

Juanma Cámara Díaz  
Pol Espinasa Vilarrosa  
Daniel Suárez Valverde

# Repte BonÀrea

## Grup Miguel

**bon****rea**

<b>Objectiu</b>	<b>3</b>
<b>Descripció, plantejament i metodologia</b>	<b>3</b>
<b>Explicació de les tasques realitzades</b>	<b>4</b>
Entrenament de la xarxa neuronal	4
Desenvolupament de l'aplicació mòbil	5
<b>Interfície que veu l'usuari al usar l'aplicació</b>	<b>5</b>
<b>Eines</b>	<b>6</b>
<b>Llenguatges de programació</b>	<b>6</b>
<b>Problemes sorgits durant el desenvolupament</b>	<b>7</b>
<b>Possibles millores</b>	<b>7</b>

# Objectiu

En aquest repte, es busca la implementació d'un sistema automatitzat de detecció de forats als prestatges de les neveres dels supermercats BonÀrea amb les càmeres de seguretat.

La intenció del nostre projecte és desenvolupar una aplicació mòbil disponible per tots els treballadors dels supermercats que els avisi la manca d'algun producte en alguna d'aquestes neveres, i així poder saber quan reposar els productes...

## Descripció, plantejament i metodologia

El sistema funciona mitjançant una xarxa neuronal anomenada *Yolo v.3* la qual comprova les imatges en temps real i les etiqueta buscant els forats dels prestatges de les neveres. Aquesta guarda les coordenades on estan aquests buits, per posteriorment enviar aquesta informació a l'aplicació mòbil. L'aplicació llegeix les dades rebudes i conta quants forats hi ha i a quina zona de la nevera estan. Posteriorment llença una notificació a l'usuari indicant a quina nevera hi ha mancances de productes, en quina quantitat i a quina zona estan aquestes mancances.

Com que les imatges s'analitzen en temps real, si un empleat rep una notificació d'un buit i l'omple, automàticament a l'aplicació desapareixerà la notificació i marcarà altre cop la nevera com a plena.

# Explicació de les tasques realitzades

## Entrenament de la xarxa neuronal

Per entrenar la xarxa neuronal que analitza les imatges de les càmeres de seguretat, hem utilitzat la xarxa anomenada Yolo en la seva versió número 3. Per començar hem considerat que les 44 imatges proporcionades se'ns podien quedar escasses a l'hora d'entrenar la xarxa neuronal, així que mitjançant un script en python que hem desenvolupat, hem aplicat diferents filtres i hem manipulat les imatges per generar-ne més, arribant a tenir 571 imatges que han estat etiquetades manualment, marcant cada buit en les neveres amb un software anomenat "Vott".

Posteriorment hem entrenat la xarxa neuronal amb "Venv" una eina desenvolupada amb python dissenyada per la creació d'entorns virtuals lleugers. Aquesta mitjançant un script de python de la pròpia xarxa neuronal, usa la GPU de la targeta gràfica per entrenar-se, aprenent els patrons i la informació de les imatges etiquetades anteriorment.

```
Epoch 00054: ReduceLROnPlateau reducing learning rate to 9.999999974752428e-08.  
251/251 [=====] - 157s 626ms/step - loss: 23.3826 - val_loss: 26.6523  
Epoch 55/78  
251/251 [=====] - 177s 704ms/step - loss: 23.9942 - val_loss: 26.1527  
Epoch 56/78  
251/251 [=====] - 158s 628ms/step - loss: 23.4527 - val_loss: 25.4502  
Epoch 57/78  
251/251 [=====] - ETA: 0s - loss: 23.3789  
Epoch 00057: ReduceLROnPlateau reducing learning rate to 1.0000000116860975e-08.  
251/251 [=====] - 157s 626ms/step - loss: 23.3789 - val_loss: 26.6979  
Epoch 58/78  
251/251 [=====] - 157s 626ms/step - loss: 23.5094 - val_loss: 26.6807  
Epoch 59/78  
251/251 [=====] - 156s 623ms/step - loss: 23.4992 - val_loss: 26.3263  
Epoch 60/78  
251/251 [=====] - 158s 628ms/step - loss: 23.1176 - val_loss: 24.9612  
Epoch 61/78  
251/251 [=====] - 155s 618ms/step - loss: 23.7752 - val_loss: 27.1937  
Epoch 62/78  
251/251 [=====] - 150s 597ms/step - loss: 23.3854 - val_loss: 25.6035  
Epoch 63/78
```

*Captura de pantalla durant un dels intents d'entrenament de la red neuronal*

L'entrenament ha constatat de 27 èpoques inicials, durant les quals ha entrenat les capes bloquejades, és a dir ha après unes bases per poder dur a terme més endavant un entrenament més complex i més profitós. Posteriorment ha dut a terme un entrenament que ha constatat 51 èpoques, el qual ha sigut més exhaustiu, ja que ha usat les capes desbloquejades. Aquest procés ha durat unes 7 hores degut als errors que han anat sorgint durant l'entrenament. El procés simplificat un cop els errors han estat simplificats, ha sigut de dues hores aproximadament. El resultat ha

sigut sorprenentment bo, ja que la taxa d'errors ha estat quasi nul·la. Per poder seguir comprovant la fiabilitat de la xarxa, hem editat manualment amb Photoshop algunes de les 44 imatges inicials, afegint productes a llocs on hi havia un buit, o eliminant productes d'alguns llocs per generar buits. Els resultats han seguit sent sorprenentment bons pel poc entrenament del qual ha disposat la xarxa.

## Desenvolupament de l'aplicació mòbil

Un cop hem tingut una xarxa neuronal entrenada i perfectament funcional, hem procedit amb el desenvolupament de l'aplicació mòbil que haurien d'utilitzar els empleats del supermercat per tenir constància d'on falten productes.

L'aplicació ha sigut desenvolupada utilitzant l'entorn de desenvolupament “*Android Studio*”.

Inicialment s'ha desenvolupat el frontend de l'aplicació, començant per la part que no es visible a l'usuari, les classes i funcions, on bàsicament s'ha implementat 6 classes, la “APICheck”, la qual és l'encarregada d'actualitzar constantment la informació rebuda, el “MainActivity”, bàsicament s'encarrega del funcionament de l'app en general. Dues classes enfocades als prestatges, la “Adabter” la qual conté informació de tots els prestatges i la “View” la qual conté la informació que es mostra per pantalla a l'usuari. A continuació tenim la “TipolIncidencia”, classe que s'encarrega de gestionar el símbol d'alerta que apareix si hi ha alguna mancança de stock. I finalment la “SplashActivity” classe que s'encarrega de generar la pantalla d'inici de l'aplicació.



Un cop implementat el front-end hem prosseguit amb la part del servidor back-end. El servidor ha sigut desenvolupat amb JavaScript utilitzant el motor Node JS, un motor pensat per córrer JS des del costat del servidor. A més a més utilitzem la llibreria “express” per habilitar una funcionalitat de “post” i una de “get” per tal de poder enviar i rebre informació del servidor.

*Interfície que veu l'usuari al usar l'aplicació*

# Eines

Per la realització del projecte hem utilitzat una varietat diversa de programes tant per l'aplicació mòbil com per la xarxa neuronal i el servidor.

**Vott:** Vott és una aplicació electrònica de codi obert i gratuït per a l'anotació i l'etiquetatge d'imatges desenvolupada per Microsoft. L'hem utilitzat per etiquetar tots els espais buits de les imatges proporcionades, a més d'una gran quantitat editades, és a dir, girades en diferents angles, per tenir una base de dades més ampla a l'hora d'entrenar al nostre programa.

**Android Studio:** Android Studio és l'entorn de desenvolupament integrat oficial per a la plataforma Android. L'hem empleat per a la creació de l'aplicació que va instal·lada al telèfon i que mitjançant un model client-servidor rep les dades de si s'ha de reposar o no. Està pensada perquè tots els empleats del supermercat la portin a sobre i puguin veure al moment i en temps real les prestatgeries.

**Photoshop:** Adobe Photoshop es un editor de fotografies desenvolupat per Adobe System Incorporated. És una eina que hem utilitzat per editar algunes de les imatges afegint i traient aliments per tenir altres dades amb les que poder comprovar si la nostra xarxa funcionava.

**Postman:** Postman és una plataforma que afegeix eines per simplificar la construcció API, molt útil a l'hora de testejar el nostre servidor.

**Node:** Node.js és un entorn en temps d'execució multiplataforma de codi obert, per a la capa de servidor, basat en el llenguatge de programació JavaScript. L'hem utilitzat per a la creació d'un servidor rest on rebrà les dades que mostrarà per pantalla.

## Llenguatges de programació

**Python:** Python és un llenguatge de programació multiparadigma, ja que suporta orientació a objectes, programació imperativa i programació funcional, i és el que hem utilitzat per a la xarxa neuronal.

**Java:** Java és un llenguatge de programació molt utilitzat i una plataforma informàtica la qual hem emprat en l'aplicació.

**JavaScript** Java Script és un llenguatge de programació interpretat, que està orientat a objectes. Està molt enfocat en el '*scripting*' i l'hem utilitzat en el servidor.

## Problemes sorgits durant el desenvolupament

Durant el desenvolupament del projecte ens han sorgit diferents problemes. L'inconvenient més evident que hem patit ha sigut la manca de recursos dels quals hem disposat. Entre ells, les poques imatges que se'ns han proporcionat, amb una quantitat més elevada d'imatges i més variada hauríem pogut entrenar la xarxa neuronal d'una forma més eficient i exhaustiva, aquest inconvenient, però, l'hem pogut solucionar amb l'script esmentat prèviament per manipular les imatges i crear varietat. Per altra banda la capacitat dels nostres ordinadors ens ha limitat, ja que no hem disposat de GPU's gaire potents per poder entrenar la xarxa neuronal i només hem pogut fer els entrenaments i les proves amb un dels ordinadors, el qual tampoc era molt potent, però suficient per fer la tasca encara que fos a un ritme més lent. A part d'aquests problemes, no hi ha hagut gaires inconvenients més i si els hi ha hagut han sigut petits detalls que s'han pogut resoldre fàcilment amb una mica de recerca.

## Possibles millores

Durant el desenvolupament del projecte se'ns ha anat acudint diferents millores que es poden implementar per fer aquest sistema de detecció automàtica de l'estoc molt més eficient, però que no hem pogut afegir per falta de temps o de material i equipament.

Una de les possibles millores podria ser el canvi d'imatges. Intentar aconseguir imatges de més qualitat i detall i amb un angle més frontal, per així no només poder veure si hi falta estoc, si no comprovar també, quin producte falta i exactament la posició on falta.

A més a més es podria dur a terme una implementació en l'àmbit global a totes les prestatgeries de tots els supermercats usant diferents càmeres, per tenir així controlat tot l'estoc i no només el d'una nevera de carn en concret.