aprofitant més informació.
- El mòdul d'*attention* identifica automàticament els *patches* rellevants.
- Tolera bé la variabilitat dins d'un mateix pacient (*patches* sans + *patches* positius).

L'estabilitat entre folds demostra que el model generalitza bé i que no depèn d'un conjunt concret de pacients. En conjunt, el System 2 és un classificador robust i pràctic per detectar la presència d'*H. pylori* en mostres histopatològiques sense necessitat d'anotacions manuals de *patch*.

6 Materials i codi

Per poder reproduir tot el pipeline del projecte —des del preprocessat de *patches* fins a l'entrenament dels dos sistemes i la generació de resultats— hem mantingut tot el codi organitzat en un repositori

públic de GitHub:

https://github.com/Naanaa17/Helicobacter_Detection

Al repositori s'hi pot trobar:

- **Estructura de carpetes** per deixar clar què és cada part (entrenament, MIL, autoencoder, scripts de Slurm, resultats, etc.).
- **Scripts d'entrenament** del System 1 (autoencoder) i System 2 (MIL + Attention).
- **Codi de preprocessat** per carregar WSI, generar *patches*, normalitzar i construir els *datasets* de PyTorch.
- **Implementació de K-Fold** estratificat per pacient, utilitzat per evitar *data leakage*.
- **Generadors de mètriques i figures**, incloent corbes ROC, matrius de confusió i visors de reconstruccions.
- **Fitxers README** amb instruccions per executar tot el pipeline tant localment com al clúster de la UAB amb **Slurm**.

Tot el codi s'ha desenvolupat en Python (PyTorch), i està pensat perquè sigui fàcil de modificar per fer noves proves o ampliar el projecte. A més, totes les comandes d'execució utilitzades durant l'entrenament (tant en local com al clúster) estan documentades al repositori.

7 Conclusions i treball futur

En aquest projecte hem implementat dos enfocaments diferents per detectar *H. pylori* en imatges de biòpsies: un autoencoder que aprèn com són els *patches* sans i un model MIL amb *attention* que treballa directament a nivell de pacient. Tot plegat ens ha permès comparar dues maneres de veure el problema: una més basada en “què és normal” i l'altra en “què diu el pacient”.

El **System 1** funciona bé detectant anomalies perquè l'autoencoder és molt bo reconstruint teixit sa. Quan hi ha zones marronoses típiques del bacteri, el model pateix més i això genera errors més grans. Afegir el terme de color (HSV/Hue) ha ajudat bastant, sobretot perquè el vermell és una pista clara en aquest tipus de tinció.

El **System 2**, en canvi, treu molt millor rendiment gràcies al mòdul d'*attention*. El fet que el model pugui “decidir” quins *patches* importen més li dóna una flexibilitat extra, i això es nota tant en les ROC com en la matriu de confusió. A nivell pràctic, seria la solució més útil si s'hagués de portar a un entorn clínic.

Pel que fa a limitacions, hem treballat amb un únic dataset i no hem aplicat normalització de color avançada, cosa que pot afectar si es canvia d'hospital o protocol de tinció. A més, el System 2 és computacionalment més car.

Treball futur

Hi ha diverses coses que es podrien millorar o ampliar:

- Afegir augmentacions específiques de teixit gastroduodenal.
- Testjar autoencoders més moderns (per exemple, amb blocs residuals, skip connections).

- Extendre el problema a classificació multi-nivell (NEGATIVA / BAIXA / ALTA).
- Fer mapes d'atenció o *heatmaps* per entendre millor on es fixa el model.

En general, els dos models funcionen molt bé per al dataset que hem utilitzat, i el pipeline complet implementat en el repositori ens ha servit per treballar totes les parts d'un projecte real de visió per computador aplicat a salut.