Entrenar un model basat en intel·ligència artificial i “Machine Learning” requereix de molts recursos hardware per aconseguir una velocitat d’entrenament òptima. Després d’investigar diverses opcions per solucionar aquest inconvenient, es va decidir utilitzar Google ColabNotebooks que és un full de Python amb l’execució remota en servidors de Google amb la possibilitat d’utilitzar GPUs o TPUs [].

A més, aquests fulls de Python permeten incloure totes les llibreries necessàries per dur a terme projectes d’intel·ligència artificial, com la llibreria FastAI que és la que s’ha utilitzat []. Aquesta llibreria permet definir un conjunt d’imatges com a model d’entrenament i un altre conjunt per validar el model resultant. També permet millorar els resultats de l’entrenament aplicant diferents mesures a les imatges com per exemple, rotació, deformació o canvis de mides de les imatges, entre d’altres.

L’objectiu principal de realitzar la creació d’aquest model és que, gràcies a la intel·ligència artificial i al “Machine learning”, es pogués crear un petit script en Python que identifiqués a quin producte correspon una imatge donada utilitzant el model exportat després de l’entrenament. Aquest script retorna el TOP 3 de coincidència amb els productes del model.

El primer pas del procés ha consistit en generar una base de dades amb 138 productes. El nom d’aquests productes han sigut extrets de “residuonvas.cat” i s’han cercat per Internet 50 imatges representatives de cadascun. **A la figura 1** es pot observar l’estructura dels directoris per l’entrenament del model. D’aquestes 50 imatges, el 80% s’han utilitzat per entrenar el model de forma que busqui similituds i les etiqueti amb el nom del directori corresponent. L’altre 20% s’ha utilitzat per validar el model i millorar-lo. Per poder entrenar al model, s’ha començat a partir d’un model genèric de xarxa neuronal entrenat prèviament (resnet152 []), de forma que els resultats fossin més acurats amb una millor taxa d’èxit.

Un cop s’inicia el procés d’entrenament es van realitzar “loops” anomenats èpoques a tot el conjunt de les imatges. A mesura que van passant les èpoques, les imatges s’apropen més a una etiqueta que una altra i el model comença a entendre que aquell tipus d’imatge correspon a una etiqueta. Després de cada època, s’utilitza el model de validació per comprovar si realment el model ha après correctament com assignar les etiquetes. El resultat d’aquesta assignació per cada època correspon a la taxa d’error (error\_rate). **A la figura 2** es pot observar el resum de les èpoques i els valors finals obtinguts. Després de X èpoques, el resultat ha sigut satisfactori ja que tant la pèrdua d’informació del model d’entrenament (train\_loss) com la del model de validació (valid\_loss) és baixa, al voltant del X% i la taxa d’error (error\_rate) no supera el X%. Després de l’entrament i abans d’exportar el model s’ha emprat una eina que inclou la llibreria FastAI que consisteix en la creació d’una matriu de confusió del model (**figura 3**). En aquesta matriu es pot comprovar que el model està prou bé entrenat i que hi ha escasses confusions amb el conjunt d’imatges proporcionat inicialment.

Per tant, en forma de resum, un exemple general seria el següent: El producte identificat amb l’etiqueta ‘0’ té 50 imatges. Aquest producte correspon a una ampolla d’oli. D’aquestes 50 imatges, 40 s’han utilitzat per entrenar el model aplicant-hi tècniques de rotació i establint la mida de la imatge en 224px per costat, de forma que totes les imatges tinguin relació 1:1, i les 10 restants s’han utilitzat per validar el model. Després de l’entrenament, es pot comprovar **a la mateixa figura 3** que la confusió és nul·la i que per tant el model ha après correctament a identificar aquest producte. Ara amb el model exportat, es pot enviar una imatge d’una ampolla d’oli a l’script i aquest ha de retornar com a major coincidència l’etiqueta ‘0’.

Després de realitzar tot aquest procés durant moltes hores, s’ha aconseguit un bon resultat amb una baixa taxa d’error, i per tant s’ha exportat el model i s’ha emmagatzemat en el servidor del sistema.

Per últim, en el servidor s’ha emmagatzemant l’script en Python realitzat que permet comparar una imatge amb el model exportat. Per poder executar aquest script, el servidor executa la següent comanda i rep el resultat de la forma que es mostra en **la figura 4.**

Aquest resultat s’envia a la aplicació que és qui sap identificar i relacionar les etiquetes amb els noms dels productes.