UAB
Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

This is the **published version** of the bachelor thesis:

Colillas Ceballos, Joan. *Anàlisi i classificació d'imatges reals i generades per IA mitjançant descriptors estadístics*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Grau en Enginyeria de Dades)

This version is available at https://ddd.uab.cat/record/337337

under the terms of the CC BY-SA license.

Anàlisi i classificació d'imatges reals i generades per IA mitjançant descriptors estadístics

Joan Colillas Ceballos

Resum— En els darrers anys, els models generatius d'intel·ligència artificial han assolit un nivell de realisme que dificulta cada vegada més la distinció entre imatges reals i imatges sintètiques. Aquest treball presenta el desenvolupament d'un sistema lleuger per a l'anàlisi i classificació d'imatges reals i generades per intel·ligència artificial mitjançant descriptors estadístics. A diferència dels enfocaments basats en xarxes neuronals profundes aplicades directament sobre els píxels de la imatge, el sistema transforma cada imatge en un vector de característiques relacionades amb el color, la freqüència, la textura, el soroll residual, l'entropia i la nitidesa. Aquest vector s'utilitza per entrenar i comparar models tabulars de classificació. També s'ha desenvolupat una extensió web capaç d'executar localment l'extracció de característiques i la predicció dins del navegador, amb l'objectiu que el sistema sigui viable en un ordinador convencional. Els resultats mostren que aquest enfocament pot obtenir un bon rendiment en entorns controlats, però també evidencien una sensibilitat important al canvi de domini entre conjunts de dades. Per aquest motiu, el sistema es planteja com una eina de suport a la interpretació, i no com un veredicte absolut sobre l'origen d'una imatge.

Paraules clau— Models generatius, anàlisi freqüencial, descriptors estadístics, classificació d'imatges, models tabulars, extensió web, interpretabilitat, Midjourney, StableDiffusion, DALL·E, models de text a imatge, anàlisi forense d'imatges.

Abstract— In recent years, generative artificial intelligence models have reached a level of realism that makes it increasingly difficult to distinguish between real and synthetic images. This project presents the development of a lightweight system for the analysis and classification of real and AI-generated images using statistical descriptors. Unlike approaches based on deep neural networks applied directly to image pixels, the system transforms each image into a feature vector related to color, frequency, texture, residual noise, entropy and sharpness. This vector is used to train and compare tabular classification models. A web extension has also been developed to execute both feature extraction and prediction locally within the browser, with the aim of making the system viable on a conventional computer. The results show that this approach can achieve good performance in controlled environments, but they also reveal a significant sensitivity to domain shift between datasets. For this reason, the system is proposed as an interpretative support tool rather than an absolute verdict on the origin of an image.

Terms— Generative models, frequency analysis, statistical descriptors, image classification, tabular models, web extension, interpretability, Midjourney, Stable Diffusion, DALL·E, text-to-image models, image forensics.

1 Introducció - Context del treball

En els darrers anys, la generació d'imatges mitjançant intel·ligència artificial ha experimentat una evolució molt significativa. Models com Stable Diffusion[1], Midjourney[2] o DALL·E[3] han demostrat la capacitat de crear contingut visual cada vegada més realista a partir de simples instruccions textuals. Aquesta evolució ha fet que la frontera entre les imatges reals i les imatges generades artificialment sigui progressivament més difícil d'identificar, tant per als usuaris com per als sistemes automàtics de detecció.

La manipulació d'imatges no és una tècnica nova. Des dels inicis de la fotografia ja existien tècniques per alterar o modificar contingut visual, i amb l'arribada de l'edició digital aquestes pràctiques es van fer molt més accessibles. Així doncs, la intel·ligència artificial generativa introdueix un canvi important: ja no es tracta només de modificar una fotografia existent, sinó de sintetitzar imatges completament noves amb un nivell de realisme elevat. Aquest fet planteja reptes relacionats amb la confiança en el contingut digital, la desinformació i la capacitat crítica dels usuaris davant les imatges que consumeixen en entorns digitals.

En aquest context, la detecció d'imatges generades per IA continua sent un problema obert. Moltes aproximacions actuals es basen en models d'aprenentatge profund entrenats directament sobre els píxels de la imatge. Tot i que aquests enfocaments poden obtenir bons resultats en entorns controlats, també poden requerir grans volums de dades, un cost computacional elevat i una execució poc adequada per a dispositius convencionals. A més, sovint ofereixen una classificació final sense explicar de manera clara quins indicis han influït en la decisió.

La identificació d'imatges generades per IA es pot abordar des de diferents tipus d'indicis. Alguns són perceptibles, com errors anatòmics, textos deformats, perspectives incoherents, ombres poc plausibles o elements visuals que resulten estranys per a l'observador. Altres indicis, en

- *E-mail de contacte: joan.colillas@autonoma.cat*
- *Treball tutoritzat per: Alexandra Gomez Villa (Àrea de Ciències de la Computació i Intel·ligència Artificial)*
- *Curs 2025/26*

canvi, no són evidents a simple vista i requereixen tècniques d'anàlisi computacional per ser detectats, com ara diferències en la distribució de freqüències, el soroll residual, les textures locals o altres propietats estadístiques de la imatge [4]. Tot i que els indicis perceptibles i semàntics són rellevants per entendre la dificultat del problema, no han estat l'objecte principal de desenvolupament d'aquest treball.

Aquest treball parteix d'una aproximació alternativa basada en descriptors estadístics i models tabulars. En lloc de processar directament la imatge amb una xarxa neuronal profunda, s'extreuen uns "features" que es guarden en un vector de característiques relacionades amb propietats com el color, la freqüència, la textura, el soroll residual, l'entropia o la nitidesa. Aquest plantejament permet explorar una solució més lleugera i interpretable, orientada a una possible execució local dins d'una extensió de navegador.

La hipòtesi que es planteja és que es possible detectar o tenir indicis que una imatge està generada per intel·ligència artificial mitjançant descriptors estadístics derivats de qualsevol imatge a analitzar.

Així doncs, el treball se situa en la intersecció entre l'anàlisi d'imatge, l'aprenentatge automàtic i el desenvolupament d'una eina web lleugera. La proposta no pretén oferir una predicció absoluta sobre l'origen d'una imatge, sinó estudiar fins a quin punt un conjunt de descriptors computacionals pot aportar informació útil per diferenciar imatges reals i generades per intel·ligència artificial.

2 Objectius

Aquest treball té com a finalitat principal desenvolupar un sistema capaç d'analitzar imatges i estimar si el seu origen és real o generat per intel·ligència artificial. A partir d'aquesta finalitat general, es defineixen els següents objectius específics:

- Estudiar descriptors computacionals per a l'anàlisi d'imatges.
- Investigar diferents famílies de característiques que puguin aportar informació útil per diferenciar imatges reals i generades per IA.
- Construir un pipeline robust d'extracció de característiques.
- Desenvolupar un procés capaç de transformar cada imatge en un vector numèric de característiques. Aquest vector ha de representar diferents propietats de la imatge i, alhora, mantenir un cost computacional baix.
- A partir dels vectors de característiques extrets, entrenar diferents models tabulars per tal que puguin donar una classificació per tal de classificar les imatges com a reals o generades per intel·ligència artificial.
- Desenvolupar una extensió web funcional, que sigui capaç de carregar imatges de la pàgina a la que s'estigui accedint, i donar informació sobre els features i transformacions que s'estàn utilitzar per a obtenir la predicció.
- Avaluar el sistema en diferents conjunts de dades.

3 Estat de l'art

3.1 Detecció d'imatges generades per intel·ligència artificial

Diversos estudis han analitzat fins a quin punt les persones són capaces de diferenciar imatges reals i generades. Per exemple, el treball Seeing is not always believing [5] mostra que els participants humans presenten dificultats importants en aquesta tasca, amb una taxa d'error considerable en la classificació d'imatges reals i sintètiques.

Altres estudis també apunten que les imatges generades per IA sovint són classificades erròniament com a humanes o reals [6], especialment quan no presenten errors visuals evidents.

Tradicionalment, els indicis utilitzats per les persones per identificar imatges generades acostumen a estar relacionats amb errors perceptibles: anatomies incorrectes (com tindre 6 dits en una mà), textos deformats, incoherències físiques, ombres poc plausibles, perspectives estranyes o elements de l'escena que no tenen sentit. Tot i això, aquests indicis no sempre són presents, i els models generatius actuals han reduït molts dels errors més evidents.

3.2 Enfocaments basats en aprenentatge profund

Una part important de la recerca actual en detecció d'imatges generades per IA es basa en models d'aprenentatge profund. Aquests enfocaments solen entrenar classificadors directament sobre els píxels de la imatge o sobre representacions apreses automàticament mitjançant xarxes neuronals convolucionals, arquitectures basades en transformers o models multimodals. En molts casos, aquests sistemes poden obtenir resultats elevats quan s'entrenen i s'avaluen sobre conjunts de dades similars.

No obstant això, aquests models també presenten limitacions. En primer lloc, acostumen a requerir grans volums de dades i recursos computacionals elevats, tant durant l'entrenament com, en alguns casos, durant la inferència. En segon lloc, sovint funcionen com a sistemes de caixa negra, ja que proporcionen una predicció final sense indicar de manera clara quins patrons concrets de la imatge han estat determinants. També, una altra limitació important és la generalització. Diferents treballs han mostrat que els detectors poden tenir un rendiment alt en imatges generades per models coneguts, però perdre robustesa quan s'avaluen sobre imatges procedents de generadors, resolucions, compressions o fonts diferents [7]. Aquest fenomen, conegut com a canvi de domini, és especialment problemàtic en la detecció d'imatges generades per IA, ja que els models generatius evolucionen ràpidament i les imatges que circulen a internet poden haver estat reescalades, comprimides o editades.

3.3 Descriptors computacionals i anàlisi forense d'imatge

Com a alternativa o complement als models profunds, diversos treballs han explorat l'ús de descriptors computacionals per detectar diferències estadístiques entre imatges reals i generades. Aquest enfocament parteix de la idea que

una imatge generada artificialment pot no seguir exactament les mateixes regularitats que una fotografia capturada per una càmera. Encara que visualment la imatge sembli realista, poden aparèixer diferències en la distribució de freqüències, el soroll residual, les textures locals, la nitidesa o les relacions entre píxels.

Un altre enfocament relacionat és el *watermarking* d'imatges generades per IA. A diferència dels detectors basats en trobar patrons estadístics o artefactes visuals, aquesta tècnica consisteix a inserir una marca d'aigua imperceptible dins la imatge, ja sigui durant el procés de generació o posteriorment. Aquesta marca pot ser detectada més endavant per verificar l'origen sintètic de la imatge o facilitar-ne la traçabilitat. Per tant, el *watermarking* no intenta deduir si una imatge és generada a partir dels seus indicis visuals, sinó que introdueix un senyal verificable de procedència. Tot i això, aquest enfocament també presenta reptes importants, especialment pel que fa a la robustesa davant transformacions com compressió, retallades, reescalat o possibles atacs destinats a eliminar o falsificar la marca [8].

L'anàlisi freqüencial és una de les línies més rellevants en aquest àmbit. Alguns treballs han observat que les imatges generades per xarxes generatives poden presentar artefactes en el domini de la freqüència, sovint relacionats amb operacions d'upsampling o amb la manera com el model sintetitza la imatge [9] [10]. Aquest tipus d'anàlisi és interessant perquè permet detectar patrons que no sempre són visibles en el domini espacial. En aquest sentit, la transformada de Fourier i altres representacions freqüencials poden ajudar a caracteritzar diferències entre fotografies reals i imatges generades.

3.4 Eines existents per a la detecció d'imatges generades

Actualment existeixen diverses eines comercials i extensions que permeten estimar si un contingut ha estat generat per IA. Algunes d'aquestes eines funcionen directament des del navegador i permeten analitzar imatges mentre l'usuari navega. Per exemple, algunes extensions permeten escanejar imatges mitjançant un menú contextual o una interfície pròpia, i altres serveis ofereixen detecció multimodal sobre imatges, text, àudio o vídeo [11] [12].



Fig 1: Exemple d'interfície d'una extensió existent de detecció d'imatges IA [12].

Aquestes eines mostren que hi ha una demanda creixent de solucions accessibles per verificar contingut digital. No obstant això, moltes d'aquestes plataformes funcionen com a serveis tancats: no sempre indiquen amb detall quin tipus de model utilitzen, si la inferència es realitza localment o en un servidor extern, ni quins indicadors concrets han portat a la classificació final. Això limita la transparència del procés i dificulta que l'usuari pugui interpretar el resultat més enllà d'una puntuació o etiqueta.

4 MÈTODES I DESCRIPTORS PRINCIPALS

Aquest apartat presenta les principals famílies de descriptors utilitzades durant el desenvolupament del sistema. La selecció d'aquests mètodes respon a tres criteris principals: que poguessin capturar indicis no perceptibles de la imatge, que fossin interpretables i que tinguessin un cost computacional baix, i que part tant fossin compatibles amb una execució local dins del navegador.[13]

Durant les primeres fases del projecte es van explorar diferents mètodes d'anàlisi d'imatge. No obstant, més tard alguns es van descartar o simplificar per reduir redundàncies, evitar descriptors poc informatius i garantir una implementació fiable en JavaScript. Per aquest motiu, en aquest apartat es resumeixen les famílies de descriptors més importants en el desenvolupament de l'eina.

4.1 Descriptors estadístics i de color

Els descriptors de color resumeixen la distribució cromàtica de la imatge. Per a cada canal RGB o HSV es calculen estadístics com la mitjana, la desviació estàndard, l'asimetria o l'entropia. També es calculen correlacions entre canals, que poden indicar fins a quin punt dues components de color varien conjuntament.

De manera general, si (X_c) representa els valors d'un canal de color (c), la mitjana i la desviació estàndard es poden expressar com:

$$\mu_c = \frac{1}{N}\sum_{i=1}^{N} X_{c,i} \qquad \sigma_c = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_{c,i} - \mu_c\right)^2}$$

Aquests descriptors permeten capturar diferències en intensitat, saturació, brillantor o equilibri cromàtic entre imatges reals i generades.

4.2 Descriptors freqüencials i soroll residual

L'anàlisi freqüencial permet estudiar com es distribueix l'energia de la imatge entre freqüències baixes, mitjanes i altes. Aquesta informació és rellevant perquè algunes imatges generades poden presentar patrons espectrals diferents dels d'una fotografia real, especialment per efectes associats al procés de generació o de reconstrucció de la imatge [14].

En aquest treball, a partir de l'espectre freqüencial es calculen mesures com l'energia total i la proporció d'energia en diferents bandes. Per a una banda freqüencial (B), aquesta proporció es pot expressar com:

$$R_B = \frac{\sum_{(u,v)\in B} |F(u,v)|^2}{E_{total}}$$

On $F(u,v)$ representa l'espectre de Fourier de la imatge i E_{total} l'energia total del domini freqüencial.

A més de l'anàlisi directa de la imatge, també s'utilitzen descriptors de soroll residual. El residual intenta

ressaltar els detalls d'alta freqüència i reduir el pes de la semàntica de l'escena [15]. Es pot entendre com la diferència entre la imatge original i una versió suavitzada:

$$R(x,y) = I(x,y) - (G_\sigma * I)(x,y)$$

On $I(x,y)$ representa la intensitat de la imatge original en la posició (x,y), G_σ és un filtre gaussià amb desviació estàndard σ, i (*) indica l'operació de convolució. D'aquesta manera, $((G_\sigma * I)(x,y))$correspon a una versió suavitzada de la imatge, mentre que $R(x,y)$ conserva principalment les diferències locals, el soroll i els detalls d'alta freqüència.

Sobre aquest residual es calculen estadístics com la desviació, l'entropia, l'asimetria, la curtosi i la distribució freqüencial del mateix residual. Aquesta família és important perquè una fotografia real conté soroll i patrons associats al procés físic de captura, mentre que una imatge generada pot presentar estructures residuals diferents.

4.3 Textura local, gradients i vores

Els descriptors de textura permeten analitzar patrons locals de la imatge. En aquest treball s'han utilitzat tècniques com els Local Binary Patterns (LBP) i la Gray-Level Co-occurrence Matrix (GLCM). Els LBP codifiquen microtextures comparant cada píxel amb els seus veïns, mentre que la GLCM descriu la relació estadística entre nivells de gris propers [16]. Aquestes tècniques poden detectar si una imatge presenta textures massa uniformes, massa suaus o amb patrons locals poc naturals.

També s'han incorporat descriptors basats en gradients i vores. En concret, s'han calculat la mitjana de la magnitud del gradient, la desviació estàndard d'aquesta magnitud i la densitat de vores . Aquests descriptors mesuren els canvis locals d'intensitat i permeten capturar contorns, detalls fins i possibles diferències de nitidesa. En imatges generades, algunes zones poden presentar vores molt netes però amb poc detall intern, o bé textures més uniformes del que seria habitual en una fotografia real.

4.4 Entropia, wavelets i estadístiques d'escenes naturals

L'entropia mesura la variabilitat o el desordre de la distribució d'intensitats. En aquest treball s'utilitza tant de manera global com local, per capturar la complexitat general de la imatge i la variabilitat en petites regions.

També s'han considerat descriptors basats en wavelets, que permeten analitzar la imatge en diferents escales i orientacions. La descomposició en subbandes permet capturar detalls horitzontals, verticals i diagonals, cosa que pot ser útil quan els artefactes de generació no apareixen de manera uniforme en tota la imatge.

Finalment, s'han utilitzat estadístiques d'escenes naturals mitjançant coeficients MSCN [17]. Aquests coeficients normalitzen cada píxel respecte al seu entorn local:

$$\hat{I}(x,y) = \frac{I(x,y) - \mu(x,y)}{\sigma(x,y) + C}$$

On $\mu(x,y)$i $\sigma(x,y)$ són la mitjana i la desviació estàndard locals, i C és una constant petita per evitar divisions per zero. Aquest tipus de descriptor permet analitzar fins a quin punt una imatge segueix regularitats habituals de les fotografies naturals.

5 METODOLOGIA I DISSENY EXPERIMENTAL

En aquest apartat, es descriuran els experiments realitzats, les dades amb les que s'ha treballat, el disseny final de l'extensió i els resultats obtinguts.

5.1 Datasets

Per al desenvolupament i l'avaluació del sistema s'han utilitzat quatre conjunts de dades diferents. Aquesta decisió respon a la necessitat de no limitar l'estudi a un únic origen d'imatges, ja que la detecció d'imatges generades per IA és un problema molt sensible al domini del dataset. Factors com el model generatiu utilitzat, la font de les imatges reals, la compressió, la resolució, el tipus de contingut visual o el procés de preprocessament poden tenir un gran impacte al comportament dels descriptors i, per tant, al rendiment dels models.

5.1.1 Defactify

Defactify Image Dataset [18] és el conjunt de dades principal utilitzat en aquest treball. Es tracta d'un dataset format per imatges reals i imatges sintètiques generades a partir de diversos models generatius. Les imatges reals provenen de MS COCO[19], mentre que les imatges generades inclouen mostres produïdes per models com Stable Diffusion 2.1 [1], SDXL [20], Stable Diffusion 3 [21], DALL·E 3 [3] i Midjourney v6 [2], i en trobem 16.000 imatges en cada classe.

Tot i que el dataset complet conté un volum molt elevat d'imatges, en aquest projecte s'ha treballat amb un subconjunt de 16.000 imatges, combinant tant imatges reals com artificials (8.000 imatges IA i 8.000 reals).

5.1.2 AI vs. Human-Generated Images

El segon dataset utilitzat és l'AI vs Human Generated Dataset [22] disponible a Kaggle. En aquest projecte, aquest dataset s'ha utilitzat principalment per estudiar la generalització del sistema i per analitzar el comportament dels descriptors en un domini diferent del de Defactify.

Aquest dataset és especialment rellevant perquè presenta una distribució diferent de la del conjunt principal. Les imatges reals i generades provenen d'un altre procés de recopilació i generació, fet que permet comprovar si els patrons apresos pel model són realment generals o si depenen massa del dataset d'entrenament.

En aquest etapa del projecte, es va descarregar el dataset complet de 80.000 imatges.

5.1.3 AI vs. Real Images dataset

El tercer dataset utilitzat és l'AI vs Real Images Dataset [23] disponible a Kaggle. Aquest conjunt inclou diverses categories visuals com *animals*, *city*, *food*, *nature* i *people*. Aquesta organització per categories ha estat especialment útil per analitzar si el rendiment del sistema és homogeni

o si depèn del tipus de contingut visual de la imatge.

Aquest no s'ha utilitzat com a conjunt principal d'entrenament, sinó com a dataset extern d'avaluació.

En aquest cas també es va descarregar el dataset sencer, el qual es composava d'aproximdament 950 imatges repartides en 5 classes: *Animals, City, Food, Nature* i *People* (150 imatges reals i 50 generades per IA en cada clase).

En aquest cas, hi havia una diferència significativa entre el nombre de d'imatges reals i d'IA, però per tal d'evitar biaixos o inconsistències a la hora d'avaluar els models amb aquest conjunt, es van limitar les imatges reals en cada classe a 50 per tal d'igualar-les amb les imatges generades per intel·ligència artificial.

5.1.4 Dataset Manual

Finalment, es va construir un dataset manual propi format per aproximadament 100 imatges descarregades d'internet (combinant imatges reals i generades). A diferència dels datasets anteriors, aquest conjunt no prové d'una font única ni segueix necessàriament una distribució controlada, i és precisament per això és especialment útil com a prova externa.

5.2 Pipelines i extractors de característiques

5.2.1 Pipeline inicial d'extracció en python

En una primera fase del treball, l'extracció de característiques es va implementar mitjançant un pipeline en Python. Aquest enfocament va permetre experimentar de manera ràpida amb diferents tècniques d'anàlisi d'imatge i validar si els descriptors estadístics podien contenir informació útil per diferenciar imatges reals i generades per intel·ligència artificial. El procés era el següent:

- **Configuració del dataset i de la finestra de treball**: L'usuari indicava la ruta del dataset, el directori de sortida i la mida de redimensionament de les imatges.
- **Càrrega i preprocessament de les imatges**: El pipeline carregava cada imatge, la convertia a RGB i la redimensionava a la mida de window indicada.
- **Extracció de descriptors (features)**: En aquesta etapa es calculaven els 126 features i es compilaven en un vector de característiques per després obtenir les prediccions.
- **Exportació dels resultats**: Per últim, es generaven els fitxers CSV amb les característiques i les etiquetes que posteriorment s'utilitzarien per entrenar els models.

La implementació en Python va resultar pràctica i va permetre obtenir resultats satisfactoris en una fase inicial del projecte. No obstant això, presentava dues limitacions principals:

- **Diferències entre entorns d'extracció**: el càlcul d'alguns descriptors no coincidien exactament amb la implementació en navegador a causa de discrepàncies entre el codi Python i el codi JavaScript. Aquest fet podia provocar que el model s'entrenés amb valors diferents als que rebria en l'extensió.
- **Temps d'execució elevat**: el procés d'extracció tenia un cost computacional considerable.

5.2.2 Pipeline final alineat amb l'extensió en JavaScript

Després de la primera fase d'extracció en Python, es va desenvolupar un segon pipeline orientat a reproduir amb més exactitud el comportament real de l'extensió web.

No obstant, el fet d'implementar un nou extractor de característiques va suposar un nou repte: traslladar l'extracció de descriptors a un entorn JavaScript. Alguns dels mètodes utilitzats en la versió inicial en Python depenien de llibreries especialitzades o de càlculs que eren difícils de reproduir amb seguretat i consistència dins del navegador. Per aquest motiu, no es va fer una traducció directa de tots els descriptors del primer pipeline, sinó una selecció més compacta i controlada.

Aquesta reducció del nombre de característiques va respondre principalment a dos criteris. D'una banda, es van eliminar descriptors redundants o poc informatius, ja que alguns aportaven informació molt similar entre si o podien afegir soroll al model sense millorar-ne el rendiment. D'altra banda, es van descartar o simplificar alguns mètodes que eren massa complexos d'implementar de manera fiable en JavaScript, especialment aquells que requerien operacions molt específiques de processament d'imatge o dependien de llibreries pròpies de Python.

Com a resultat, el segon pipeline va passar a treballar amb un conjunt més reduït de descriptors, mantenint les famílies de característiques més rellevants per al projecte: color, freqüència, soroll residual, textura, entropia, gradients, wavelets, MSCN/NSS i nitidesa. En concret, es van pasar de uns 126 features a 73. Aquesta decisió va permetre obtenir un extractor més lleuger, més ràpid i més fàcil d'integrar dins de l'extensió. Finalment, els passos del pipeline definitiu són els següents:

- **Preparació prèvia de les imatges**: Les imatges del dataset es llegien dels directoris corresponents i es convertien a PNG. Aquesta conversió facilitava que el navegador rebés les imatges en un format controlat i reduïa possibles diferències derivades del format original.
- **Lectura i codificació de les imatges**: Un script en node.js llegia cada imatge i les seves metadades, com la ruta, el nom del fitxer, la classe i, si era necessari, la categoria.
- **Execució en un navegador controlat**: El pipeline utilitzava la llibreria de Puppeteer per obrir Chromium en mode headless, és a dir, sense interfície gràfica visible. Aquest entorn permetia executar codi JavaScript en condicions molt semblants a les d'un navegador real i així realitzar proves.
- **Càrrega del mateix extractor que utilitza l'extensió**: Dins d'aquest entorn de navegador es carregava el fitxer feature_extractor.js, que és el mateix mòdul utilitzat per l'extensió final. Això assegurava que el redimensionament, les transformacions i el càlcul dels descriptors seguissin el mateix procediment durant l'entrenament i durant la inferència.

- **Extracció de descriptors i generació del CSV final**: Per cada imatge, el navegador aplicava les transformacions definides a l'extractor i generava el vector final de característiques. En aquesta versió del pipeline es calculaven 73 descriptors numèrics, que posteriorment es guardarien en un fitxer CSV juntament amb l'etiqueta i les metadades bàsiques de cada imatge.

5.2.3 Rendiment en els pipelines

Era important tenir en compte el rendiment dels pipelines a la hora d'extreure i exportar els descriptors numèrics, ja que poden donar una estimació del que es tardarà a extreure els features en l'extensió web final.

No obstant, s'ha de tindre en compte que no es pot fer una comparativa directe, ja que en el pipeline de Python extreia més de 120 features, mentre que en el pipeline JavaScript final se n'extrein 73. En la taula següent es pot veure el temps mitjà que es triga a extreure els descriptors d'una imatge (segons/imatge):

TAULA 1: Temps d'extracció (s) de features per cada imatge en cada pipeline i window .

Pipeline/Window d'imatge	**128x128**	**256x256**
Python *(126 features)*	0.275	1.15
JavaScript *(73 features)*	0.09	0.138

Aquests resultats mostren que l'extractor final implementat en JavaScript no només és més coherent amb l'entorn real de l'extensió, sinó que també presenta un millor rendiment temporal. Per altra banda, també presenta un millor factor d'escalabilitat, ja que, amb el doble de resolució, l'increment ha sigut del x1,53 en el temps d'execució, mentre que amb Python l'increment ha sigut de x4,18.

Aquesta millora és especialment important perquè el sistema final ha de funcionar de manera local dins del navegador i oferir una resposta en un temps raonable per a l'usuari.

5.3 Models de classificació utilitzats

Un cop extretes les característiques de les imatges, el problema es transforma en una tasca de classificació supervisada sobre dades tabulars. Cada imatge queda representada per un vector de descriptors numèrics i una etiqueta binària que indica si és real o generada per intel·ligència artificial. A partir d'aquests vectors es van entrenar i comparar tres models: Random Forest, XGBoost i MLP. Aquesta selecció permetia comparar models basats en arbres amb una alternativa neuronal lleugera, tots ells aplicats sobre els mateixos descriptors.

El Random Forest es va utilitzar com a model principal per la seva robustesa en dades tabulars, la seva capacitat per treballar amb descriptors diversos i la seva facilitat d'exportació a JavaScript, aspecte important per a la integració dins de l'extensió web. XGBoost es va incloure com una alternativa basada en boosting d'arbres, habitualment molt competitiva en problemes tabulars, però més sensible a l'ajust d'hiperparàmetres. Finalment, l'MLP es va utilitzar com una alternativa neuronal lleugera aplicada al vector de descriptors, amb l'objectiu de comprovar si una xarxa simple podia aprofitar relacions no lineals entre característiques sense utilitzar directament els píxels de la imatge.

5.4 Procediment d'entrenament i avaluació

El procés d'entrenament i avaluació del sistema no es va plantejar com un únic experiment tancat, sinó com una evolució progressiva del pipeline. Al llarg del desenvolupament es van anar introduint canvis en la mida d'anàlisi de les imatges, en el conjunt de característiques extretes, en la implementació del pipeline i en els datasets utilitzats. Aquesta evolució va permetre detectar limitacions de les primeres versions i ajustar el sistema final a l'objectiu principal del projecte: obtenir un detector lleuger, interpretable i executable dins d'una extensió de navegador.

5.4.1 Primeres implementacions dels models classificadors

El primer experiment tenia com a objectiu comprovar si els descriptors estadístics permetien diferenciar imatges reals i generades per IA. Per fer-ho, es va utilitzar el pipeline inicial en Python amb el subconjunt de 16.000 imatges de Defactify i es van entrenar les primeres versions de Random Forest, XGBoost i MLP amb els 126 descriptors disponibles.

Inicialment, es va treballar amb imatges redimensionades a 128×128 píxels. Els resultats, recollits a l'apèndix A1.1, ja mostraven que l'enfocament era viable, amb accuracies de test superiors al 84% en els tres models. Posteriorment, es va repetir l'experiment amb una resolució de 256 × 256 píxels per comprovar si d'aquesta manera es podia obtenir millor detall. El resultat va ser una millora generalitzada: el Random Forest va passar del 84,1% al 85,9%, XGBoost del 86,3% al 89,1% i l'MLP del 86,5% al 90,2%. Aquesta millora indicava que una resolució més alta conservava més informació útil per als descriptors, especialment en textura, freqüència, residuals i vores.

5.4.2 Implementació de models compatibles amb el navegador

Després d'observar la viabilitat de la implementació, el següent pas va consistir a adaptar el sistema perquè els models poguessin ser utilitzats dins de l'extensió web. Per fer-ho, es va treballar amb el nou pipeline d'extracció en JavaScript, basat en el mateix feature_extractor.js que utilitza l'extensió. Aquesta decisió permetia que els descriptors calculats durant l'entrenament coincidissin amb els que posteriorment rebria el model durant la inferència en navegador.

Aquest nou pipeline (com s'ha comentat anteriorment) extreia menys features que amb el pipeline anterior (de 126 a 73 features) seleccionats amb l'objectiu de reduir redundàncies, eliminar característiques poc útils i mantenir només aquelles que es podien implementar de manera fiable en JavaScript. Per això, aquesta etapa va ser vital per veure is realment hi hauria una millora o pitjora significativa respecte la anterior implementació.

També, va ser en aquesta fase (i també gràcies al menor

temps d'execució per tal d'extreure els features) que es va provar a entrenar amb un dataset nou complet de 80.000 imatges: l'AI vs Human Generated Dataset [22] disponible a Kaggle, per veure si la mida del dataset presentava alguna correlació amb una millora en l'accuracy.

Els models en l'experiment varen obtenir accuracies internes elevades, superiors al 93% en validació i test, els resultats en datasets externs van ser considerablement pitjors. No obstant, en el dataset manual, les accuracies van baixar fins al 56% en Random Forest i al 58% en XGBoost i MLP, molt per sota dels models entrenats amb Defactify. En el dataset AI vs Real Images, el rendiment també va ser inferior en Random Forest i XGBoost, tot i que la diferència va ser menys extrema. Aquest comportament mostra que entrenar amb un volum més gran d'imatges no garanteix necessàriament una millor generalització, ja que el model pot aprendre patrons específics del dataset d'entrenament.

5.4.3 Combinació de datasets i reducció de descriptors

En aquesta fase es va provar si reduir el nombre de descriptors podia millorar la generalització dels models. L'objectiu era seleccionar característiques que no només fossin útils en un dataset concret, sinó que mantinguessin un comportament similar tant a Defactify com al dataset de Kaggle. Per fer-ho, es va analitzar la distribució de cada descriptor en tots dos conjunts i es van conservar només aquells que mantenien la mateixa direcció entre imatges reals i generades i que presentaven una separació mínima entre classes mitjançant Cohen's d.

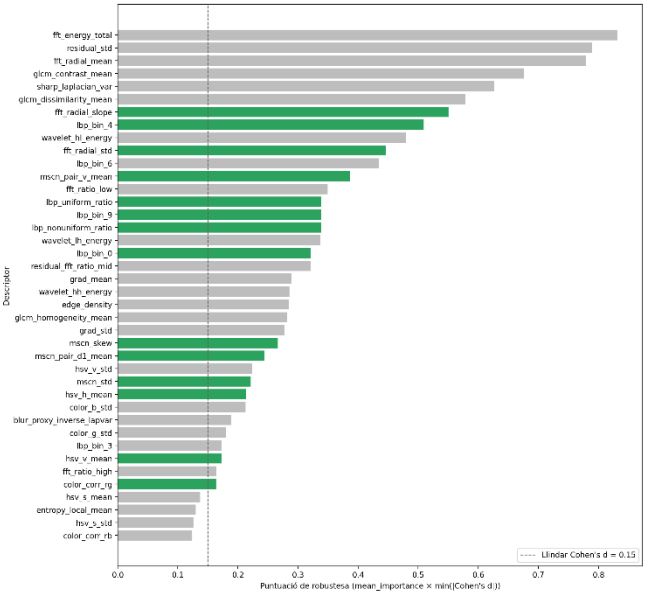


Fig. 2: Rànquing de descriptors per robustesa

Com es mostra a la figura 2, els descriptors es van ordenar segons una puntuació de robustesa que combinava la seva importància mitjana amb la separació mínima entre classes. Tot i això, el fet que un descriptor aparegui més amunt en el rànquing no implica necessàriament que sigui millor candidat per al model reduït. La selecció no depenia només de la puntuació individual, sinó també de la consistència entre datasets i del llindar mínim establert. Per aquest motiu, alguns descriptors amb puntuacions elevades no van ser seleccionats, mentre que altres amb valors més moderats sí que es van mantenir perquè presentaven un comportament més estable entre dominis.

El subconjunt final estava format per 14 descriptors, principalment freqüencials, de textura LBP, estadístiques MSCN/NSS i alguns descriptors de color. Amb aquests descriptors es van generar nous fitxers de característiques i es van tornar a entrenar els models, amb l'objectiu de comprovar si un vector més compacte i consistent entre datasets podia reduir la dependència respecte al domini d'entrenament.

Els resultats van mostrar que la reducció de descriptors disminuïa clarament el rendiment intern dels models. Amb el conjunt complet de descriptors i els datasets combinats, les accuracies de test eren del 88,4% en Random Forest, 90,1% en XGBoost i 93,3% en MLP. En canvi, amb només 14 descriptors, aquests valors baixaven fins al 77,8%, 78,3% i 80,3%, respectivament. Això indica que, tot i seleccionar característiques més consistents entre datasets, la reducció eliminava informació útil per a la classificació.

En els datasets externs, el comportament va ser mixt. En el dataset manual, la reducció va millorar el Random Forest i el XGBoost, passant del 65% al 75% i del 67% al 72%, respectivament. Així doncs, aquesta millora no es va mantenir en el dataset AI vs Real Images, on els models amb el conjunt complet de descriptors van obtenir millors resultats. Per aquest motiu, es va concloure que la reducció de descriptors podia ajudar en alguns escenaris concrets, però no proporcionava una millora generalitzada de la robustesa del sistema.

5.4.5 Model final i optimitzacions

A partir de la comparativa global dels experiments, es va seleccionar com a configuració final el model entrenat amb el dataset Defactify utilitzant el pipeline JavaScript compatible amb navegador i el conjunt complet de 73 descriptors. Aquesta configuració correspon a l'evolució del primer model entrenat amb Defactify, però adaptada al nou extractor de característiques utilitzat també per l'extensió web.

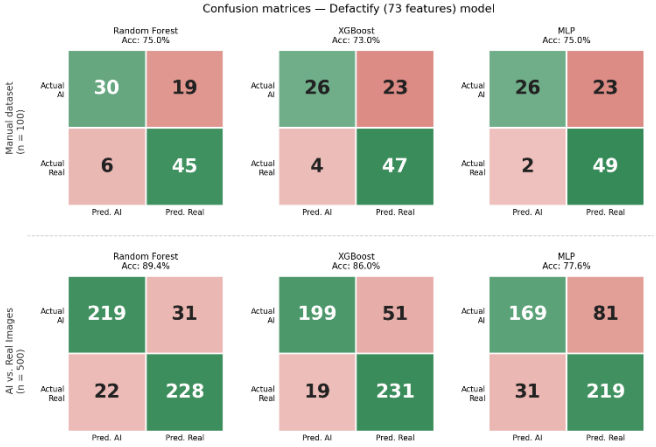


Fig 3: Matrius de confusió dels models de la primera fase

Podem observar que el model és molt més robust a la hora d'identificar imatges Reals que aquelles generades

per intel·ligència artificial, aconseguint un recall entre les dues clases de 61,2% per a la IA i 88,2% per a les Reals. Per això, una de les primeres opcions que es van valorar va ser fer un procés d'optimització d'hiperparàmetres per tal d'analitzar si era possible millorar la capatitat de generalització del sistema, aplicats als tres models: Random Forest, XGBoost i MLP.

No obstant, això plantejava un repte molt gran, i es que en els datasets d'entrenament (Defactify) les accuracies i recalls ja eren molt elevades tot i que en els datasets externs ho fossin menys, per tant, no tindria molt de sentit fer la optimització sobre el dataset original de Defactify, les millores possibles dins del mateix dataset eren petites i, per altra banda, podrien no reflectar un canvi significatiu en un dataset extern. Per aquest motiu, l'optimització d'hiperparàmetres no es va plantejar únicament com una cerca per maximitzar l'accuracy sobre Defactify, sinó com un procés orientat a comprovar si era possible millorar la generalització del sistema.

Després de realitzar l'experiment, els canvis principals van ser els següents: (per mirar els canvis de tots els hiperparàmetres mirar apèndix A2)

En el cas del Random Forest, el canvi principal va ser passar a una configuració amb més capacitat d'aprenentatge. Es va augmentar el nombre d'arbres de 300 a 500 i la profunditat màxima de 14 a 18. També es va reduir el nombre mínim de mostres per fulla de 8 a 1, fet que permet als arbres generar decisions més específiques. D'altra banda, es va reduir max_features de 0,60 a 0,45, afavorint una major diversitat entre arbres. En conjunt, aquests canvis fan que el model sigui menys restrictiu i pugui capturar millor patrons subtils entre descriptors.

Amb el XGBoost, es va passar d'un model amb molts arbres poc profunds a un model amb menys arbres però amb més profunditat. El nombre d'estimadors es va reduir de 1000 a 500, la profunditat màxima va augmentar de 3 a 5 i la taxa d'aprenentatge va baixar de 0,08 a 0,03, aconseugint un aprenentatge més gradual. També es va augmentar el subsample de 0,7 a 0,9 i es va deixar d'utilitzar un scale_pos_weight inferior a 1, que podia reduir massa el pes de la classe IA.

Finalment, amb el MLP els canvis van ser més moderats. La mida de les capes internes va augmentar de (64, 32) a (96, 48), donant més capacitat al model per aprendre relacions no lineals entre descriptors. També es va reduir el batch_size de 256 a 128, fet que pot fer l'entrenament una mica més flexible. La resta de paràmetres principals, com l'activació ReLU, l'optimitzador Adam, l'early stopping i la regularització alpha, es van mantenir pràcticament iguals.



Fig. 4: Matrius de confusió de la segona fase

Després del canvi, el comportament general dels models es va mantenir força similar en el dataset manual, amb petites variacions, però sí que es van veure diferències més clares en el dataset AI vs Real. En general, el Random Forest continuava sent el model més estable: mantenia el mateix resultat en el conjunt manual i només baixava lleugerament en el dataset gran, tot i que es tornava una mica més conservador, detectant menys imatges IA però classificant millor les reals. XGBoost presentava una petita millora en el manual, però empitjorava en el dataset AI vs Real, de manera que el canvi no semblava aportar una millora global clara. En canvi, el MLP va ser el model que més va canviar: millorava tant en el manual com en el dataset AI vs Real, sobretot perquè detecta moltes més imatges generades, però això també provoca un augment considerable dels falsos positius sobre imatges reals.

Un cop finalitzada aquesta fase, es va valorar també l'opció de realitzar un *threshold sweep* per intentar millorar el comportament.

Aquesta prova consistia a variar el llindar a partir del qual una imatge es classifica com a IA (llindar el qual era del 0.5 en la fase anterior). D'aquesta manera, es podia ajustar el comportament del detector sense tornar a entrenar els models.

El resultat del procediment va resultar en que, els millors thresholds per a cada model eren els següents: 0.49 per al Random Forest, 0.33 per al XGBoost i 0.06 per al MLP. En el cas del Random Forest, el valor òptim era molt proper al threshold habitual de 0,5, fet que indicava que les seves probabilitats estaven força ben equilibrades. Per a XGBoost, el threshold òptim era més baix, cosa que suggeria que el model tendia a assignar probabilitats una mica més moderades a la classe IA i que, amb un tall de 0,5, podia classificar algunes imatges generades com a reals. Pel que fa al MLP, aquest era més extrem: el threshold de 0,06 indicava que les seves probabilitats no estaven ben calibrades en termes absoluts. És a dir, encara que el model assignés valors baixos a la classe IA, aquests valors seguien contenint informació útil per separar les classes. Per aquest motiu, en el MLP no s'havia d'interpretar una probabilitat baixa com una baixa evidència d'IA, sinó com una sortida

poc calibrada que necessitava un punt de tall específic.



Fig. 5: Matrius de confusió de la tercera fase

Finalment, un cop aplicats els hiperparàmetres ajustats i els nous thresholds específics per a cada model, es va observar una millora general en el comportament dels classificadors, tot i que no de la mateixa manera en tots els casos. El Random Forest va continuar sent el model més estable, mantenint un bon rendiment tant en el dataset manual com en el dataset AI vs Real Images. XGBoost va obtenir resultats molt propers al Random Forest tant en el conjunt extern com en el manual. En canvi, en el cas del MLP, els canvis en el threshold no van afectar de manera significativa el dataset manual, però sí que van empitjorar la detecció d'imatges generades respecte a la prova anterior, amb una reducció del recall IA del 85,8% al 77,6%. Tot i això, el model va presentar un comportament més equilibrat entre ambdues classes.

Per a la selecció del model definitiu es va comparar el comportament dels tres models al llarg de les diferents fases d'ajust. En el cas del Random Forest i del XGBoost, es van conservar les versions finals amb hiperparàmetres optimitzats i llindars de decisió ajustats, ja que oferien el comportament més equilibrat. En canvi, per al MLP es va mantenir la versió amb hiperparàmetres optimitzats però amb el llindar original, ja que el threshold sweep no aportava una millora clara en el seu cas i podia empitjorar la detecció d'imatges generades.

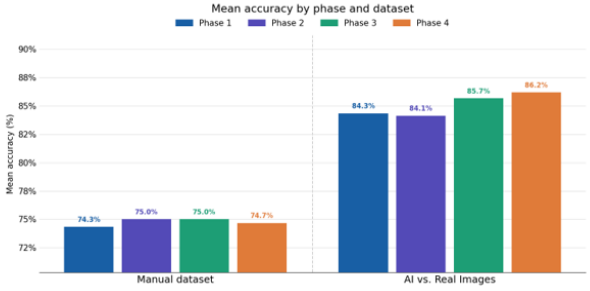


Fig. 6: Precisions mitjanes de cada fase i dataset

Com es mostra a la figura 6, l'evolució de l'accuracy mitjana entre fases va ser moderada. En el dataset manual, els resultats es van mantenir pràcticament estables al voltant del 75%, fet que indica que les diferents fases d'ajust no van modificar de manera 9ic kan9te9ve el comportament en aquest conjunt. En canvi, en el dataset AI vs Real Images sí que s'observa una millora progressiva, passant del 84,3% en la primera fase al 86,2% en la fase final. Això suggereix que els ajustos aplicats van tenir un impacte positiu, però limitat, en la generalització cap a aquest dataset extern.

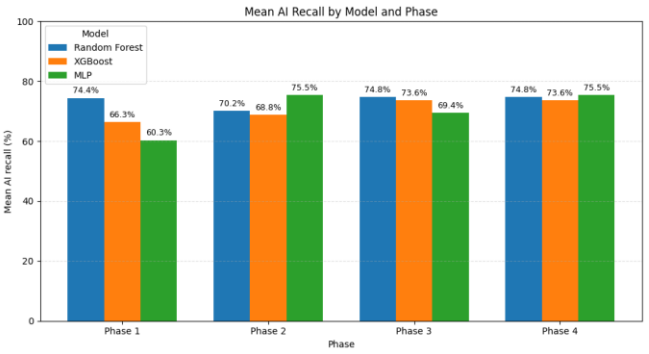


Fig. 7: Mitjanes dels *recalls* en cada model i fase

La Figura 7 mostra el recall mitjà de la classe IA, és a dir, la capacitat dels models per detectar imatges generades. El Random Forest es manté estable al voltant del 75%, mentre que XGBoost millora progressivament del 66,3% al 73,6%. L'MLP presenta un comportament més variable, però acaba obtenint el millor valor en la fase final. En conjunt, les optimitzacions aporten millores moderades, especialment en el dataset AI vs Real Images I en la detecció d'imatges IA, però no resolen completament el problema de generalització. Per això, la selecció final del model no es va basar només en l'accuracy, sinó també en l'estabilitat entre datasets I l'equilibri entre classes.

6 AILENS: EXTENSIÓ WEB

Un dels objectius finals del projecte era poder implementar una extensió per a navegador capaç d'analitzar i donar una predicció sobre qualsevol imatge que l'usuari estigués visualitzant en la pàgina web en la que es trobés. Aquesta extensió representa la integració pràctica del sistema desenvolupat durant el treball, ja que permet portar el model entrenat i el procés d'extracció de característiques a un entorn d'ús real. El funcionament és el següent:

- L'usuari pitja la icona de l'extensió. A continuació, s'obre un desplegable amb 3 opcions:
  a. **Page images:** Es carreguen els títols de totes les imatges de la pàgina i l'usuari selecciona la que desitji.
  b. **Upload a local imatge:** L'usuari selecciona una imatge guardada en el seu dispositiu
  c. **Pick an image:** Clica una imatge de la pàgina per seleccionar-la.
- Després d'haver carregat la imatge amb qualsevol de les 3 opcions, es mostra la pestanya principal donantel resultat de la predicció.

Les altres pestanyes són les següents:

- **Indicators**: mostra els valors dels indicadors en diferents visualitzacions.

- **Models**: mostra la predicció donada per cadascun dels tres models (Random Forest, XGBoost i MLP).
- **Vector**: mostra el vector dels 73 descriptors que s'utilitzen per obtenir la predicció.
- **Time**: mostra un resum de diferents mètriques de rendiment i temps d'execució referents a les prediccions.

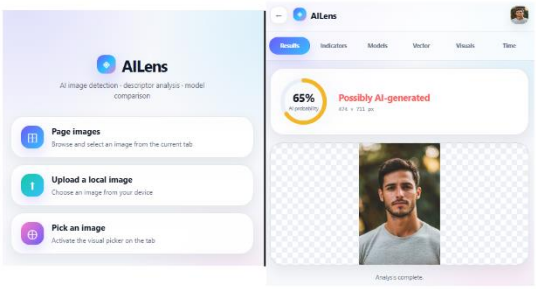


Fig. 8: A l'esquerra, interfície inicial amb opcions de selecció d'imatges. A la dreta, pestanya de resultats. (altres imatges de la interfície en l'apèndix, en la secció A3)

7. Conclusions i pròxims passos

Els resultats obtinguts mostren que l'ús de descriptors estadístics permet construir un detector funcional d'imatges generades per intel·ligència artificial sense necessitat d'utilitzar directament la imatge completa com a entrada d'un model profund. Tot i treballar amb un vector reduït de característiques, els models han aconseguit un rendiment elevat en el dataset principal i resultats competitius en datasets externs. Això indica que propietats com la textura, la distribució freqüencial, el color, el soroll residual, l'entropia o la nitidesa contenen informació rellevant per diferenciar, en molts casos, imatges reals i sintètiques.

Tot i això, també s'ha comprovat que aquests descriptors no són suficients per resoldre el problema de manera completa. El rendiment disminueix quan les imatges provenen de fonts diferents, generadors diferents o condicions de compressió i resolució diverses. Això mostra que el sistema és capaç d'aprendre patrons útils, però que part d'aquests patrons depenen del domini del dataset. Per tant, els descriptors utilitzats són adequats per obtenir una predicció raonable i interpretable, però no permeten garantir una detecció infal·lible en qualsevol context.

Entre els models provats, el Random Forest ha estat l'opció més equilibrada per al sistema final, ja que ha combinat bons resultats, estabilitat en datasets externs, interpretabilitat i facilitat d'integració. XGBoost i MLP també han obtingut resultats competitius, però amb un comportament menys estable segons el conjunt d'avaluació. L'ajust d'hiperparàmetres i del llindar de decisió ha ajudat a millorar lleugerament el comportament dels models, especialment a l'hora d'equilibrar la detecció d'imatges IA i reals, però les millores han estat moderades. Finalment, s'ha pogut desenvolupar una extensió web funcional capaç d'executar tot el procés localment dins del navegador: carregar una imatge, extreure'n els descriptors, aplicar el model entrenat i mostrar una predicció probabilística a l'usuari. Això demostra que l'enfocament proposat és viable en un entorn real d'ús, especialment quan es prioritzen la privadesa, l'eficiència i la interpretabilitat. Com a treball futur, seria interessant incorporar descriptors locals o per regions, ampliar la diversitat dels datasets d'entrenament i explorar combinacions amb models visuals més avançats.

Com a pròxim pas, es pretén publicar una versió més polida i definitiva de l'extensió per a que qualsevol usuari la pugui utilitzar.

8. Bibliografia

[1] Stability AI, 2022, Des. *Stable Diffusion v2.1 and DreamStudio Updates 7-Dec 22*. [Web / nota de llançament]. Stability AI.
[2] Midjourney, s. D. *Midjourney*. [Web / model page]. Midjourney.
[3] OpenAI, 2023, Oct. *DALL·E 3*. [Web / model page]. OpenAI.
[4] Farid, H., 2025. How to spot fake AI photos. TED Talk, TED2025, April 2025. TED. [En línia].
[5] Lu, Z., Huang, D., Bai, L., Qu, J., Wu, C., Liu, X. I Ouyang, W., 2023, Abr. *Seeing is not always believing: Benchmarking Human and Model Perception of AI-Generated Images*. [Preprint]. arXiv.
[6] Velásquez-Salamanca, D., Martín-Pascual, M. Á. I Andreu-Sánchez, C., 2025, Jul. *Interpretation of AI-Generated vs. Human-Made Images*. [Article]. Journal of Imaging.
[7] Maiano, L., Benova, A., Papa, L. I Stockner, M., 2024, Mai. *Human Versus Machine: A Comparative Analysis in Detecting Artificial Intelligence-Generated Images*. [Article]. IEEE Security & Privacy.
[8] J. Cao, Q. Li, Z. Zhang i J. Ni, "Secure and Robust Watermarking for AI-generated Images: A Comprehensive Survey," arXiv, 2025.
[9] Zhang, X., Karaman, S. I Chang, S.-F., 2019, Jul. *Detecting and Simulating Artifacts in GAN Fake Images*. [Preprint]. arXiv.
[10] Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D. I Holz, T., 2020, Mar. *Leveraging Frequency Analysis for Deep Fake Image Recognition*. [Preprint]. arXiv.
[11] Enlaight Software Design, s. D. *AI Image Detector by IsItAI.com*. [Extensió web]. Chrome Web Store.
[12] Hive, s. D. *Hive AI Detector*. [Extensió web]. Chrome Web Store.
[13] Farid, H., 2025, Abr. *How to Spot Fake AI Photos*. [Vídeo]. TED.
[14] Karageorgiou, D., Papadopoulos, S., Kompatsiaris, I. I Gavves, E., 2025, Jun. *Any-Resolution AI-Generated Image Detection by Spectral Learning*. [Paper]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
[15] Bammey, Q., 2024. *Synthbuster: Towards Detection of Diffusion Model Generated Images*. [Article]. IEEE Open Journal of Signal Processing.
[16] Sendjasni, A. I Larabi, M.-C., 2026, Mar. *Multi-Feature Fusion Approach for Generative AI Images Detection*. [Preprint]. arXiv.
[17] Baru, L. B., Boddeda, R., Patel, S. A. I Gajapaka, S. M., 2025. *Wavelet-Driven Generalizable Framework for Deepfake Face Forgery Detection*. [Workshop paper]. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW).
[18] Rajarshi-Roy-research, 2026, Mai. *Defactify_Image_Dataset*. [Dataset]. Hugging Face.
[19] Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L. I Dollár, P., 2014. *Microsoft COCO: Common Objects in Context*. [Dataset / paper]. European Conference on Computer Vision (ECCV).
[20] Stability AI, 2023, Jul. *Announcing SDXL 1.0*. [Web / nota de llançament]. Stability AI.
[21] Stability AI, 2024, Jun. *Stable Diffusion 3 Medium*. [Web / nota de llançament]. Stability AI.
[22] Sala, A., s. D. *AI vs. Human-Generated Images*. [Dataset]. Kaggle.
[23] Ghai, R., s. D. *AI vs Real Images Dataset*. [Dataset]. Kaggle.

9. Ús de la IA generativa

S'ha emprat intel·ligència artificial generativa per revisar i redactar textos, amb l'objectiu de garantir-ne la correcció gramatical i millorar-ne la fluïdesa per al lector.

APÈNDIX

A1. GRÀFICS DE RENDIMENT DELS MODELS

A1.1 Primeres implementacions dels models classificadors

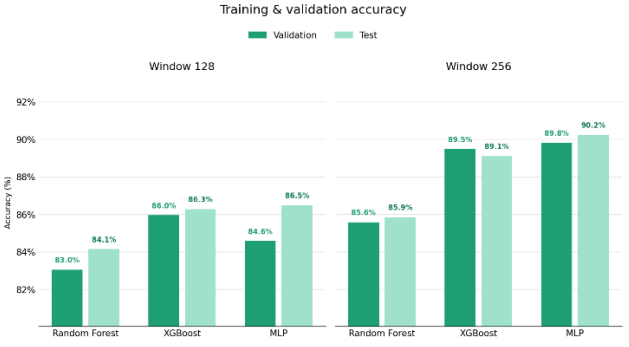


Fig 9. Accuracies de validació i test dels primers models amb finestres de 128×128 i 256×256 píxels.

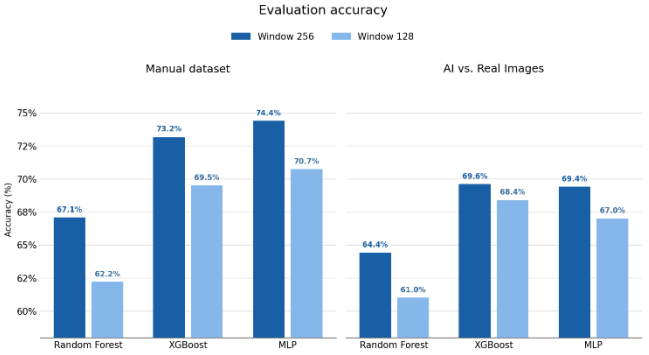


Fig 10. Accuracies d'avaluació dels primers models amb finestres de 128×128 i 256×256 píxels.

A1.2 Implementació de models compatibles amb el navegador

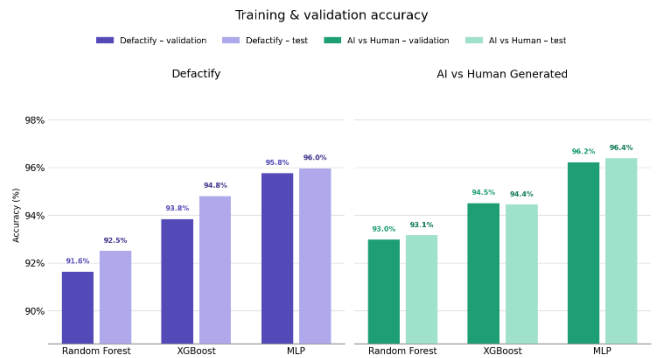


Fig 11. Accuracies de validació i test dels models amb els nous features i pipeline.

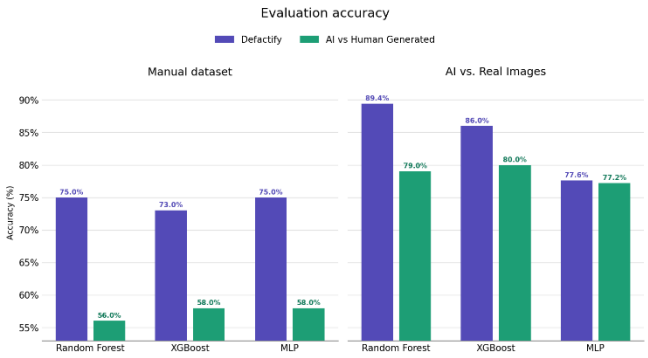


Fig 12. Accuracies d'avaluació dels models amb els nous features i pipeline.

A1.3 Experiments finals: combinació de datasets i reducció de descriptors

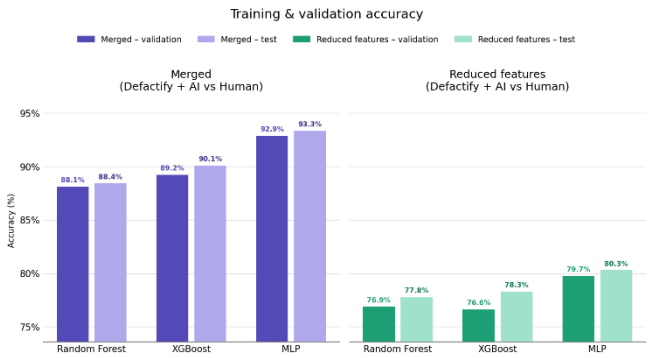


Fig 13. Accuracies de validació i test dels models amb els datasets units i amb menys descriptors.

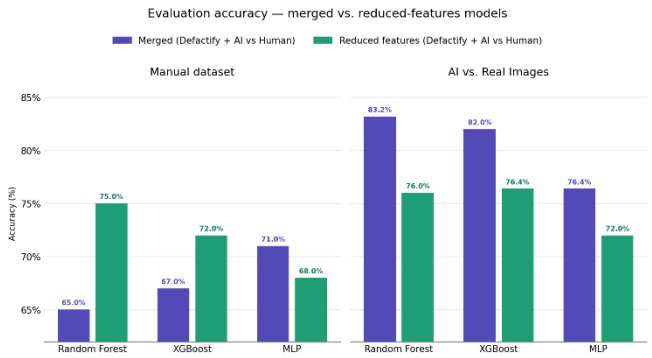


Fig 14. Accuracies d'avaluació dels models amb els detestes units i amb menys descriptors.

A2. Canvis ens els hiperparàmetres

TAULA 2: Canvis en els hiperparàmetres principals.

Model	Hiperparàmetre	Versió antiga	Versió nova
Random Forest	n_estimators	300	500
Random Forest	max_depth	14	18
Random Forest	max_features	0.6	0.45
Random Forest	min_samples_split	2	2
Random Forest	min_samples_leaf	8	1
Random Forest	bootstrap	true	True
Random Forest	class_weight	balanced	balanced
Random Forest	random_state	—	RANDOM_SEED
Random Forest	n_jobs	—	-1
XGBoost	n_estimators	1000	500
XGBoost	max_depth	3	5
XGBoost	learning_rate	0.08	0.03
XGBoost	subsample	0.7	0.9
XGBoost	colsample_bytree	0.85	0.8
XGBoost	min_child_weight	8	—
XGBoost	gamma	0.3	—
XGBoost	reg_alpha	0.01	—
XGBoost	reg_lambda	1	—
XGBoost	scale_pos_weight	0.6	—
XGBoost	objective	—	binary:logistic
XGBoost	eval_metric	—	logloss
XGBoost	tree_method	—	hist
XGBoost	random_state	—	RANDOM_SEED
XGBoost	n_jobs	—	-1
MLP	hidden_layer_sizes	(64, 32)	(96, 48)
MLP	activation	relu	relu
MLP	solver	adam	adam
MLP	alpha	0.0001	0.0001
MLP	batch_size	256	128
MLP	learning_rate_init	0.001	0.001
MLP	max_iter	250	250
MLP	early_stopping	true	True
MLP	validation_fraction	0.12	0.12
MLP	n_iter_no_change	25	20
MLP	random_state	—	RANDOM_SEED
MLP	verbose	—	False

A3. PESTANYES DE L'EXTENSIÓ WEB (AILENS)

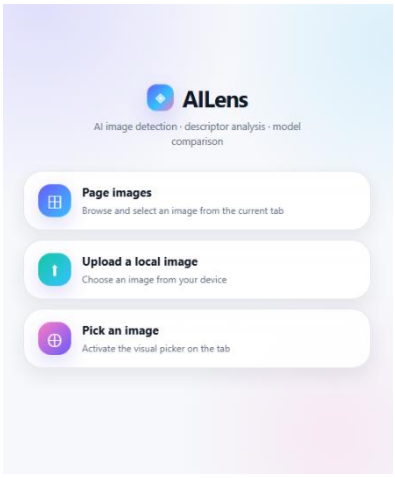


Fig. 15: Pàgina principal de selecció d'imatge.

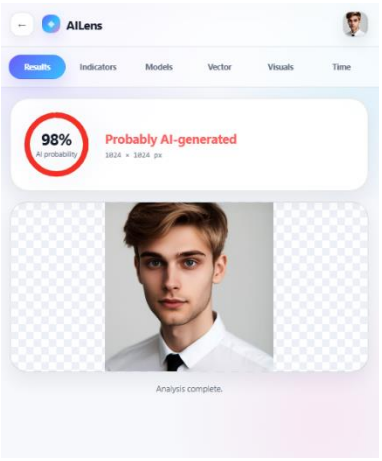


Fig. 16: Pestanya de resultats.

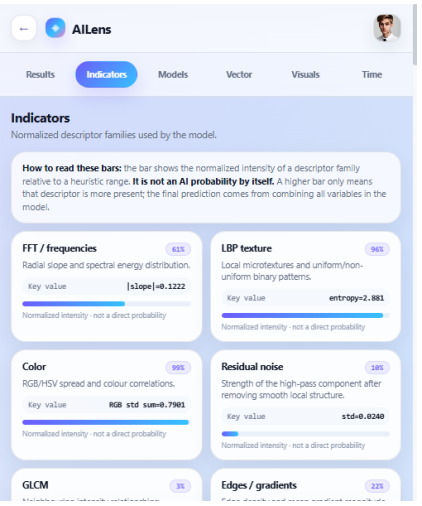


Fig. 17: Pestanya d'indicadors.

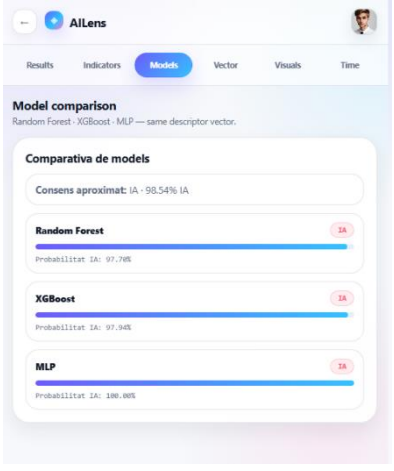


Fig. 18: Pestanya de les prediccions dels 3 models.

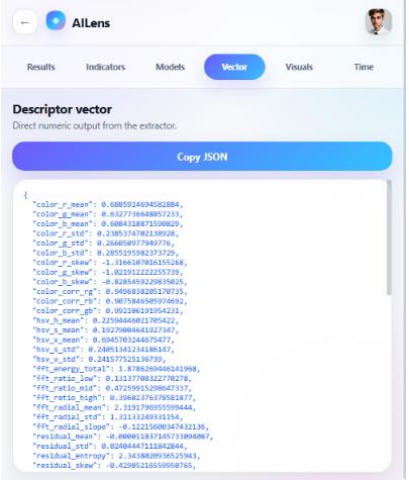


Fig. 19: Pestanya del Vector de característiques.

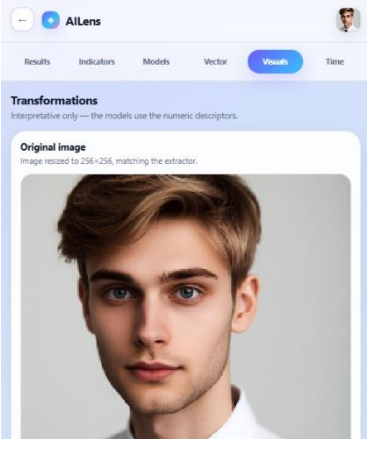


Fig. 20: Pestanya de les transformacions visuals (imatge redimensionada a la window de 256x256).

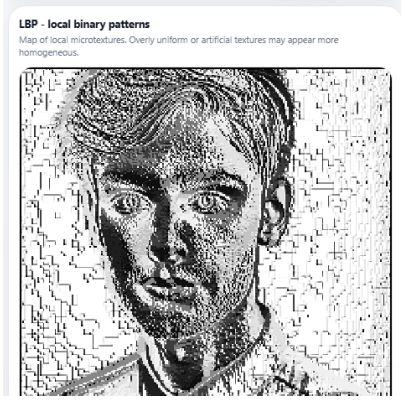


Fig. 21: Pestanya de les transformacions visuals (LBP).

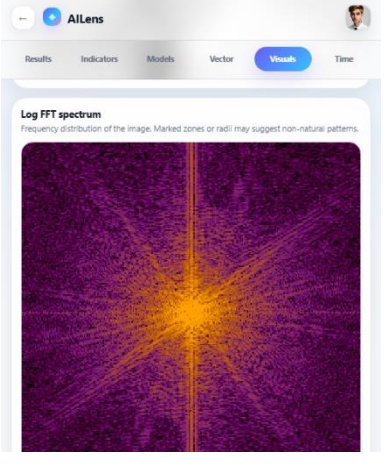


Fig. 22: Pestanya de les transformacions visuals (FFT).

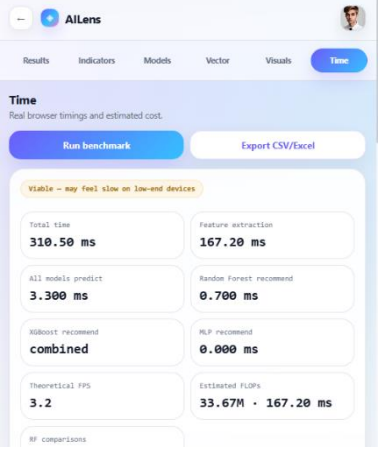


Fig. 23: Pestanya Time amb les mètriques de temps i execució.