UAB
Universitat Autònoma de Barcelona

Dipòsit digital de documents de la UAB

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

Ochoa García, Raúl; Vazquez Corral, Javier, dir. Desenvolupament d'un Sistema d'Image Enhancement Interactiu Basat en Exemples Estilístics. 2024. (Enginyeria Informàtica)

This version is available at https://ddd.uab.cat/record/308773

under the terms of the  license

Desenvolupament d'un Sistema d'Image Enhancement Interactiu Basat en Exemples Estilístics

Raul Ochoa García

Resum — Aquest treball demostra que és possible guiar l'edició automàtica d'imatges a partir de la selecció d'exemples estilístics, millorant l'experiència dels usuaris. Els mètodes actuals aborden la tasca d'Image Enhancement amb un enfocament d'extrem a extrem, imitant l'estil d'un fotògraf expert. Això limita la capacitat dels usuaris per influir en el resultat final, sovint generant insatisfacció. Per resoldre aquesta limitació, proposem un sistema basat en tècniques d'aprenentatge profund com Xarxes Neuronals Convolucionals (*CNN*s) i *Transformers* que permet als usuaris seleccionar exemples de referència per definir l'estil desitjat. El sistema extreu característiques de la imatge exemplar i les integra a l'espai latent d'una arquitectura en forma de U amb blocs de *Transformer*, aplicant millores personalitzades. Els resultats mostren millores en la il·luminació i els tons dels colors, mantenint la coherència visual de la imatge original. Aquest enfocament simplifica l'edició fotogràfica, fent-la accessible per a tothom, independentment del seu nivell d'experiència.

Paraules clau — Aprenentatge profund, conjunt de dades, codificador, decodificador, xarxa neuronal convolucional (CNN), Restormer, classificador, funció de pèrdua, millora d'imatges

Abstract — This work demonstrates that it is possible to guide automatic image editing based on the selection of stylistic examples, improving the user experience. Current methods approach the Image Enhancement task with an end-to-end approach, imitating the style of an expert photographer. This limits the ability of users to influence the final result, often generating dissatisfaction. To solve this limitation, we propose a system based on deep learning techniques such as Convolutional Neural Networks (*CNN*s) and *Transformers* that allows users to select reference examples to define the desired style. The system extracts features from the exemplary image and integrates them into the latent space of a U-shaped architecture with *Transformer* blocks, applying personalized improvements. The results show improvements in lighting and color tones, while maintaining the visual coherence of the original image. This approach simplifies photo editing, making it accessible to everyone, regardless of their level of experience.

Index Terms — Deep learning, dataset, encoder, decoder, convolutional neural network (CNN), Restormer, classifier, loss function, image enhancement

1 Introducció - Context del treball

Aquest projecte sorgeix a partir de la proposta dels professors Javier Vázquez Corral, com a tutor del treball, i David Serrano, i s'alinea amb la recerca publicada a la *European Conference on Computer Vision* el 2024, on es van produir avenços en el camp de l'Image Enhancement.

La coherència estètica i l'impacte visual són aspectes determinants de l'èxit en sectors com les xarxes socials i el màrqueting en el context digital en el que ens trobem. Aquests factors influeixen directament en la capacitat de capturar l'atenció i en la transmissió d'un missatge.

És per això que definir un estil compacte de millora d'imatges és cada vegada més necessari. Encara que ja existeixen eines amb gran renom especialitzades en aquesta tasca, sovint estan dissenyades per a usuaris experts, ja que exigeixen coneixements avançats d'edició i un cert sentit estètic, el que fa que les persones amb poca experiència o amb un sentit estètic poc desenvolupat es trobin amb barreres importants per aconseguir resultats consistents i de qualitat. Fins i tot els experts tenen dificultats per respectar un mateix estil en grans conjunts d'imatges, pel que resulta necessari plantejar un enfocament més accessible i personalitzat.

Per fer front a aquests inconvenients el projecte té com a objectiu canviar l'enfocament cap a un model interactiu on l'usuari sigui capaç de guiar el procés d'edició seleccionant un seguit d'exemples que defineixin l'estil que desitja.

La motivació principal del treball rau en la necessitat de proporcionar un Sistema d'Image Enhancement Interactiu en el que els usuaris més inexperts puguin, sense esforç ni coneixements previs, retocar grans conjunts d'imatges d'una manera simple i consistent des del punt de vista estilístic.

- *E-mail de contacte: 1598545@uab.cat*
- *Menció de Computació*
- *Treball tutoritzat per Javier Vázquez Corral (departament)*
- *Curs 2024/25*

2 ESTAT DE L'ART

2.1 Tècniques tradicionals de millora d'imatges

Històricament, la millora de les imatges [1] s'ha enfocat en tècniques automàtiques i manuals. Els mètodes tradicionals, com ara la igualació d'histograma o les variants dels mètodes retinex, modifiquen els rangs de color i il·luminació de la imatge per millorar el contrast. Aquests enfocaments tenen limitacions, ja que solen aplicar una millora genèrica sense considerar les preferències estètiques de l'usuari. A l'any 2011, el treball de Vladimir Bychkovsky [21] va servir com a base per tots els treballs posteriors en el camp d'Image Enhancement. Els autors van crear una base de dades anomenada *MIT-A5K*, formada per 5000 parells de fotografies retocades per cinc fotògrafs. Això va servir per entrenar models basats en CNNs basats en aquestes referències.

2.2 Enfocaments basats en *Deep Learning*

Amb l'avenç de l'aprenentatge profund, s'han desenvolupat enfocaments més avançats que utilitzen xarxes neuronals per resoldre el problema de millora d'imatges. Aquests enfocaments busquen obtenir resultats més precisos que els mètodes tradicionals. Tot i això, aquests mètodes, encara que efectius en tasques generals de millora, no tenen en compte les preferències subjectives dels usuaris.

2.2.1 *PieNet*: millora personalitzada d'imatges

El treball de Sing Bing Kang l'any 2010 [19] va ser un dels primers intents per abordar la personalització en la millora d'imatges, permetent als usuaris ajustar manualment un conjunt de paràmetres en imatges representatives per després aplicar aquests ajustaments a noves imatges. Aquest enfocament, però, implicava un esforç considerable per part de l'usuari. Posteriorment, Juan Carlos Caicedo l'any 2011 [20] va ampliar el sistema de Sing Bing Kang utilitzant filtratge col·laboratiu per considerar els resultats de millora d'altres usuaris amb preferències similars. Tot i així, aquests mètodes no aconseguien capturar completament les preferències estètiques individuals. Per abordar aquesta limitació s'han proposat tècniques d'aprenentatge mètric, com ara l'ús de la *triplet loss* [1], per aprendre un espai de característiques on les preferències d'un usuari específic es representin mitjançant un vector de preferència. Aquest enfocament permet agrupar imatges amb estils preferits per l'usuari i diferenciar-les d'aquelles que no li agraden, millorant significativament la personalització en comparació amb els mètodes previs.

El treball més recent i avançat en aquest camp és *PieNet* [1], una xarxa basada en aprenentatge profund que proposa una arquitectura *encoder-decoder*. *PieNet* utilitza vectors de preferència, que es generen en demanar a l'usuari seleccionar entre 10 i 20 imatges d'un conjunt aleatori. Aquesta informació es fa servir per generar resultats ajustats a les preferències estètiques de l'usuari. Permet una personalització amb èxit, però amb un esforç considerable per part de l'usuari, al haver de seleccionar una gran quantitat d'imatges. El problema principal d'aquest mètode és que precisament assigna el mateix vector de preferència a totes les imatges, ignorant el seu contingut.

2.2.2 Models basats en estil i contingut

Per abordar aquesta limitació, models com el *Masked Style Modeling* [2], inspirat en l'emmascarament de llenguatge, prediu característiques considerant tant el contingut com l'estil de les imatges. Aquest enfocament utilitza un codificador per predir l'estil d'una imatge no vista, basant-se en característiques de les imatges preferides per l'usuari.

A diferència de models previs com *PieNet*, que apliquen un únic estil a totes les imatges d'un usuari, l'enfocament d'estil emmascarat permet una personalització conscient del contingut on l'estil aplicat varia segons les característiques visuals de la imatge.

2.2.3 Model *Restormer*

L'article de Syed Waqas Zamir i Aditya Arora [13], entre d'altres, proposa un model basat en blocs de *Transformer* amb l'objectiu de millorar les deficiències de les *CNN* en tasques de restauració d'imatges. S'enfoca principalment en reduir la complexitat computacional i en l'eficiència en capturar interaccions globals entre píxels.

Encara que les *CNN* presenten certa utilitat a l'hora de restaurar imatges, el seu camp és limitat i no són capaces de modelar relacions més enllà de les locals. Els *Transformers*, emprant el seu mecanisme de *self-attention*, poden capturar grans contextos a canvi d'una alta complexitat computacional.

En aquest context, *Restormer* ofereix un avenç en la restauració mitjançant els blocs:

1. **Multi-Dconv Head Transposed Attention (MDTA)**: redueix la complexitat atès que utilitza convolucions profundes per capturar el context local abans de calcular la connexió global entre les característiques.
2. **Gated-Dconv Feed-Forward Network (GDFN)**: xarxa *feed-forward* amb un mecanisme que regula el flux d'informació, transmetent únicament les característiques que considera més rellevants.

El model *Transformer* es va desenvolupar per primera vegada per processar seqüències en tasques de llenguatge natural. S'ha anat adaptant en tasques de visió, com ara reconeixement d'imatges, segmentació i detecció d'objectes. Els *Vision Transformers* descomposen una imatge en una seqüència de finestres i aprenen les seves relacions mútues (*self-attention*), encara que no arriben a captar les dependències globals. A causa d'aquestes característiques, els models *Transformer* també s'han estudiat per als problemes de visió com la superresolució i la coloració d'imatges. *Restormer* supera aquests enfocaments modelant relacions sense dividir les imatges en parts. El model *Restormer* s'ha provat en tasques de restauració com:

1. Eliminació de pluja (*deraining*)
2. Desenfocament de moviment (*motion deblurring*)
3. Desenfocament de focus (*defocus deblurring*)
4. Eliminació de soroll (*denoising*)

2.3 Esquemes d'entrenament

L'entrenament per al *PIE* basat en contingut requereix parells d'imatges originals i millorades. Es proposa un esquema que utilitza imatges retocades per usuaris reals en plataformes com *Flickr*, creant parells d'imatges degradades i millorades mitjançant un model de degradació entrenat. Això permet que el model aprengui a millorar imatges tenint en compte tant el contingut com les preferències de múltiples usuaris.

L'entrenament de *Restormer* comença amb imatges petites i lots grans, però a mida que avança s'augmenta la mida progressivament i es redueix la mida dels lots. La seva arquitectura i la técnica progressiva que empra estabilitza el model i optimitza la complexitat computacional.

2.4 Comparació amb mètodes existents

En experiments quantitatius, el model basat en estil emmascarat ha demostrat superar els mètodes anteriors, tant en mètriques objectives (com *PSNR* i *SSIM*) com en estudis d'usuari, on els resultats generats van ser preferits consistentment. Aquest enfocament resol problemes de personalització limitada i manca de consciència del contingut que afectaven els models com *PieNet* basats en vectors de preferència únics.

Per la seva part, *Restormer* és significativament més eficient en termes de còmput. utilitza una arquitectura de Transformer optimitzada per a la restauració d'imatges d'alta resolució, combinant la capacitat dels transformadors per capturar interaccions globals amb un disseny que redueix la complexitat computacional. Això s'aconsegueix mitjançant els blocs *MDTA* i *GDFN*, que milloren la qualitat de les imatges restaurades. Restormer ha establert nous estàndards de rendiment en múltiples conjunts de dades de referència.

3 Objectius

A continuació s'enumeren, per ordre de prioritat, els objectius definits:

- **Objectiu 1**. Investigar i implementar algoritmes d'extracció d'embeddings d'imatges per capturar de manera precisa l'estil i l'aparença de les imatges seleccionades per l'usuari.
- **Objectiu 2**. Comparar mètodes com *encoders o CLIP* i determinar quin d'ells manté millor l'estil.
- **Objectiu 3**. Entrenar xarxes neuronals per millorar automàticament les imatges basant-se en estils personalitzats.
- **Objectiu 4**. Desenvolupar una interfície d'usuari accesible i amigable dirigida principalment als usuaris més innexperts i sense nocions.

4 Metodologia

L'enfocament triat per desenvolupar aquest treball ha sigut el Scrumban, com a combinació de les metodologies àgils *Scrum* i *Kanban*.

Aquest unió de metodologies permet aprofitar aspectes positius de cadascuna d'elles. Scrum permet dividir el projecte en fases, conegudes com a Sprints, amb una durada determinada, en les que es desenvolupen tasques relacionades amb els objectius. En aquest cas, els Sprints tindran una durada d'una setmana. En referència a Kanban, aquest mètode s'utilitza per consultar de manera visual l'estat de les tasques programades a través d'un taulell dividit en les columnes "To do", "In progress" i "Done".

Cal destacar que degut a que aquesta metodologia està orientada per projectes en grup, cal incorporar certs canvis.

4.1 Introducció d'esmenes

Encara que d'acord amb el tutor es va determinar realitzar Sprints setmanals, no en cadascun d'ells s'ha pogut reportar la finalització d'una tasca, atès que hi han hagut certes tasques que, per qüestions de complexitat, han requerit més temps que d'altres. Llavors, el feedback s'ha anat rebent a mida que es progressava en la tasca enlloc de rebre'l només quan aquesta finalitzava.

5 Planificació

Aquesta secció presenta el calendari del projecte, que ha sigut dividit en 4 parts essencials. Cadascuna d'aquestes seccions comprèn subfases, que corresponen a tasques específiques amb dates d'inici i de finalització i els Sprints durant els que s'han desenvolupat. D'acord amb el tutor, es va considerar que per un millor seguiment i valoració contínua del treball es realitzarien Sprint setmanals.

La FIGURA 1 mostra de manera ordenada la distribució de les fases i les subfases, seguides de les columnes amb les dates i els Sprints esmentats. La primera reunió entre l'estudiant i el tutor, en la que es va presentar i contextualitzar el treball, és l'única tasca que no s'ha considerat que pertanyia a cap Sprint.

Fases i subfases	Inici	Final	Sprint
Investigació			
Reunió estudiant/tutor	09/09/2024	09/09/2024	--
Recerca de projectes relacionats	10/09/2024	14/09/2024	1
Lectura de Papers de conferències	16/09/2024	22/09/2024	2
Recerca de tutorials sobre Pytorch	23/09/2024	29/09/2024	3
Recerca de notebooks sobre Pytorch	23/09/2024	29/09/2024	3
Definició			
Creació Scrumban Jira	15/09/2024	15/09/2024	1
Objectius	23/09/2024	06/10/2024	3 – 4
Metodologia	23/09/2024	06/10/2024	3 – 4
Implementació			
Primeres proves amb Pytorch	26/09/2024	06/10/2024	3 – 4
Entrenaments inicials amb Pytorch	07/10/2024	20/10/2024	5 – 6
Treballar amb Restormer	21/10/2024	24/11/2024	7 – 11
Extracció embedding de poques imatges	25/11/2024	15/12/2024	12 – 14
Relacionar imatge amb embedding	16/12/2024	12/01/2025	15 – 18
Producte objectiu	13/01/2025	26/01/2025	19 – 20
Documentació			
Informe inicial	23/09/2024	06/10/2024	3 – 4
Informe de progrés I	28/10/2024	10/11/2024	8 – 9
Informe de progrés II	02/12/2024	15/12/2024	13 – 14
Proposta Informe final	07/01/2025	19/01/2025	18 – 19
Proposta de presentació	24/01/2025	02/02/2025	20 – 21
Dossier TFG	27/01/2025	04/02/2025	21 – 22
Defensa TFG	17/02/2025	20/02/2025	24

FIGURA 1: Planificació TFG

Durant el desenvolupament del treball, la planificació ha patit diverses modificacions, encara que entre el període de temps comprès entre el lliurament de l'informe inicial i l'informe de progrés I la planificació inicial es va seguir correctament fins al Sprint 6.

De cara als Sprints 7 i 8, per tal d'afegir dificultat i anar orientant la feina cap el propòsit del treball, el tutor Javier Vázquez i en David Serrano van suggerir començar a treballar amb un model de restauració d'imatges. Per aquest motiu i el que es relata a la subsecció *6.2 Proves amb Restormer*, la planificació es va haver de modificar, definint una nova subfase del Sprint 7 al 10. D'acord als problemes que va comportar el desenvolupament d'aquesta subfase, el treball en aquesta part es va allargar fins ben entrat el Sprint 11.

Després, la implementació de *Extracció d'embedding de poques imatges*, referent al desenvolupament de l'*autoencoder* i a l'extracció de característiques que aquesta xarxa neuronal fa, va començar al Sprint 12 i es va allargar durant 3 Sprints. Això va causar una altra redistribució del calendari.

Al Sprint 19 es va construïr el producte objectiu del treball, fent una sinapsi dels blocs més importants vistos durant la fase de *Implementació*. Això ha donat lloc a disposar d'un sistema de millora d'imatges flexible i senzill d'utilitzar.

L'objectiu 4 estava lligat a una fase especifica de disseny, però degut a les modificacions efectuades en el transcurs del treball aquesta fase es va acabar descartant, encara que es considera una tasca pendent per treballs a futur.

La FIGURA 2, el Diagrama de Gantt que es recull a la secció d'Annex, mostra la distribució del calendari del projecte. Es mostren indicadors individuals tant per les subtasques com per les fases, que abarquen tot el llarg de les seves subfases. Es representa també el Sprint en el que ens trobem.

6 DESENVOLUPAMENT

Aquesta secció detalla les diferents etapes en termes de implementació del projecte. Abasta des de la introducció teòrica i pràctica al món de les *CNN*, emprant diferents datasets per a tasques de classificació d'imatges, com el *CIFAR-10* i el *MNIST*, fins a l'ús de models com el Restormer. Cal destacar la creació d'un autocodificador per representar dades de manera comprimida, del que es va derivar en un classificador d'aquestes dades extretes. Per últim, es van ajuntar tots aquests mòduls per donar lloc a un sistema d'edició basat en el context. Essencialment, les tasques d'entrenaments s'han llançat en un servidor a causa de la complexitat computacional dels diferents models.

FIGURA 3: Batch de 4 imatges del dataset *CIFAR10*

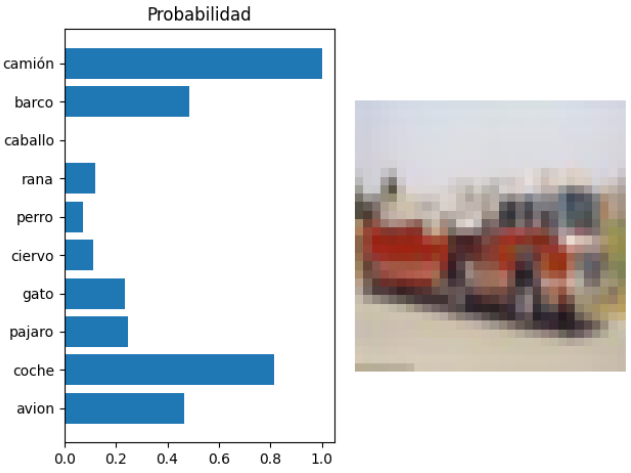


FIGURA 4: Classificació de *CIFAR10*

6.1 Capacitació amb xarxes neuronals

S'han realitzat entrenaments model de *CNN* per començar a introduir-me en el funcionament de les xarxes neuronals convolucionals. En aquests entrenaments el que s'ha fet ha sigut classificar la pertinença d'imatges a una classe mitjançant els datasets *CIFAR10* [8] (animals i vehicles) i *MNIST* (números del 0 al 9) [3].

La FIGURA 3 mostra un grup d'imatges format a partir de la lectura del dataset, mentre que la FIGURA 4 realitza la classificació d'una imatge d'un d'aquests *batches* en les diferents classes.

6.2 Proves amb *Restormer*

El model introduït al Sprint 7 és el *Restoration Transformer* o *Restormer*. Aquest model, extret d'un repositori de la *Computer Vision and Pattern Recognition Conference (CVPR)* 2022 [11], assoleix bons resultats quan es tracta d'eliminar distorsions (*deraining*), desenfocaments de focus (*defocus deblurring*) i de moviment (*motion deblurring*) i soroll (*denoising*).

Abans de començar a tractar el contingut del repositori torno a consultar uns vídeos del MIT on es fa una introducció al *Deep Learning* [9] i a les *CNN* [10] i opto per reproduir també un altre que explica en què consisteixen els *Transformers* [12], a mode de complement.

Per treballar amb la restauració d'imatges els tutors creuen convenient començar amb la eliminació del soroll.

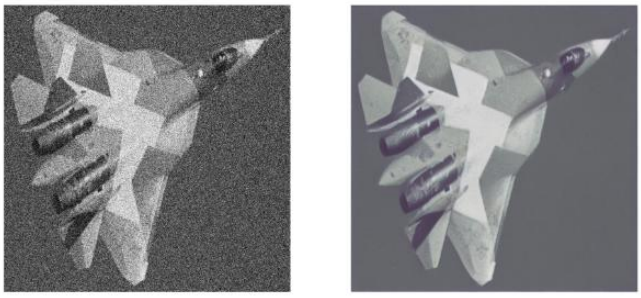


FIGURA 5: Resultats demo Real Denoising

Degut a les meves limitacions de hardware i després de tractar-ho amb en Javier i en David, es considera oportú donar-me accés al clúster del *Centre de Visió per Computador* de la UAB perquè pugui llançar entrenaments. El procés d'autorització triga més del que s'esperava, pel que mentrestant treballo amb una petita demostració del model preentrenat de la eliminació de soroll amb *Google Colaboratory*. Aquesta demo representa una versió simplificada i compacte que ja permet comprendre el funcionament del *denoising* i serveix per observar ràpidament el funcionament del model.

La FIGURA 5 mostra a la part esquerra la imatge original carregada a la demo i a la part dreta la imatge resultant després de treure-li el soroll.

Amb l'accés autoritzat al clúster he provat a llançar entrenaments del model complert del *denoising*, tot seguint les pautes indicades al repositori [11]. Això ha facilitat el treball i m'ha servit essencialment per comprendre com s'han d'executar treballs en remot.

6.3 Autoencoder

Durant el Sprint 13 els tutors determinen que ja cal començar a treballar amb el tema principal del projecte i m'introdueixen en el funcionament d'un *autoencoder* [14]. M'aconsellen com puc fer la seva implementació i em faciliten un dataset utilitzat prèviament en un TFM tutoritzat pel Javier Vázquez.

L'objectiu és desenvolupar un autocodificador complert, amb la seva part d'*encoder* i de *decoder*, per assegurarnos que el model aprèn característiques de les imatges. El dataset amb el que treballo conté 100 imatges originals extretes del dataset *MIT5K* a les que se les han aplicat 7 *Look Up Tables* (*LUT*) diferents: contrast, brillantor blau, efecte de pel·lícula, ombra, llum de lluna, contrast profund i colors de l'arc de Sant Martí.

He declarat una classe *Autoencoder* que serveix per construir una *CNN* [10]. Aquesta xarxa conté un *encoder* i un *decoder*, cadascun d'ells amb 4 capes i amb l'objectiu d'extreure *embeddings* de les imatges, a més d'un mètode *forward*.

Pel que fa l'*encoder*, a cada capa s'utilitzen la convolució, per extreure característiques, el *pooling*, per reduir la

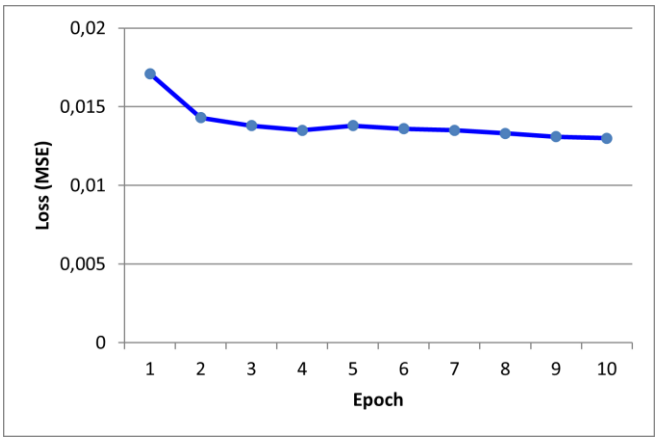


FIGURA 6: Loss function de l'autoencoder

dimensionalitat i una funció d'activació, *ReLU*. Per la seva part, el *decoder* empra el *umpooling*, per tornar a expandir les dimensions amb els índexs extrets de l'*encoder*, la convolució transposada, per desfer la convolució aplicada anteriorment, i de nou la funció d'activació *ReLU*.

El mètode *forward* defineix el flux de les dades pel model, on les imatges de dataset viatgen per les 4 capes de l'*encoder* i redueixen la seva dimensionalitat i un cop comprimides s'expandeixen a les capes del *decoder*, amb l'objectiu de reconstruir l'entrada.

Paral·lelament, una classe *ImageLUTDataset* em permet llegir les imatges vàlides del dataset, així com exclusivament aquelles a les que se les han aplicat una *LUT*, per utilitzar-les posteriorment. Aquesta classe conté també un mètode amb el que carregar directament una imatge a partir del seu índex.

Atès que les imatges del dataset tenen dimensions i orientacions diferents, el més àgil és redimensionar-les a un format quadrat encara que també es podria considerar la rotació en el cas de les que estàn en mode retrat.

El model s'entrena amb 10 *epochs*, iterant sobre el *dataloader*, calculant la pèrdua, amb *MSE*, entre la imatge reconstruïda i l'original, calculant el gradient de pèrdua respecte els pesos i actualitzant els pesos mitjantçant l'optimitzador *Adam*. La següent taula recull els resultats extrets en avaluar la reconstrucció d'imatges.

Que els valors de la pèrdua de la FIGURA 6 siguin decreixents indica que el model ha anat aprenent a extreure característiques i minimitzar les diferències entre les imatges originals i les reconstruïdes.

Tot i això, decidim probar altres tècniques més simples i directes com és el cas d'un classificador.

6.4 Classificador

Arribat el Sprint 15, el següent és aprofitar part de l'*autoencoder* desenvolupat per generar un classificador.

En aquest context, es manté la classe *Autoencoder* utilitzada durant la subsecció anterior però només es conserva l'*encoder*. El codificador està format per 4 capes que contenen una operació de convolució i una de *pooling*. Es genera un vector aplanat de dimensions reduïdes. Durant la implementació de l'*autoencoder* es va caure en l'error de crear diverses instàncies de la funció d'activació *ReLU*, pel que ara es redueix a una única instància.

Seguidament s'afegeix el que correspon al classificador, que empra les característiques compreses per classificar les imatges del dataset en una de les 7 classes. El que es fa és convertir el tensor comprimit, aplanat pel codificador, en un vector mitjançant *flatten*. Tot seguit es declaren dues capes totalment connectades, una que mapeja les 1024 característiques del vector en 128 neurones i una altra que produeix 7 valors, els corresponents a les probabilitats de pertinença a les classes.

El mètode *forward* defineix com viatgen les dades d'entrada a través de les capes del model, de manera que les convolucions s'activen amb la única funció *ReLU* definida.

Per recuperar les imatges del dataset es manté la classe *ImageLUTDataset* que es va crear anteriorment.

Degut a que es considera que redimensionar les imatges pot afavorir a perdre informació, s'opta per utilitzar imatges definides des del seu punt central amb la transformació *CenterCrop*. Així, les fotogràfies prenen una dimensió final del 1024x1024 píxels.

El dataset es divideix en 3 parts: *train, validation* i *test*. La part d'entrenament representa un 60% del conjunt de dades i les de validació i prova un 20% cadascuna.

El model s'entrena amb 60 *epochs*, iterant sobre els *batches* de dades extretes de la part de *train*: s'obtenen les prediccions del model, es calculen la pèrdua i les prediccions correctes, s'actualitzen els gradients de pèrdua respecte els paràmetres del model i s'actualitzen els pesos amb l'optimitzador. Per disposar d'un índex d'exactitut del model, es calcula la precisió d'encert, l'*accuracy*. La validació permet ajustar els hiperparàmetres del model i ajuda a detectar si es produeix *overfitting* en lloc de generalització. Per últim, la prova permet extraure el rendiment general del model i generar una figura en la que es recull la classificació de pertinença a cada classe i la imatge original amb la *LUT* que representa.

Llançat tot el procés, s'obtenen mètriques de precisió del 96% per la part d'entrenament, del 33% per la validació i del 32% per la prova, el que indica clarament que el model ha sigut sobreentrenat. Per solucionar aquest defecte s'aconsella incorporar l'ús de *dropout* [17], que consisteix en destruir neurones de la xarxa, amb les seves connexions

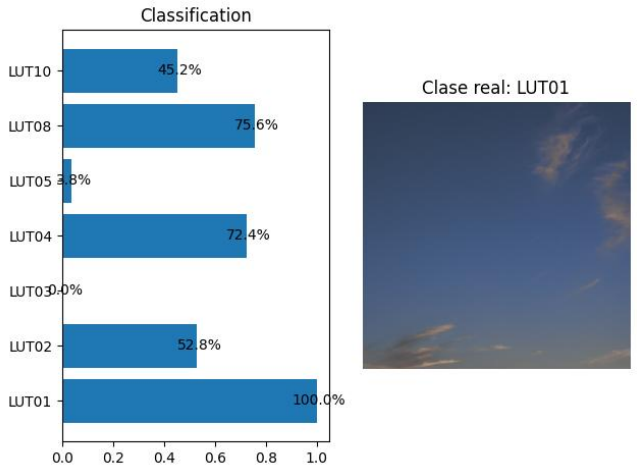


FIGURA 7: Resultats del model Classificador

d'entrada i de sortida. Això obliga el model a no dependre únicament d'aquelles neurones, sinó que ha de diversificar el seu enfocament i desenvolupar altres mètodes per aconseguir el mateix resultat. La introducció d'aquesta esmena permet aconseguir un valor per *test* de 52,5% d'*accuracy*. Fixant-nos en el resultat obtingut a la prova podem concloure que els hiperparàmetres s'ajusten gaire bé.

Les figures generades durant la fase de *test* es guarden a un directori per poder recuperarles en qualsevol moment.

La FIGURA 7 mostra el percentatge de probabilitat amb el que el model creu que la imatge pertany a cada classe.

6.5 Sistema de millora d'imatges

Aquesta subsecció presenta una proposta d'Image Enhancement. Consisteix en un compendi de tots els mòduls desenvolupats durant l'esdevenir del treball, amb la incorporació de certs aspectes i consideracions.

El sistema recupera els avenços assolits amb el classificador d'imatges exposat a la subsecció *6.4 Classificador* per guiar la millora d'imatges que realitza el *Restormer* [11] de la subsecció *6.2 Proves amb Restormer*.

Es recupera la classe *Autoencoder* definida durant el Sprint 13, amb les seves 4 capes de codificador i el classificador, encara que ara ens interessa només el tensor comprimit, aplanat amb *flatten*. Ja no cal mapejar les característiques extretes. La recuperació de les imatges del dataset es segueix realitzant amb la classe *ImageLUTDataset*, però s'incorpora la realització d'una correspondència entre les imatges originals i les editades amb una *LUT*. Aquest pas consisteix en definir, per cada imatge modificada del DataLoader, quina és la seva imatge original. Llavors, en el suposit de que el dataset està format per una imatge original i unes altres tres imatges amb una *LUT* cadascuna, es crea una referència a la imatge original per cadascuna de les altres.

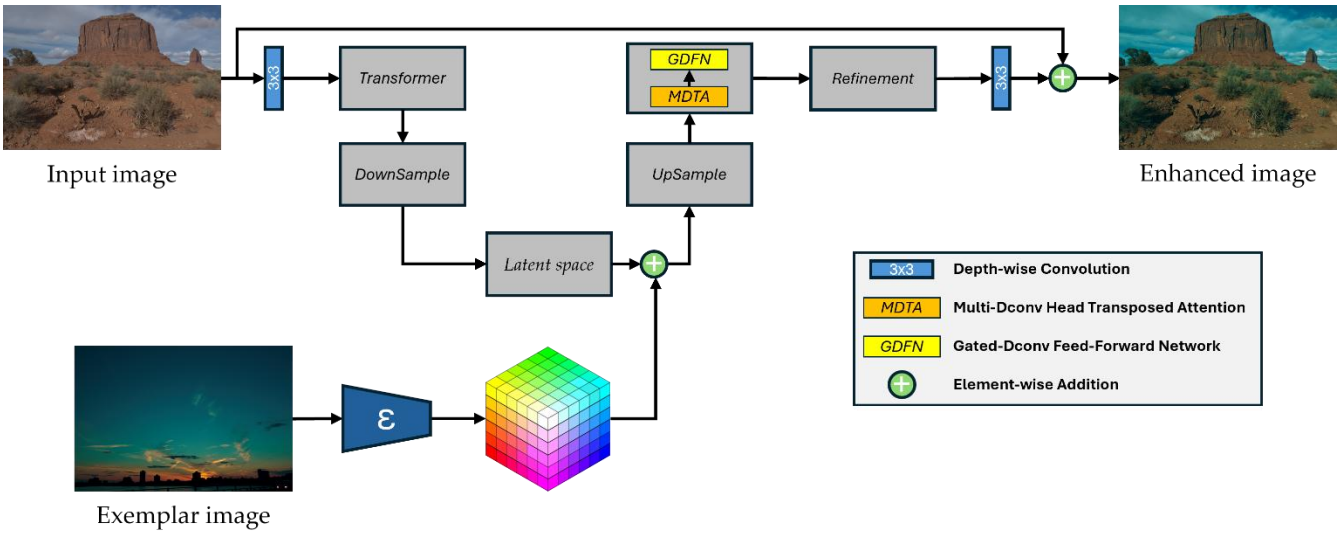


FIGURA 8: Arquitectura del Sistema d'Image Enhancement

Tal i com mostra la FIGURA 8, el model proposat segueix una estructura de *encoder* i *decoder*, així com feia el *Restormer*, començant amb una transformació de la imatge d'entrada en una representació més compacta mitjançant una convolució. Aquesta representació passa per tres blocs de *Transformer*, seguits cadascun d'ells per un nivell de *DownSampling* o de reducció de la resolució. La reducció condueix les dades fins a un quart bloc de *Transformer*, que actua com un nivell intermig corresponent al *Latent Space*, on es representen les dades de forma comprimida.

Abans de començar a decodificar les dades s'introdueixen les característiques extretes pel classificador preentrenat en etapes anteriors de desenvolupament, per tal d'orientar la reconstrucció de la imatge. El *element-wise addition* produït en l'espai intermig porta les dades cap a tres nivells de *UpSampling*, que permeten recuperar la resolució inicial alhora que es decodifiquen les dades. Després de cada reconstrucció trobem novament un bloc de *Transformer*.

A la imatge restaurada amb les caracterísques obtingudes se li apliquen refinaments addicionals abans de produïr la imatge final millorada.

El sistema empra una estrategia de normalització amb biaix, per reduir la tendència cap a valors alts o baixos i estabilitzar el model. Cada bloc de *Transformer* està format per un bloc d'*attention* anomenat *Multi-Dconv Head Transposed Attention (MDTA)*, encarregat de capturar relacions espacials entre els píxels de la imatge. Això permet entendre com els valors de regions diverses estan relacionats entre ells. Es suma un bloc anomenat *Gated-Dconv Feed-Forward Network (GDFN)*, que millora i complementa la representació de les característiques de la imatge després d'aplicar el mecanisme d'atenció *MDTA*.

L'execució del sistema consisteix en seleccionar aleatòriament la imatge original del dataset que s'introdueix a l'estructura i una imatge completament diferent a l'anterior i amb una *LUT*, que és a partir de la que s'extrauen les característiques amb el classificador per guiar la millora que es realitza. La imatge resultat produïda es compara amb una imatge de referència que es té de la imatge original però amb una *LUT*, el mateix filtre del que compta la imatge de la que extreu característiques el classificador.

Com en el desenvolupament del classificador, les imatges, deguta les seves dimensions variables, es transformen amb *CenterCrop*, prenent una dimensió de 1024x1024 píxels. El dataset es divideix novament en les parts de *train*, *validation* i *test*, amb uns percentatges del 60%, 20% i 20%, respectivament.

El model en el seu conjunt s'entrena durant 10 *epochs*. El càlcul de la pèrdua, la diferència entre la imatge reconstruïda i la de referència, es fa amb *Mean squared error (MSE)*. L'optimitzador dels pesos escollit és l'*Adam*. La següent taula recull els resultats extrets en avaluar la millora de les imatges.

Els valors de la FIGURA 9, tot i que són gaire lineals, indiquen que el sistema ha après a minimitzar les diferències entre les imatges originals i les de referència, el que permet obtenir millores d'imatges consistents.

Amb l'etapa de validació s'ajusten els hiperparàmetres del sistema i amb la prova es genera una figura en la que es mostra la imatge original, la imatge processada pel classificador i la imatge reconstruïda.

El sistema ha hagut de desenvolupar-se treballant amb el clúster del *Centre de Visió per Computador*, atès que requereix gran quantitat de VRAM i no podia fer-se en local.

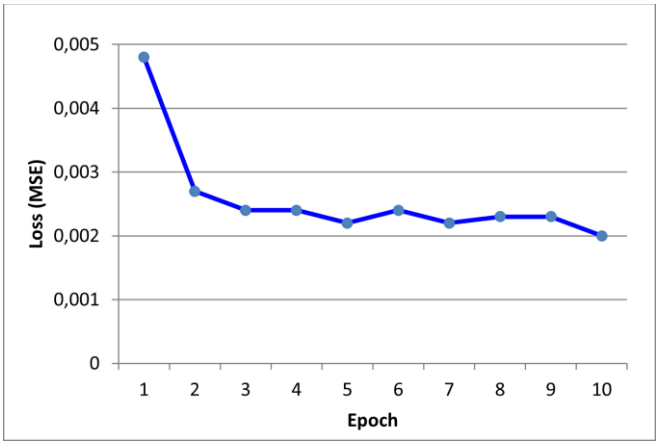
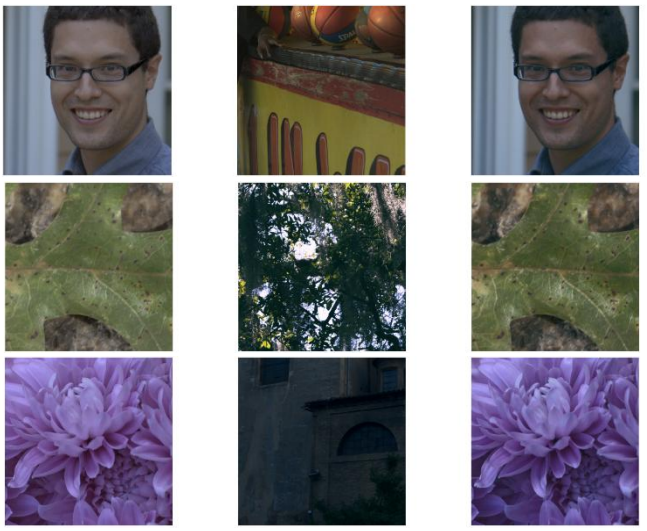


FIGURA 9: Loss function del Sistema d'Image Enhancement



Input image Exemplar image Enhanced image

FIGURA 10: Resultats del Sistema de millora d'imatges

7 RESULTATS

L'estructura de la imatge comparativa que il·lustra la FIGURA 10 permet avaluar l'evolució de les imatges al llarg del procés de millora. A la primera columna s'observen les imatges d'entrada al sistema en el seu estat original, sense cap tipus de modificació ni filtre aplicat. A la segona columna, anomenada com "Exemplar image", es mostren imatges de referència amb una LUT aplicada, que són aquelles que han sigut processades pel classificador per extraure les seves característiques per tal d'utilitzar-se com a guia per a la millora. Finalment, a la tercera columna es mostren les imatges millorades generades pel sistema proposat, evidenciant els canvis realitzats a partir de la informació que ha sigut extreta de les imatges de referència.

En quant a la qualitat, s'aprecia com la imatge millorada ha adquirit els tons de colors i de il·luminació de la imatge exemple, sense perdre els colors i les estructures de l'escena de la imatge d'entrada. S'observa, per tant, que la nitidesa és major i que el sistema ajusta els tons de la imatge original sense alterar el seu contingut. Això indica que la combinació del classificador amb l'estructura del *Restormer* ha donat peu a obtenir una reconstrucció precisa i personalitzada. Això implica, per tant, que les característiques extretes pel classificador defineixen les característiques de l'estil de les imatges, condicionant i millorant el procés del *Restormer*.

Si fem èmfasi en la FIGURA 9, a banda de confirmar la reducció de la diferència entre les imatges que ja s'ha comentat anteriorment, es veu que el model s'estabilitza en valors baixos i assoleix un aprenentatge eficient en poques *epochs*, el que evita estendre l'entrenament.

8 CONCLUSIONS

Es van definir un total de quatre objectius, tres dels quals s'han complert satisfactòriament. La investigació i implementació d'algoritmes d'extracció de característiques, com és el cas del classificador, i la seva adaptació al *Restormer* proposat per Syed Waqas Zamir i Aditya Arora ha permès obtenir resultats significatius i personalitzats. El darrer objectiu, i menys prioritari, no s'ha pogut assolir per qüestions tècniques però, sobretot, temporals.

Els resultats obtinguts mostren que el model ha estat capaç de generalitzar correctament la tasca que se l'encomanava, aplicant ajustaments estilístics sense alterar-ne el contingut original. A diferència d'enfocaments com *PieNet*, que apliquen millores de manera genèrica o que requereixen ajustaments manuals complexos, la solució proposada ha abordat aquests inconvenients amb un enfocament adaptable, integrant un classificador de característiques estilístiques que permet una personalització subjectivament més efectiva a partir d'imatges de referència seleccionades per l'usuari.

Tot i els avenços assolits, el sistema presenta certes limitacions que ofereixen futures millores. Principalment és podrien optimitzar els requeriments computacionals, ja que la complexitat del model demana un maquinari potent, el que dificulta la implementació en dispositius amb recursos limitats. Llavors, l'eficiència en el processament continua sent un punt pendent, ja que la generació d'imatges millorades encara es podria beneficiar d'una reducció en els temps d'inferència. Això bé es podria assolir mitjançant tècniques de compressió de xarxes neuronals o investigant sobre altres mètodes d'extracció i combinació de característiques.

Pel que fa les millores a futur, la incorporació d'una interfície interactiva permetria als usuaris proporcionar retroalimentació en temps real, refinant la personalització. Una altre aspecte interessant seria estendre el sistema a la millora de seqüències de vídeo, per garantir la coherència estilística entre fotogrames.

AGRAÏMENTS

Agrair principalment l'exercici realitzat pels meus tutors, Javier Vázquez Corral i David Serrano, per proporcionar-me constantment ajuda i orientació en les diferents etapes del treball, respectant els inconvenients que m'he trobat. Reconèixer la seva confiança en el meu interès i habilitats. Encomiar també el suport i els ànims rebuts per part de Jheremmy Matienzo i Oriol Cano.

BIBLIOGRAFIA

[1] H.-U. Kim, Y. J. Koh and C.-S. Kim, "PieNet: Personalized Image Enhancement Network", *Computer Vision – ECCV 2020*. Cham: Springer Int. Publishing, 2020, pp. 374–390, doi: https://doi.org/10.1007/978-3-030-58577-8_23

[2] S. Kosugi and T. Yamasaki, "Personalized Image Enhancement Featuring Masked Style Modeling", *IEEE Transactions on Circuits and Systems for Video Technology*, 2023, pp. 140-152, doi: https://doi.org/10.1109/tcsvt.2023.3285765

[3] arijit mukherjee. *PyTorch Tutorials | CNN to classify MNIST digits on Google Colab GPU*. (Sept. 23, 2018). Accessed: September 17th, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=kl3F8ILNneM&ab_channel=arijitmukherjee

[4] P. Bhavsar, "PyTorch tutorial on google colab notebook." Github.com. Accessed: Sept. 19, 2024. [Online.] Available: https://github.com/param087/Pytorch-tutorial-on-Google-colab?tab=readme-ov-file

[5] "Training a classifier." Pytorch.org. Accessed: Sept. 20, 2024. Available: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

[6] Codificando Bits. *Tutorial: ¡PYTORCH DESDE CERO!* (4 de Marzo de 2024). Accedido: 22 de Septiembre de 2024. [Vídeo Online]. Disponible: https://www.youtube.com/watch?v=QetoD5LXlEg&t=3417s&ab_channel=CodificandoBits

[7] Z. Kelta, "Introducción a las redes neuronales convolucionales (CNN)." datacamp.com. Accedido: 27 de Septiembre de 2024. [Online.] Disponible: https://www.datacamp.com/es/tutorial/introduction-to-convolutional-neural-networks-cnns

[8] Departamento de Matemática Aplicada, "05.7 Redes Neuronales Convoluciones - Introducción al Aprendizaje Automático." upm.es. Accedido: 4 de Octubre de 2024. [Online.] Disponible: https://dcain.etsin.upm.es/~carlos/bookAA/05.7_RRNN_Convoluciones_CIFAR_10_INFORMATIVO.html

[9] Alexander Amini. MIT Introduction to Deep Learning | 6.S191 (April 29, 2024). Accessed: Oct. 09, 2024. [Online Video.] Available: https://www.youtube.com/watch?v=ErnWZxJovaM&ab_channel=AlexanderAmini

[10] Alexander Amini. MIT 6.S191: Convolutional Neural Networks (May 13, 2024). Accessed: Oct. 13, 2024. [Online Video.] Available: https://www.youtube.com/watch?v=2xqkSUhmmXU&ab_channel=AlexanderAmini

[11] S. W. Zamir, A. Arora, "Restormer" Github.com. Accessed: Oct. 22, 2024 [Online.] Available: https://github.com/swz30/Restormer

[12] Alexander Amini. MIT 6.S191: Recurrent Neural Networks, Transformers, and Attention (May 06, 2024). Accessed: Oct. 25, 2024. [Online Video.] Available: https://www.youtube.com/watch?v=dqoEU9Ac3ek&t=28s&ab_channel=AlexanderAmini

[13] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan and M. Yang, "Restormer: Efficient Transformer for High-Resolution Image Restoration", 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, pp. 5718-5729, doi: 10.1109/CVPR52688.2022.00564.

[14] D. Bergmann, C. Stryker, "¿Qué es un autocodificador?" (2023, December 23). Ibm.com. Disponible: https://www.ibm.com/es-es/topics/autoencoder

[15] Marcos Conde, December 24, 2023 , "NILUT 3D LUT Dataset", kaggle.com. [Online]. Available: https://www.kaggle.com/datasets/photolab/nilut-3d-lut-dataset

[16] Xingyu, "08-AutoEncoder" Github.com. Accessed: Dec. 08, 2024. [Online.] Available: https://github.com/L1aoXingyu/pytorch-beginner/tree/master/08-AutoEncoder

[17] R. Vij. "Combating Overfitting with Dropout Regularization" towardsdatascience.com. Accessed: Jan. 02, 2024. [Online.] Available: https://towardsdatascience.com/combating-overfitting-with-dropout-regularization-f721e8712fbe

[18] D. Chuan-En Lin. "8 Simple Techniques to Prevent Overfitting" towardsdatascience.com. Accessed: Jan. 08, 2024. [Online.] Available: https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d

[19] S. B. Kang, A. Kapoor and D. Lischinski, "Personalization of image enhancement," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 1799-1806, doi: https://doi.org/10.1109/CVPR.2010.5539850

[20] J. C. Caicedo, A. Kapoor and S. B. Kang, "Collaborative personalization of image enhancement," CVPR 2011, Colorado Springs, CO, USA, 2011, pp. 249-256, doi: https://doi.org/10.1109/CVPR.2011.5995439

[21] V. Bychkovsky, S. Paris, E. Chan and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," CVPR 2011, Colorado Springs, CO, USA, 2011, pp. 97-104, doi: https://doi.org/10.1109/CVPR.2011.5995413

APÈNDIX

A1. DIAGRAMA DE GANTT

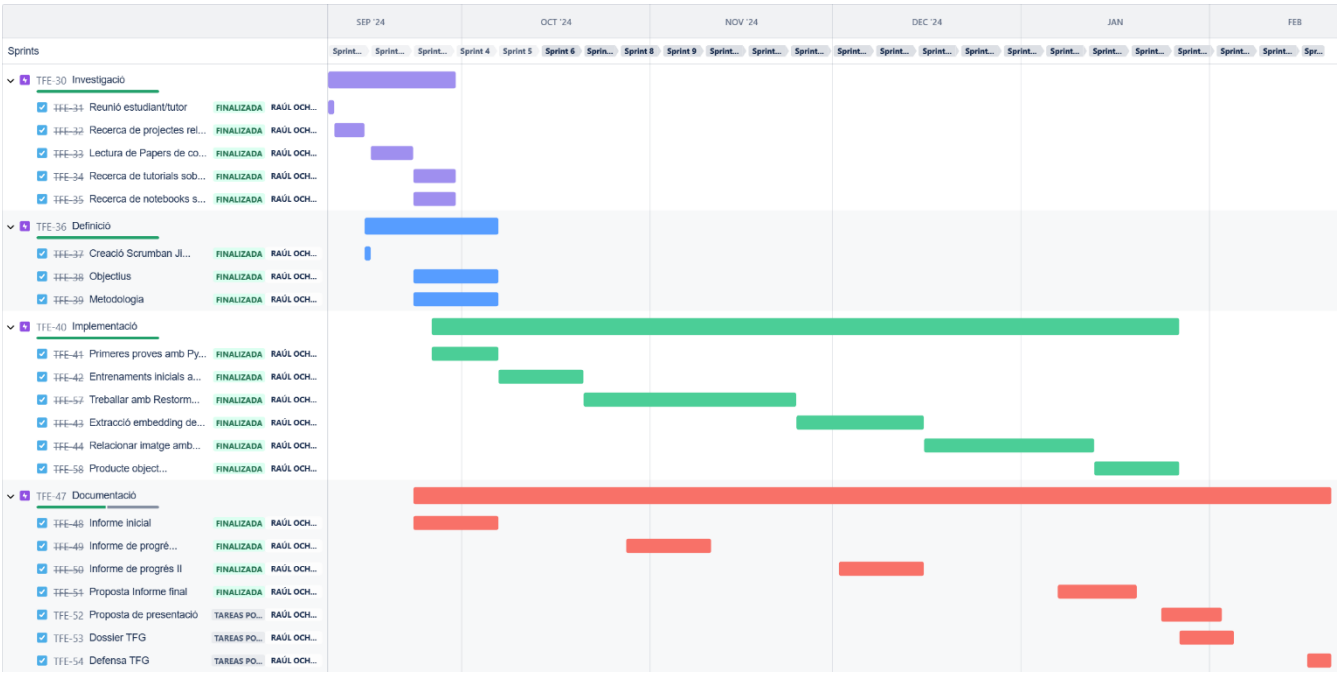


FIGURA 2: Diagrama de Gantt