



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

Herrera Valdivia, Jordi; Gutiérrez Yana, David, tut. Reconocimiento e identificación de sets de LEGO a partir de imágenes. 2025. (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/317551>

under the terms of the  license

Identificació automàtica de sets de LEGO a partir d'imatges de conjunts de peces

Jordi Herrera Valdivia

Resum—En aquest projecte s'ha dut a terme desenvolupament d'una eina per al reconeixement automàtic de sets LEGO mitjançant l'ús de tècniques d'anàlisi d'imatges i bases de dades. El procés consisteix en la captura d'imatges de peces LEGO, l'extracció de característiques com forma, color, mida i distribució de studs¹, i la comparació amb bases de dades oficials per identificar el set corresponent. El desenvolupament inclou la creació d'un pipeline que integra la segmentació d'imatges, la classificació de parts reconegudes amb mètodes per gestionar la variabilitat en la col·locació de les peces, cercant que el procés sigui adaptable a diferents escenaris. Aquest reconeixement es segueix amb un sistema d'identificació de sets originals on es té en compte errors de passos anteriors, afegint robustesa al sistema. Els resultats obtinguts mostren una precisió moderada i evidencien el potencial de l'eina proposada.

Paraules clau—Reconeixement d'imatges, Python, OpenCV, ciència de dades, identificació LEGO, classificació, segmentació d'imatges, visió per computador.

Abstract—This project follows the development for an automatic LEGO set recognition tool based on image analysis and use of existing databases. The process takes as an input images of LEGO pieces and extracts features from all parts pictured like shape, color, size and stud distribution. This info is taken and compared against an official database of sets and parts, previously curated for this specific project. The development implements a complete pipeline from image input to part recognition taking into account variability in piece orientation, looking for a working tool on as many environments as possible. These recognized pieces are forwarded to an identification system where possible errors in previous steps can be eased down, adding robustness. The results obtained show moderate accuracy and demonstrate the potential of the proposed tool.

Index Terms—Image recognition, Python, OpenCV, data science, LEGO identification, classification, image segmentation, computer vision.



1 INTRODUCCIÓ I CONTEXT DEL TREBALL

En els seus inicis al 1932, l'empresa joguina familiar LEGO va tenir els seus orígens en la fabricació de joguines de fusta fetes a mà. La persistència i evolució de la marca la van portar a començar a experimentar amb un nou concepte l'any 1946: El moldejat de plàstic per injecció. Tres anys més tard es llença al mercat la primera versió de la peça LEGO, un model primitiu sense capacitat de quedar amarrada a altres peces de manera efectiva. Aquestes es vendrien com a producte individual sense formar part d'una construcció final concreta, fet que canvia al 1955 amb la introducció del concepte LEGO System Set.

Finalment, al 1958 s'emet la patent on es modifica la peça LEGO per què ofereixi la capacitat de connexió

entre peces que mancava. Aquest mateix disseny es manté fins avui dia, fent compatibles parts de 1958 amb les fabricades al 2025. Amb el pas dels anys, els productes de la marca han traspassat la barrera de l'edat i s'han convertit en objectes de col·lecció a més de joguines.

Aquesta estabilitat i condició d'objecte de col·lecció converteix els sets produïts en un subjecte interessant per construir una eina de reconeixement. Com amb tot col·leccionable, existeix un ampli mercat de revenda per internet. És una tècnica comú vendre lots de varis sets sense identificar, pel que seria útil un mètode per saber que s'està comprant.

Aquest projecte té com a objectiu oferir una solució a aquest problema, proporcionant una eina capaç de

reconèixer i identificar sets LEGO a partir d'imatges, tant en el context de la revenda com per als col·leccionistes que desitgen catalogar la seva col·lecció.

A partir d'imatges d'entrada es retornarà l'identificador de set més probable, acompanyat d'un enllaç directe a les seves instruccions a la pàgina web oficial de la marca.

En el transcurs del document es poden trobar termes específics al vocabulari propi de LEGO. Per una major comprensió cada paraula amb aquest grau d'especificitat es trobarà marcada en la seva primera aparició i definida en un glossari.

2 ESTAT DE L'ART

Altres desenvolupadors i equips han plantejat solucions parcials al problema plantejat, ja sigui atacant únicament el problema de reconeixement de peces a partir d'imatges o la identificació de sets a partir d'un input manual.

Projectes amb un component de reconeixement d'imatges han estat utilitzats principalment utilitzat en projectes de seguiment de col·leccions de l'usuari, fent servir les imatges com a input per catalogar les peces o minifigures² detectades. Una mostra d'un projecte d'aquestes característiques és Bricksee [3], amb reconeixement de peces i relació amb sets introduïts manualment. Brickit [1] també conté una part d'identificació de peces, però enfocat a la proposta de construccions originals amb allò reconegut.

Pàgines com Brickognize [11] ofereixen reconeixement de peces, però únicament de manera individual i sense tenir en compte color. L'acompanya una API pública que realitza el mateix servei que la pàgina.

La detecció de sets també ha estat un problema ja tractat. La plataforma de creadors de MOCs³ Rebrickable [7] conté una eina d'identificació de sets. A partir de les peces que introdueix manualment l'usuari, es mostra una llista de sets que contenen parts conegudes, indicant el percentatge de completesa amb l'inventari indicat.

3 DESENVOLUPAMENT

Seguint la metodologia basada en fases, la fase 1 rep com a entrada imatges de les peces LEGO i retornarà una llista de peces reconegudes, que alimenten la següent part. La fase 2, rep la llista de la fase 1 i filtra la

base de dades creada per obtenir possibles sets de LEGO originals, idealment només un.

Les fases principals utilitzaran dos fitxers clau creats a partir d'aquests fitxers proveïts:

3.1 Tractament de la base de dades

La plataforma Rebrickable ofereix una base de dades completa de tots els productes LEGO fins 2017, a la que es pot accedir a través de diferents fitxers format .csv exportats. Es poden trobar dades aïllades de tots els colors i tipus de peces mai fabricats, en quins sets apareixen, etc. El format i nomenclatura concorden amb l'utilitzat per BrickLink [2], plataforma propietat del LEGO Group, per poder ser importable en aquesta segona plataforma.

3.1.1 Guia de colors

Existeixen 133 colors reconeixibles categoritzats en la base de dades, amb un nom, codi de color hexadecimal i flag de transparència cada un. Per una classificació més senzilla i simple de poder seguir, s'ha limitat la varietat a una paleta de 12 colors (11 reals, 1 per cas buit). A cada tonalitat de les 133 originals se li ha assignat un dels nous 12 valors. S'afegeix aquesta nova columna al fitxer .csv:



Figura 1: Paleta de colors reduïda proposada

Columna	Contingut
id	Codi identificador únic
name	Nom complet
hex	Color en format hexadecimal
is_trans	Flag de transparència
simplified	Nom del color a la paleta limitada

Figura 2: Format de la base de dades de colors

3.1.2 Guia de peces per set

Per poder fer reconeixement de sets a partir de les peces, serà important comptar amb les dades completes de cada peça de cada set, pel que serà necessari combinar la informació proveïda de diferents fitxers. Es farà servir com a base *inventory_parts.csv*, que conté per cada ID interna de set el codi de peça que hi forma part i el seu

codi de color.

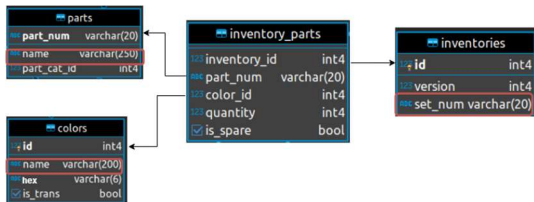


Figura 3: Taules de les bases de dades i els seus atributs seleccionats

Columna	Contingut
part_name	Nom complet de la peça
color_name	Color de la paleta simplificada
set_num	Codi general de set

Figura 4: Format de la base de dades de peces i sets

El fitxer resultant tindrà una fila per cada peça per cada set que en algun moment s'hagi produït.

3.2 Detecció de peces (Fase 1)

Com a fase d'entrada de l'eina, aquesta és la part que rep les fotografies de peces LEGO i busca donar com a resultat una llista d'objectes LegoPiece:

Atribut	Contingut
base_image	Imatge individual de la peça a color
thresholded_image	Imatge individual de la peça pasada per filtres de contrast i llindar
contours	Contorns observables obtinguts a partir de thresholded_image
centroids	coordenades representatives de cada contorn
stud_count	Nombre total de stud detectades
grid_pattern	Combinació de studs detectada
color_hex	Codi de color detectat en format hexadecimal
color_simple	Color observable identificat a

	partir del codi de color detectat
is_plate	Tipus de peça segons gruix
given_name	String amb nom col·loquial aproximat de la peça, obtingut a partir de grid_pattern, color_simple i is_plate

Figura 5: Atributs de la classe LegoPiece

El primer pas per omplir aquests camps és obtenir una subimatge de cada peça a partir d'una fotografia de varies. Això s'ha aconseguit aplicant dos filtres a les imatges input: Increment del contrast i un primer llindar. Amb l'increment de contrast permetem que cada detall de la imatge ressalti més, incloent els contorns de les peces. Un cop millorats els detalls, es detecten els contorns utilitzant un llindar adaptatiu [8] que aïlla les peces de l'entorn. Els contorns són analitzats i, a partir d'ells, s'extreuen regions de la imatge corresponents a cada peça individual.

Assignar els contorns obtinguts directament a objectes LegoPiece crearia la detecció d'imperficcions o altres detalls com a peces definitives. Per evitar aquests casos es duu a terme una reducció de soroll eliminant qualsevol contorn de menys de 500 píxels. Un cop amb dades netes, s'extreuen les coordenades dels contorns i s'extreu una versió retalla de la imatge on només es troba la peça individual.

Un cop podem iterar sobre les peces detectades en format individual, es continua amb la detecció de color en tres passos:

1. Es comença amb una detecció del codi de color mitja de la subimatge en format Hexadecimal. S'obté a partir d'un procés en dos passos on primer s'aplica un difuminat gaussià per restar impacte a ombres i reflexes, i després es crea un únic clúster amb K-Means. D'aquí extraïem el color dominant.
2. Posteriorment es fa una comparativa directe entre els 133 colors LEGO oficials presents en les bases de dades proveïdes, i s'obté el més similar.
3. A cada color oficial se li ha assignat una versió simplificada més fàcilment identificable, resultant en una paleta de 12 colors. El resultat de la detecció de color serà la versió

simplificada del color oficial identificat.

La detecció de studs, la seva distribució i el gruix de la peça requereix la diferenciació entre dues categories: Estàndard i no estàndard. S'entén com peça estàndard tota peça rectangular que es pot identificar amb una combinació de studs visible descrita en amplada per alçada:

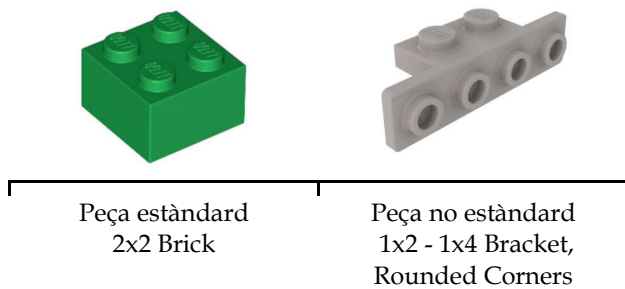


Figura 6: Diferenciació de tipus de peça

L'eina tindrà la opció de aportar dues imatges, una amb un grup de cada tipus descrit.

3.2.1 Peces Estàndard

Les peces estàndard seran tractades amb algorismes tradicionals. Per una major independència de detecció sense dependència del color, cada subimatge rep un segon procés llindar, aquest cop adaptat a la coloració general de la imatge. Aquest pas ens deixa amb una imatge binària on queden ressaltats els studs individuals i la cara frontal de la peça, aprofitant les ombres que generen.

S'apliquen una sèrie de filtres com a la primera de dos mesures de reducció de soroll en forma de erosió, per eliminar petites regions ignorables, i dilatació per ressaltar els contorns restants.

Per separar què es tracta d'un stud de la resta de components es crea un filtre a partir de les àrees relatives normalitzades a 100 dels contorns. Les àrees inferiors a 10 normalitzat seran filtrades com a soroll. L'àrea més gran (Normalitzada a 100) serà identificada com la part frontal, i els contorns restants amb àrea

pròxima a la moda seran identificats com studs:

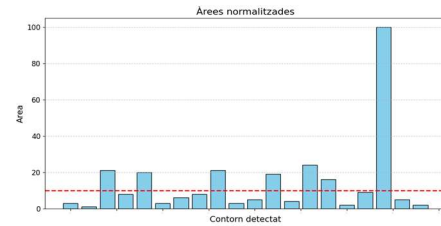


Figura 7: Representació de la reducció de soroll

Llavors, es compten els contorns de stud i s'emmagatzema a l'objecte *LegoPiece* i es calculen les coordenades dels centroides. A partir dels centroides podem aproximar una distribució dels studs amb una combinació d'amplada i alçada.

A causa de la imprecisió d'aquesta detecció, es fa una segona comprovació amb una base de dades pròpia creada a partir de totes les combinacions possibles descrites al llistat de peces públic, ordenada segons el recompte total de studs present.

L'últim pas dedicat a les peces estàndard s'enfoca en definir el gruix de la peça. Si l'espai entre la coordenada del centroide més baix i el límit inferior de la imatge és superior a 3 cops l'alçada del centroide, es considera *brick*⁴. En el cas contrari serà identificat com *plate*⁵. Aquest mètode assegura que es farà una detecció encertada independentment de l'angle de col·locació de la peça sobre el pla.

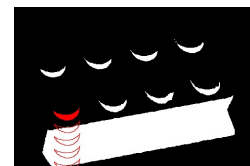


Figura 8: Representació de gruix d'un brick

Per corroborar les característiques extremes es fa ús de la API existent de reconeixement de peces individuals *Brickognize*, sense capacitat de detectar color.

3.2.2 Peces No estàndard

Les peces no estàndard seran reconegudes exclusivament per l'API pública. Les dades que ens

torna la API inclou una puntuació de confiança en la predicció. Qualsevol detecció per sota de un 75% de confiança serà automàticament rebutjada.

75% representa una puntuació que assegura que es tracta d'una peça correctament identificada, però que no rebutja resultat en casos de poca visibilitat o soroll.

La detecció de color sí es fa de manera local, amb una versió modificada de la versió per peces estàndard. S'afegeix un filtre a la subimatge corresponent a la peça el qual elimina el fons, deixant els píxels corresponents en negre (RGB (0, 0, 0)). A continuació s'aplica la detecció de color abans nombrada, però ignorant qualsevol píxel totalment negre per evitar enfosquiment per l'esborrament del fons.

3.3 Identificació de sets (Fase 2 / F2)

Rebut una llista d'objectes *LegoPiece*, aquesta segona fase pretén retornar el set d'on provenen les peces originals. La detecció de sets farà servir dues característiques principalment: *given_name* i *simple_color*. Es busca que la parella coincideixi amb la llista de peces de cada set obtingudes a partir d'una base de dades oficial.

Per considerar que *given_name* coincideix, ha de sobrepassar un *threshold* de similitud amb el nom oficial de cada peça. Per obtenir aquesta similitud s'utilitza *tokenized string matching* [4], on es separen les paraules individuals d'un string i es comparen per grups. Els colors tenen paraules determinades, pel que la comparació ha de ser exacte (obviant majúscules).

Degut a la longevitat de la marca i evolució constant de les peces produïdes, s'ha evitat utilitzar el codi identificatiu de cada peça. Moltes de les parts fabricades avui dia, encara que idèntiques visualment a les creades en dècades anteriors, contenen amb petits canvis en motlles originals o en decisions de disseny, que fan variar el codi de peça únic. Aquests canvis no són apreciables a partir d'una imatge, pel que limitar-se a

codis pot ser extremadament contraproduent.

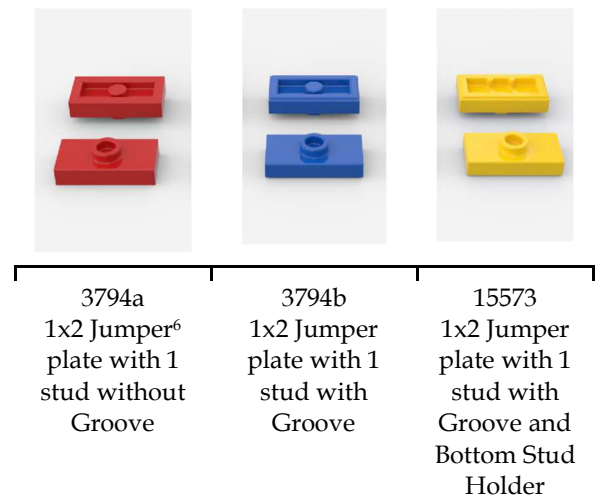


Figura 9: Peces completament diferents amb poques diferències visuals.

Per alleujar els possibles errors en la fase 1 mai es farà la cerca amb totes les peces rebudes. S'utilitza una tècnica comparable amb *cross-validation*, on es creen divisions del conjunt original, eliminant una porció en cada subconjunt. En aquest cas es fan "rondes" amb un 60% de les peces inputades, amb un mínim recomanat de 6 peces per mantenir la qualitat de la dada.

6 peces úniques representa la quantitat mínima de peces a partir de la qual no hi ha cap aportació en afegir més.

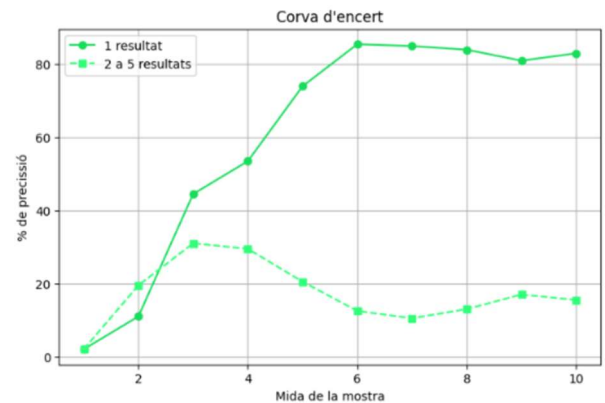


Figura 10: Estudi d'efectivitat d'afegir més peces a la fase de identificació

Aquells o aquell set amb més aparicions en aquesta fase serà considerat com a coincidència.

3.4 Presentació a l'usuari

L'eina pren la forma d'un fitxer executable amb

presentació en consola. Se li donarà la benvinguda a l'usuari amb una representació en art ASCII del text "LegoID", seguit d'unes confirmacions de la càrrega de les bases de dades necessàries pel funcionament del programa.

A continuació s'obriran dues finestres del sistema operatiu de diàleg demanant a l'usuari que seleccioni les imatges amb les peces que desitja identificar, primer peces estàndard i després no estàndard. S'informarà per pantalla en quina de les dues imatges s'està treballant en cada moment. Finalment, s'imprimeixen les peces que s'han detectat i quins (idealment només un) sets coincideixen. A més es proporcionarà un enllaç directe a les instruccions del set detectat.

4 RESULTATS

La proposta de separar el projecte en dues fases ha demostrat ser eficient tant en paral·lelització de desenvolupament com en possibilitat de testeig individual.

La fase 1 ha plantejat reptes en camps com el recompte de studs i reconeixement del color. Ha estat necessari treballar amb espectres i relacions percentuals constantment per adaptar-se a diferents resolucions. S'ha optat per utilitzar les ombres que generen les peces per fer la detecció de característiques. Això ha forçat a crear un pre-requirement a l'hora de fer les fotografies demanant un punt de vista elevat (però no zenital) i una il·luminació posterior, a més d'un fons de color el més pla possible.

La detecció de color ha necessitat post-processament per poder tenir en compte el màxim tipus de càmera possible. Depenent de la configuració de la càmera, la imatge pot tenir tendència a ressaltar algunes tonalitats o tenir filtres arbitraris de brillantor, contrast i blancor, d'entre altres. Modificar la blancor i contrast ha demostrat ser eficient per igualar les condicions de detecció.

Per l'altra banda, l'eina s'ha mostrat capaç de reconèixer la majoria de peces satisfactòriament, amb especial habilitat per aquelles més grans (2x4 i superior). Casos especials com la confusió de tonalitats grises i marrons en il·luminacions càlides han pogut resultar problemàtics, però s'han pogut resoldre en la unió de les dues fases principals.

L'enllaç entre fases ha estat subjecte de solucions

interfase. Confusions repetides com taronges i grocs o marrons i vermells s'han pogut evitar utilitzant l'espai de color LAB [9] per la comparativa del color rebut amb la paleta completa. L'espai LAB, a diferència de RGB, es basa en la percepció humana i la seva sensibilitat més alta a canvis en la brillantor que en la tonalitat en si. Aquest canvi marca la diferència en les tonalitats similars abans mencionades que es diferencien majoritàriament per la claredat.

La confusió entre blanc, gris i negre, s'han hagut d'adreçar com a cas especial. Degut a que una imatge mai contindrà negre ni blanc pur, la tendència general era classificar aquests colors com a grisos. És per això que qualsevol detecció de gris serà tractada individualment. Modificant el color obtingut per tenir la mínima saturació possible s'elimina qualsevol influència d'altres tonalitats que puguin alterar la brillantor. Un cop amb el color pur, passa per un filtre on a partir de dos umbrals es separa entre les tres possibilitats mencionades.

La fase 2, de manera aïllada, és la que mostra resultats més positius. La presència del color com a paràmetre i el mètode basat en *cross-validació* han demostrat ser útils. Aquest funcionament de l'algorisme de cerca de sets es tracta de la tercera versió implementada, iterant sobre la mateixa idea amb petites millores.

Es va iniciar el desenvolupament amb una versió 1 que assumia una llista de peces totalment correctes i no tenia en compte el color. La segona versió no millora resultats, però afegeix robustesa al tenir en compte errors en la llista d'entrada. La tercera versió i l'actual millora sobre la segona iteració afegint suport per colors.

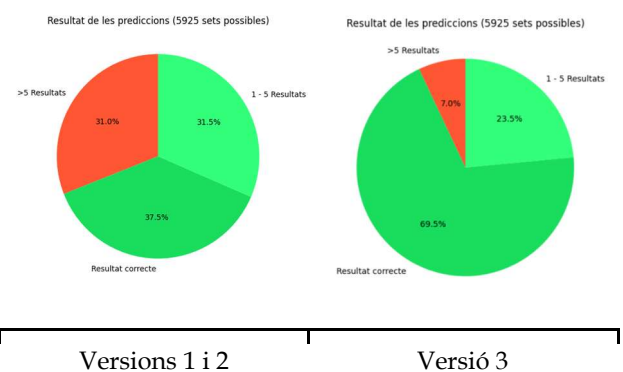


Figura 11: Evolució de la precisió de la fase de identificació

El mètode basat en una paleta reduïda de 12 colors ha

resultat funcional per mantenir classes suficientment diferenciades alhora que prou genèriques per una detecció de tonalitat més còmoda.

5 PLANIFICACIÓ

Per un desenvolupament satisfactori s'ha plantejat una organització basada en l'objectiu de paral·lelitzar el màxim de tasques possibles. El resultat ha estat una planificació que consta de dues fases principals i una de prèvia més reduïda.

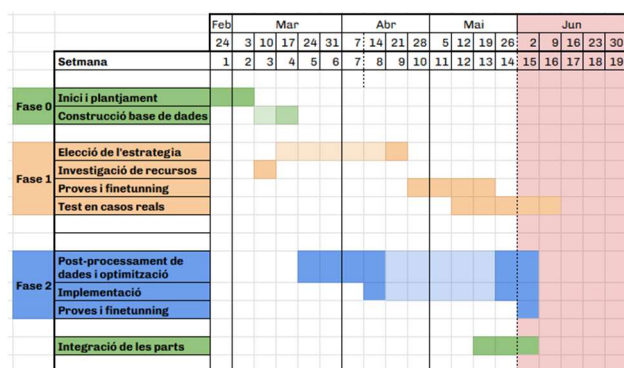


Figura 12: Diagrama de Gantt del projecte

Tot i que és inevitable que algunes parts del projecte requereix tenir passos anteriors complets, una gran porció permet treballar en dos aspectes de l'eina alhora. Les setmanes indicades en to blau representen un moviment a segon pla de les tasques indicades. La pressa d'aquestes mesures ha estat clau per poder gestionar correctament el temps de desenvolupament. Degut a dificultats en la disponibilitat de l'equip necessari pel desenvolupament de la fase 1 en alguns moments, poder tenir una planificació on encaixa poder treballar també en la fase 2 ha estat d'ajuda.

Des d'un inici s'ha cercat deixar les setmanes 15 en endavant completament lliures per poder dedicar el temps necessari a la documentació adjacent. Fita que s'ha aconseguit parcialment, sent necessàries fins a dues setmanes extres de desenvolupament de l'eina.

6 CONCLUSIONS

L'enfocament conforme a l'inici del projecte ha variat lleugerament. Tot i que no ha canviat la finalitat, diferents murs i realitats han forçat a variar mètodes. La naturalesa estandarditzada de la majoria de peces LEGO ha donat lloc a una perspectiva més tradicional des del punt de vista de metodologia. S'ha evitat una

dependència de models de machine learning com a única eina per la detecció. Sí s'ha mantingut com a comprovant i per casos concrets, però mai com a únic pas entre la fotografia i la fase d'identificació de sets.

Aquest punt de vista més basat en algorítmica ha permès transformar un dels punts febles del plantejament inicial en una fortalesa.

El repte inicial de fer front a la uniformitat de les peces LEGO, que pot complicar-ne el seu ús en models de *machine learning*, ha estat clau per fer una eina més flexible. Aprofitant que el 22% de les peces a la base de dades (les considerades estàndard) són discriminables per només 2 factors, gruix i distribució de studs, s'ha apostat fer un reconeixement manual.

Per l'altre banda en el desenvolupament ha estat necessari redefinir les premisses i objectius finals del projecte. La capacitat de detecció en conjunts desordenats és la més destacable i principal objectiu per una versió més evolucionada de l'eina. La versió actual presenta requeriments específics per al seu correcte funcionament, com ara la necessitat d'un punt de vista concret i unes condicions d'il·luminació òptimes, preferiblement amb il·luminació posterior. Aquestes limitacions marquen una direcció clara per a la millora i evolució del projecte en el futur.

El desenvolupament d'un projecte d'aquestes dimensions ha posat a prova la capacitat de prova i error i replantejament de plans inicials. Tot i que els passos generals s'han mantingut amb pocs canvis, la manera d'aconseguir resultats passos intermitjos ha evolucionat en cada prova realitzada. Un exemple clar és l'obtenció del color, va ser actualitzat d'una simple mitja en format RGB a l'aplicació de múltiples filtres per l'obtenció del RGB dominant que obvia el color de fons. Poder donar un ús pràctic a coneixements anteriors de la marca LEGO i el seu ecosistema de productes ha suposat una motivació extra per poder continuar endavant amb el projecte amb una actitud positiva.

GLOSSARI

1. **Stud:** Punts o indentacions en una peça, generalment a la part superior.
2. **Minifigura:** Figures estàndard de LEGO.
3. **MOC:** *My Own Creation*, construcció original no desenvolupada de forma oficial per LEGO.
4. **Brick:** Gruix estàndard de peça de LEGO, equivalent a 9.66 mm
5. **Placa / Plate:** Part d'un terç del gruix d'una peça bàsica amb studs visibles.
6. **Jumper plate:** Placa amb menys studs a la part superior que entrades a la part inferior.

BIBLIOGRAFIA

- [1] Brickit — Build new things from your good old bricks. (s. f.). <https://brickit.app/>
- [2] BrickLink. (s. f.). *BrickLink API | BrickLink*. <https://www.bricklink.com/v2/api/welcome.page>
- [3] Bricksee: Building made easy. (s. f.). <https://www.bricksee.app/>
- [4] *rapidfuzz.fuzz - RapidFuzz 3.13.0 documentation*. (s. f.). <https://rapidfuzz.github.io/RapidFuzz/Usage/fuzz.html>
- [5] Convert RGB color to English color name, like «green» with Python. (s. f.). *Stack Overflow*. <https://stackoverflow.com/questions/9694165/convert-rgb-color-to-english-color-name-like-green-with-python>
- [6] Er. (s. f.). *awesome-lego-machine-learning/README.md at main · 360er0/awesome-lego-machine-learning*. GitHub. <https://github.com/360er0/awesome-lego-machine-learning/blob/main/README.md>
- [7] Rebrickable API Documentation | Rebrickable - Build with LEGO. (s. f.). <https://rebrickable.com/api/>
- [8] Rosebrock, A. (2023, 9 juny). *Adaptive Thresholding with OpenCV (cv2.adaptiveThreshold)* - PyImageSearch. PyImageSearch.

<https://pyimagesearch.com/2021/05/12/adaptive-thresholding-with-opencv-cv2-adaptivethreshold/>

[9] Sharma, A. (2024, 30 novembre). *OpenCV LAB Color Space: Understanding, Advantages, and Real-World Applications*. Medium.

<https://medium.com/@alakhsharmacs/mastering-opencv-lab-color-space-understanding-advantages-and-real-world-applications-ce1bc193a328>

[10] The LEGO Group. (2024, 20 novembre). *The LEGO Group history*. <https://www.lego.com/en-us/aboutus/lego-group/the-lego-group-history?locale=en-us>

[11] Vidal, J., Vallicrosa, G., Martí, R., & Barnada, M. (2023). *Brickognize: Applying Photo-Realistic Image Synthesis for Lego Bricks Recognition with Limited Data*. *Sensors*, 23(4), 1898. <https://doi.org/10.3390/s23041898>

DADES DEL PROJECTE

E-mail de contacte: jordiherrera.hv@gmail.com

Menció realitzada: Computació

Treball tutoritzat per: David Gutiérrez Yana

Curs 2024/25