# Anàlisi semàntic d'imatges en xarxes socials amb tècniques de deep learning

Marc Reyes Martí

**Resum–** Aquest projecte proposa un mètode per a classificar diferents imatges segons quins aliments es poden veure en elles. S'utilitzen dos conjunts d'imatges etiquetades segons quin aliment es veu en elles, un dividit en 101 classes i l'altre en 256 classes. Utilitzant aquests conjunts s'analitzen diferents tècniques d'entrenament de xarxes neurals convolucionals per a generar un model capaç de distingir quin aliment si pot visualitzar. S'analitzen diferents arquitectures de xarxes convolucionals que han obtingut bons resultats amb el dataset Imagenet, com per exemple la ResNet. El model proposat pel conjunt de dades dividit en 101 classes assoleix un 81,59% i el proposat pel conjunt de dades dividit en 256 classes assoleix un 63,21%.

**Paraules clau–** Aprenentatge profund, xarxes neurals, xarxes neurals convolucionals, reconeixement de menjar, mapa d'activació de classes, aprenentatge automàtic

**Abstract–** This project proposes a method for classifying different images depending on what foods we can see in the pictures. Use two sets of images tagged as what food is in them, one divided into 101 classes and other divided into 256 classes. We using these sets to analyze different techniques of training convolutional neural networks to generate a model able to distinguish which food if viewable. We analyze a differents convolutional network architectures that have obtained good results with the dataset Imagenet, one example is ResNet. The proposed model for dataset that is divided into 101 classes has obtained a 82.07% and the proposed model for dataset that is divided into 256 classes has obtained a 63.54%.

**Keywords–** Deep learning, neural networks, convolutional neural networks, food recognition, Class Activation Mapping, machine learning

## 1 INTRODUCCIÓ

En les xarxes socials, com poden ser facebook, instagram, twitter o Flickr, cada dia milions d'usuaris de tot el món comparteixen milions de fotografies en les que podem trobar moltes imatges d'aliments i que ens han servit per poder portar a terme el nostre projecte. Utilitzant mètodes d'aprenentatge computacional es pot extreure informació del que surt en una imatge, per exemple podem classificar si en una imatge surt una bicicleta o no.

E-mail de contacte: marc.reyesma@e-campus.uab.cat
Menció realitzada: Computació
Treball tutoritzat per: Jordi Gonzàlez (CVC-UAB)

La finalitat d'aquest treball és poder crear una tecnologia que serveixi per a detectar trastorns alimentaris, amb la intenció de poder realitzar informes dietètics sobre els costums alimentaris de la població catalana i/o turistes.
Per a poder extreure informació dels costums alimentaris de la població es poden utilitzar les diverses imatges que la gent comparteix a les seves xarxes socials; i tenint en compte que l'aprenentatge computacional ens serveix per a poder treure informació d'una imatge ens podem plantejar si d'un conjunt d'imatges d'aliments podríem arribar a ser capaços de classificar-les segons l'aliment que es pot visualitzar en elles. En el cas de ser capaços de classificar quin aliment es pot visualitzar en una imatge, pot servir en molts àmbits diferents. Per

exemple, si tenim un empresari que vol crear un negoci d'aliments, pot agafar un conjunt molt gran d'imatges de diferents xarxes socials i veure quins són els menjars que interessen més a la gent segons el que comparteixen a les seves xarxes socials.
Un altre exemple d'ús, és trobar persones amb problemes alimentaris analitzant les imatges que comparteixen a les xarxes socials. Per a reconèixer cadascun dels aliments que es poden visualitzar a la imatge utilitzarem un algoritme de machine learning conegut com deep learning, que permet crear un model utilitzant un conjunt d'entrenament i ens indica quina és la probabilitat de que una imatge concreta pertanyi en cadascuna de les diferents classes en les que es divideix el conjunt d'entrenament.

## 1.1 Objectius

El principal objectiu d'aquest treball és classificar les imatges d'una base de dades de menjar segons quin aliment si pot veure en elles. Per assolir aquest principal objectiu el que farem serà crear un model utilitzant deep learning que ens serveixi per a poder classificar les fotos compartides en les xarxes socials, segons quins aliments apareixen. Aquestes imatges fetes per usuaris de xarxes socials tindran una variabilitat en il·luminació, punt de vista i aparença, fent necessari l'ús de models complexos, com són les xarxes neuronals.
Per tant, a partir de l'objectiu principal s'han definit 3 subobjectius:

1. Aprendre a fer servir el framework Caffe per tal de dissenyar i entrenar un model d'una xarxa convolucional per a resoldre el problema pel que ha estat dissenyada.

2. Utilitzar diferents CNN amb la finalitat de trobar quines ens donen una millor precisió i amb quines circumstàncies.

3. Entendre les CNN punteres del mercat amb la finalitat de donada una CNN qualsevol poder entendre el seu funcionament. Per a fer-ho mirarem codi lliure i articles publicats en les universitats més prestigioses del món.

## 1.2 Estructura del document

Primerament, s'explicarà el problema a resoldre i quines possibles solucions podem trobar. També s'explicarà el framewok caffe i una breu introducció de les xarxes neurals i les xarxes convolucionals, acabant la part teòrica amb una breu descripció de les CNN utilitzades. Seguidament, es poden veure els resultats obtinguts amb les CNN utilitzades i els resultats obtinguts utilitzant CAM, seguit per les conclusions del projecte. Al final de l'article podem trobar les referències utilitzades i un apèndix amb imatges.

## 1.3 Estat de l'art

En el treball s'han utilitzat dos datasets diferents i recerquem els millors resultats de precisió obtinguts per altres persones utilitzant els mateixos datasets.
El millor resultat obtingut utilitzant el dataset food-101 és amb una precisió del 77,4% [16], per a obtenir aquesta precisió van utilitzar finetuning utilitzant una xarxa pre-entrenada amb el dataset Imagenet, la xarxa utilitzava l'arquitectura GoogleNet i per obtenir el model es va fer un aprenentatge de 250.000 iteracions.
Seguidament, mirem l'estat de l'art de xarxes entrenades amb Food-256. El millor accuracy trobat, sense utilitzar Fisher Vector(FC), és de 54,7% [16] i s'ha obtingut utilitzant l'arquitectura GoogleNet, aplicant finetuning amb una xarxa entrenada amb Imagenet i un entrenament de 72.000 iteracions.
Utilitzant food-256 i CNN hem trobat millors resultats que els esmentats anteriorment, aquest resultat es d'un 63,77%, per a obtenir aquest accuracy s'ha utilitzat la xarxa AlexNet, s'ha aplicat finetuning d'una xarxa entrenada amb 2000 categories i s'ha utilitzat Fisher Vector (FV). Aquest mateix grup d'investigació també va obtenir un 77,35% utilitzant el dataset food-101 i utilitzant la mateixa metodologia que pel dataset food-256.

# 2 El problema a resoldre

El problema a resoldre és classificar imatges segons quin és l'aliment principal que es visualitza en la imatge. Crearem models que ens serviran per a classificar quin aliment es visualitza a la imatge. Per a resoldre el problema ens trobem dos entrebancs principals, el primer és que es necessiten moltes imatges per a fer l'aprenentatge i el segon que el temps d'aprenentatge és molt elevat.

## 2.1 Material relacionat

Anteriorment diferents grups d'investigació s'han trobat amb el mateix problema que ens trobem nosaltres, es a dir, classificar imatges segons quin aliment es pot visualitzar en cadascuna d'elles. Per a resoldre'l han utilitzat tècniques d'aprenentatge computacionals diferents. De les tècniques utilitzades ni han que són globals i altres de locals. Les tècniques globals són aquelles que es fixen en tota la imatge i les locals es fixen en zones determinades de la imatge. De mètodes globals utilitzats trobem el BOW (Bag-of-Words Histogram)[19] i el IFV (Improved Fisher Vectors)[20]. Els mètodes locals utilitzats són RF (Random Forest Classification)[21], RCF (Randomized Clustering Forests)[22] i MLDS (Mid-Level Discriminative Superpixels)[23]. Dels mètodes nombrats anteriorment el que dona més bons resultats és MLDS.

## 2.2 Com resoldre'l

Per a resoldre el problema d'identificar quin aliment es visualitza a cada imatge, hem utilitzat Convolutionals Neurals Networks(CNN). Les CNN donen millors resultats que altres tècniques d'aprenentatge computacional per la seva alta capacitat per aprendre tinguent una gran quantitat d'inputs o imatges d'aprenentatge. Mirant l'estat de l'art obtingut i veient els resultats obtinguts utilitzant altres tècniques de computació amb el mateix dataset, podem observar que les CNN són la millor tècnica per el nostre problema. Les CNN intenten replicar l'estructura d'un cervell per a obtenir un model molt complex que classifiqui les nostres imatges.
Hem buscat diferents arquitectures de CNN de les millors universitats del món per a obtenir un model el més precís possible, que ens serveixi per a classificar les nostres imatges. Fent un aprenentatge de poques iteracions mirarem quina es l'arquitectura que ens dóna millors resultats.

## 2.3 Datasets utilitzats

En aquest projecte s'han utilitzat dos conjunts d'imatges. Aquests conjunts d'imatges s'anomenen food-101[1] i food-256[2], són uns conjunts d'imatges prèviament classificades segons quin és l'aliment que es pot visualitzar en cadascuna de les imatges. En el cas del food-101 podem trobar imatges de menjar classificades en 101 classes i en el food-256 trobem imatges de menjar classificades en 256 classes. Cada classe correspon a un menjar diferent. Alguns exemples de classes en el food-101 són les següents: apple pie, beet salad, chocolate mousse, ... i en el dataset food-256 algunes de les classes que podem trobar són les següents: rice, sushi, chicken rice, ... El dataset food-101 està compost per 101.000 imatges dividides en 75.750 imatges per l'aprenentatge i 25.250 imatges pel test. El food-256 està compost per 28.280 imatges dividides en 22.526 imatges per l'aprenentatge i 5.754 imatges per test.

# 3 CAFFE

Caffe és un framework de deep learning ràpid i modular que ha estat desenvolupat per Berkeley Vision and Learning Center (BVLC) i contribuents de la comunitat. Caffe utilitza diferents llenguatges com són C++, Python i Matlab, també s'utilitza CUDA per a poder executar en la GPU. Per exemple, per modificar les imatges d'entrada a la xarxa es modifica un .cpp escrit en llenguatge c++.
Caffe té diferents característiques com són:

- Arquitectura expressiva que fomenta l'aplicació i la innovació.
- Permet fer l'aprenentatge del model amb GPU o amb CPU.
- És un framework que treballa molt ràpid, permet processar 60M d'imatges per dia utilitzant una GPU NVIDIA K40, el que venen a ser 1ms/imatge quan es fa test i 4ms/imatge quan es fa aprenentatge.
- Caffe s'utilitza en molts camps relacionats amb el reconeixement com són la visió, la parla o multimèdia.

Si comparem Caffe amb altres frameworks destinats a deep learning podem trobar algunes diferències. No trobem cap altre framework que permeti escollir si utilitzar GPU o CPU i que també permeti utilitzar models pre-entrenats.

Caffe es diferencia d'altres CNN frameworks en 2 punts principals:

- L'aplicació està feta completament amb C++, el que facilita la intregració amb sistemes i interfícies fetes amb C++. L'ús de la CPU ens permet fer experiments amb un model ja entrenat de manera més simple.
- L'ús del finetuning permet un aprenentatge més ràpid per poder començar l'aprenentatge amb una xarxa amb pesos interessants i que no són inicialitzats de manera aleatòria.

# 4 CONVOLUTIONAL NEURAL NETWORK

Una Convolutional Neural Network tracta d'extreure les característiques d'una imatge com són les formes o textures. Les CNNs a diferència d'una xarxa neuronal són explícitament utilitzades per les imatges. Les xarxes neuronals no funcionen bé per imatges amb molta resolució[14]. En utilitzar la CNN a diferència de les xarxes neuronals tenim molts menys paràmetres, el que fa que l'aprenentatge sigui molt més ràpid. Però, tenim suficients paràmetres com per extreure característiques de les imatges i en les últimes capes relacionar aquestes característiques entre elles per a decidir a quina classe pertany cada imatge.
Les primeres capes s'encarreguen de detectar contorns i establir patrons de detecció dels contorns, les capes posteriors el que busquen en les imatges són patrons més complexes i en diferents posicions de la imatge. Per posar un exemple, les primeres capes poden buscar si la imatge té un cercle amb punxes, i les capes posteriors poden buscar si aquestes punxes estan situades a la part inferior dreta de la imatge; i finalment utilitzant tots els patrons trobats i totes les característiques trobades es farà una suma ponderada de cadascuna d'elles per arribar a un valor final

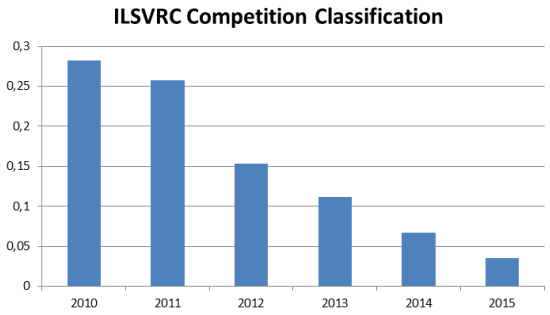


Figure 1: Resultats de la competició ILSVRC, error Top-5 dels anys 2010[6], 2011[6], 2012[14], 2013[7], 2014[7] i 2015[7].

per a cadascuna de les classes. Utilitzant aquests valors la classe que tingui el valor més alt significa que la imatge pertany a ella.
Utilitzar molts paràmetres ens permet no fixar-nos en la imatge en concret, com per exemple si surt una pilota o no, sinó que ens permet fixar-nos en la textura i en la forma del que es pot veure en la imatge. També ens permet aprendre a relacionar una característica amb una altre, el que vol dir que si en una imatge surten dos o més característiques directament es classifiqui la imatge com una classe en concret.
Les xarxes convolucionals tot i ser una tecnologia molt nova va començar a desenvolupar-se fa molts anys. El primer fet relacionat amb les CNN va succeir al 1958 quan Rosenblatt va proposar els perceptrons [3]. El següent fet important va ser al 1986 quan Rumelhart va presentar Multilayer perceptrons i backpropagation [4]. La tecnologia no va passar a ser interessant per les grans companyies fins l'any 2012 en el que es va presentar a la ILSVRC l'arquitectura Alexnet creada per Alex Krizhevsky, Ilya Suskever i Geoffrey Hinton, i que va guanyar la competició ILSVRC (ImageNet Large-Scale Visual Recognition Challenge) de l'any 2012, entrenada i testejada amb el dataset Imagenet [5]. L'arquitectura va obtenir un error amb top-1 d'un 37,5% i amb top-5 d'un 17,0%. Aquest fet va propiciar que l'any següent a la competició es presentessin molts més grups per intentar guanyar el certamen. A partir d'aquest fet cada any s'han creat noves arquitectures com són ZF Net (2013), VGG Net (2014), GoogleNet (2015) i Microsoft ResNet(2015). Les grans companyies han dedicat molts esforços per a crear les seves pròpies arquitectures i intentar guanyar la competició ILSVRC.
Mirant l'error obtingut pel guanyador de la competició ILSVRC desde l'any 2010 a 2015, que es pot veure en la Figura 1, podem afirmar que des de l'any 2012 les CNN han agafat molta importància i cada vegada són arquitectures més potents que permeten reduir el top-5 cada any i de manera significativa des del 2012.

Una CNN està composta especialment per a 3 tipus diferents de capes:

- Capa convolucional: aquesta capa aplica una convolució. La convolució tracta de donat un input (la imatge) li apliquem a sobre un filtre o kernel que ens retorna un mapa d'activació. Aquest filtre intenta buscar uns patrons o característiques en la imatge indicant-nos en el mapa d'activació si l'ha trobat i en quina zona de la imatge.

- Capa de pooling: normalment es situa darrere la capa convolucional, tot i que entre les dos hi trobem capes de normalització (Batch Normalization)[8]. La funció de les capes de pooling és reduir l'alçada i l'amplada, el nombre de mapes d'activació no varia. Aquest fet produeix la pèrdua d'informació, tot i que és beneficiós per no tenir overfitting. Si utilitzem un filtre de 2x2 de cada 2x2 de la imatge agafem el valor més gran, utilitzant aquest filtre reduïm el tamany a la meitat pel que fa a width i height mantenint la profunditat.

- FC(Fully Connected): Aquesta última capa està totalment connectada amb la capa anterior, cada neurona de la capa anterior està connectada amb les neurones d'aquesta capa amb la finalitat de decidir el valor que ha de prendre cada neurona de la capa FC. Per a fer-ho es fa una suma ponderada de totes les sortides de les neurones de la capa anterior, figura 2. Aquesta capa té tantes neurones com números de classes es vol classificar.

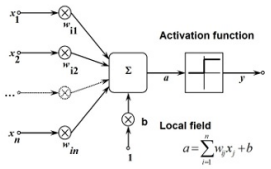


Figure 2: Exemple càlcul valor neurones capa FC

## 4.1 AlexNet

AlexNet és una arquitectura composta per 5 capes convolucionals algunes seguides per una capa max-pooling i finalment trobem 3 capes fully Connected[14]. Les capes convolucionals no sempre tenen la mateixa mida de kernel, el kernel varia segons la capa entre els següents valors: 11x11, 5x5 i les ultimes 3 són 3x3. Les capes de pooling utilitzades tenen una mida de kernel de 3 i un stride de 2 que provoca que la imatge input passi a ser un output amb la mida del weight i el width a la meitat. Les dues primeres capes fully Connected tenen 4096 sortides i la última té 1000 sortides que en les nostres proves haurem de canviar-la depenent del dataset que utilitzem.

Després de cada capa convolucional trobem una capa anomenada ReLU[14] aquesta capa aplica una funció $f(x) = max(0, x)$, si el valor de x és inferior a 0, x pren el valor de 0, utilitzant aquesta capa s'accelera la convergència del descens de gradient comparades amb altres funcions com sigmoid o tanh, això passa per ser lineal i no saturant.

### 4.2 GoogleNet

GoogleNet és una arquitectura composta per 22 capes, o 27 capes si comptem les capes de pooling. Va ser una arquitectura desenvolupada per investigadors de Google. La principal diferència que trobem amb AlexNet és la seva profunditat, molt superior a AlexNet. La xarxa GoogleNet va guanyar la competició de classificació ILSVRC l'any 2014 obtenint un error top-5 d'un 6,66%, la competició tractava de classificar imatges distribuïdes en 1000 classes.
Aquesta xarxa utilitza la idea Network in Network[11]. Aquesta idea ens proposa canviar l'operació convolucional per una MLP o petita xarxa neuronal, que ens serveix per a capturar millor les petites variacions locals que es poden trobar a la imatge. En aquesta arquitectura aquestes petites xarxes neuronals utilitzades són anomenades Inception Modules.
Aquesta arquitectura està composta primer per 3 capes convolucionals i seguidament podem trobar 9 inception modules o Networks in Network[11]figura 6. Un inception module utilitza en paral·lel diferents filtres convolucionals de 1x1, 3x3 i 5x5. El resultat de cadascún dels 4 camins es concatena en profunditat, aquest fet provocaria tenir una dimensionalitat molt alta i per eliminar aquesta dimensionalitat s'utilitzen les capes convolucionals 1x1 que serveixen per reduir la profunditat abans de fer la concatenació de blocs en paral·lel. Un cop feta la concatenació aquest resultat és l'entrada del següent Inception Module. En aquesta arquitectura trobem 3 capes Fully Connected que ens indiquen l'accuracy del nostre model o el loss al mig de la xarxa. Els dos classificadors auxiliars a capes intermitges ens ajuden a que el gradient flueixi millor (ja que el gradient en un camí molt llarg es perd).

### 4.3 VGG

Es tracta d'una arquitectura més profunda que la xarxa AlexNet però menys profunda que GoogleNet[15]. Va guanyar el ILSVRC del 2014 en la prova de localització i va quedar en segona posició en el ILSVRC 2014 en la prova de classificació darrere de l'arquitectura GoogleNet. Va ser creada per un grup d'investigació de la universitat d'Oxford. Podem trobar dos versions de VGG segons la profunditat, VGG – 16 layers i VGG – 19 layers.
Amb aquesta arquitectura la seva entrada sempre han de ser imatges de 224x224. Utilitza capes convolucionals de 3x3 que tenen un camp receptiu petit. També utilitza capes de max pooling amb un stride de 2.
Aquesta arquitectura utilitza moltes capes convolucionals amb filtres de mida 3x3 que és beneficiós perquè utilitzar tres filtres de 3x3 dóna el mateix camp efectiu que un kernel 7x7 però utilitza menys paràmetres que el 7x7. En concret utilitza 13 capes convolucionals 3x3 i tres capes Fully Connected de les que dues són amb 4096 sortides i una amb 1000 sortides. El camp efectiu de dues capes convolucionals 3x3 és el mateix que una 5x5, i el camp efectiu de tres capes convolucionals de 3x3 és el mateix camp efectiu d'una 7x7. Les millores d'utilitzar cadenes de capes 3x3 respecte les 5x5 o les 7x7 es troba en els següents punts:

- Fa que la funció de decisió sigui més discriminativa.
- Redueix el número de paràmetres. Si utilitzem 3 capes 3x3 i C canals necessitem $3 * 3^2 * C^2 = 27C^2$ paràmetres i si utilitzem una capa de 7x7 en necessitem $7^2 * C^2 = 49C^2$.

### 4.4 ResNet

Resnet és l'arquitectura amb més capes de les arquitectures amb les que hem treballat [12]. Va ser presentada l'any 2015 en la competició ILSVRC quedant primera en les competicions ImageNet detection i Imagenet localization. De l'arquitectura resnet hi ha 3 tipus: resnet-50, resnet-101 i resnet-152. Nosaltres hem treballat amb la resnet-50 que està formada per 49 capes convolucionals i 1 capa fully Connected. Resnet utilitza residual connections [12]. L'ús de residual connections ens permet poder tenir una xarxa amb moltes capes i que el backpropagation arribi a les primeres capes. En el cas de no utilitzar residual connections l'error no arribaria correctament i la xarxa no aprendria, estaria morta. L'ús de residual connections també millora la velocitat del train.

## 5 RESULTATS

### 5.1 Data augmentation

Hi ha vegades en les que el número d'imatges de les que disposem no són suficients per aprendre una xarxa molt complexa, quan ens trobem en aquest problema podem resoldre-ho de dues maneres diferents: primer podem augmentar el dataset amb més imatges de les xarxes socials, o segon podem augmentar el nostre dataset fent transformacions de les imatges que ja tenim. En el nostre cas s'ha aplicat la segona manera, i es tracta d'una tècnica anomenada

data augmentation [9].
La tècnica data augmentation ens serveix per afegir imatges diferents al nostre dataset. Aquestes imatges utilitzades són modificades lleugerament "respecte" les originals del dataset i serveixen per a tenir una nova imatge per aprendre. La tècnica data augmentation dóna bons resultats quan el número d'imatges de les que disposem és insuficient per aprendre un model molt complex.
Per a estudiar si millora o no el nostre model en aplicar data augmentation hem aplicat diferents tècniques de data augmentation sobre les imatges dels nostres dos datasets i hem fet aprenentatge creant diferents models per a veure si l'accuracy millorava o no.
Les proves s'han fet amb aprenentatges de 25.000 iteracions i un learning rate de 0,001. El learning rate és diferent en les capes modificades respecte l'arquitectura utilitzada per Imagenet, en les capes modificades el learning rate es multiplica per 10. Això es fa perquè les capes modificades no se'ls aplica finetuning, els pesos de la capa són inicialitzats aleatòriament, i volem que aprenguin més que aquelles on els pesos són d'un model entrenat anteriorment, en el nostre cas utilitzant els pesos d'un model entrenat amb el dataset Imagenet.
La CNN utilitzada per a provar el data augmentation és la GoogleNet, a l'arquitectura de la CNN no se li ha fet cap modificació respecte a la utilitzada a posteriori per a ser comparada amb la resta d'arquitectures estudiades. L'única diferència està en que les imatges d'entrada a la xarxa per a l'aprenentatge se'ls apliquen transformacions respecte a les originals.
Hem executat una prova sense aplicar cap de les diferents tècniques de data augmentation. En aquesta prova hem obtingut un 76,94% en utilitzar el dataset food101 i un 61,82% en utilitzar el dataset food256.
Les tècniques de data augmentation provades es poden visualitzar a la figura 7 (Apèndix) i són les següents:

- Mirror: Aquesta tècnica genera un reflex de la imatge, els píxels que estaven a l'esquerra passen a la dreta de la imatge i els de la dreta a l'esquerra. Utilitzant aquesta tècnica hem obtingut un 77,27% d'accuracy utilitzant el dataset food101 i un 62,43% d'accuracy utilitzant el dataset food256. Veient la diferència d'accuracy que produeix aplicar aquesta tècnica, s'ha decidit fer totes les proves de les següents tècniques utilitzant aquesta.

- Alter RGB: Aquesta tècnica varia aleatòriament el valor dels canals de cada píxel entre 5 i -5. S'ha obtingut un accuracy d'un 77,15% utilitzant el dataset food101 i d'un 60,6% utilitzant el dataset food256.

- Rotate: gira la imatge entre 8 i -8 graus aleatòriament. En aplicar aquesta tècnica s'ha obtingut un accuracy d'un 76,53% amb el dataset food101 i un 62,63% amb el dataset food256.

- Gamma correction: modifica la il·luminació segons un valor gamma, els valors gamma utilitzats són els següents: 0.8, 0.9, 1.1, 1.2 i 1.3. S'ha obtingut un accuracy d'un 77,15% utilitzant el dataset food101 i d'un 62,50% utilitzant el dataset food 256.

- Smoothing: Convoluciona la imatge amb un kernel 5x5 de tot uns, provocant l'emborronament de la imatge. S'obté un accuracy d'un 76,73% utilitzant el dataset food101 i d'un 62,06% utilitzant el dataset food256.

Mirant els resultats obtinguts, a la taula 1, podem veure que hi ha tècniques que milloren l'accuracy en un dataset i no en un altre. En el cas del food101 no hi ha cap tècnica que superi l'accuracy obtingut per la tècnica mirror. En el cas del food256 trobem que la tècnica rotate juntament amb mirror i la tècnica gamma correction juntament amb el mirror obtenen millor accuracy que la tècnica mirror. Aquest fet ens porta a provar d'adjuntar el mirror, el rotate i el gamma correction en un mateix model obtenint un 62,63%.

## 5.2 Proves amb diferents arquitectures

Les primeres decisions primordials eren decidir les configuracions bàsiques de les diferents CNN com són les transformacions de les imatges (data augmentation) i si utilitzar finetuning o no i en quines capes utilitzar el finetuning. Per a decidir quina transformació era la més adient d'aplicar, s'han provat diferents transformacions utilitzant la xarxa GoogleNet, aquest procediment està explicat en l'apartat del data augmentation, i veient els bons resultats d'aplicar mirror s'ha decidit aplicar-lo en tots els aprenentatges de les diferents arquitectures. Un cop decidit quin data augmentation fem, passem a decidir si fem el finetuning o no. En aquest cas per decidir si apliquem finetuning o no utilitzarem la CNN AlexNet.
El finetuning es basa en utilitzar els pesos d'una xarxa entrenada per a l'inicialització dels pesos, en el nostre cas hem utilitzat els pesos d'una xarxa entrenada amb el conjunt d'imatges ImageNet. ImageNet és un conjunt molt gran d'imatges classificades en 1000 classes. En el nostre cas haurem de modificar la última capa de la nostre CNN canviantli el nom i el número de sortides a 101 o 256 segons el dataset que utilitzem. Després de fer les diferents proves amb AlexNet s'ha decidit fer finetuning a totes les capes menys a la última, aquests resultats

| | Basic | Mirror | Alter RGB | Rotate | Gamma Correction | Smoothing |
|---|---|---|---|---|---|---|
| Food-101 | 76,94% | 77,27% | 77,15% | 76,53% | 77,15% | 76,73% |
| Food-256 | 61,82% | 62,43% | 60,6% | 62,63% | 62,50% | 62,06% |

Table 1: Accuracy aplicant diferents transformacions a les imatges d'aprenentatge utilitzant GoogleNet.

| Food101 | FC6 | FC7 | FC8 | No FV |
|---|---|---|---|---|
| 15.000 | 61.87% | 62,15% | 66,74% | 40,19% |
| 30.000 | 64,74% | 65,49% | 67,25% | 50,17% |

Table 2: Accuracy aplicant diferents tipus de finentuning amb dataset food 101 utilitzant AlexNet.

| Food256 | FC6 | FC7 | FC8 | no FV |
|---|---|---|---|---|
| 15.000 | 47.23% | 53,80% | 54,72% | 21,51% |

Table 3: Accuracy aplicant diferents tipus de finentuning amb dataset food 256 utilitzant AlexNet.

estan explicats al següent apartat. En la taula 4 podem veure els diferents resultats que han donat els models segons l'arquitectura i el dataset utilitzat.

#### 5.2.1 AlexNet

La primera CNN utilitzada va ser la Alexnet. Aquesta arquitectura ens ha servit per comprovar si aplicar finetuning ens dóna millors models o no i en quines capes no hem d'aplicar finetuning. Primerament es va fer una execució de la CNN sense finetuning obtenint un accuracy d'un 51,09% amb food101 i 45.000 iteracions i un accuracy d'un 28,22% amb food256 i 45.000 iteracions.
Per a decidir si finetuning és beneficiós per l'accuracy del nostre model o no, hem fet diferents proves. Tenint en compte que la CNN AlexNet al final té 3 capes Fully-Connected hem fet les següents proves:

- Aplicar finetuning a totes les capes excepte la última capa FC, aquesta prova l'hem anomenat FC8.
- Aplicar finetuning a totes les capes excepte la última i penúltima capa FC, aquesta prova l'hem anomenada FC7.
- Aplicar finetuning a totes les capes excepte les 3 capes FC i l'hem anomenat FC6.

Utilitzant el dataset food 101 s'ha fet cada prova fent 15.000 i 30.000 iteracions i en el dataset food 256 s'ha fet cada prova fent 15.000 iteracions.

Mirant la taula 2 i 3 podem concloure que s'obtenen els millors resultats fent finetuning a totes les capes excepte la última capa FC. També podem concloure que és millor aplicar finetuning que no aplicar-lo, en el cas d'utilitzar el dataset food101 passem d'un accuracy d'un 51,09% a 66,75% i si utilitzem el dataset food256 passem d'un accuracy d'un 28,22% a un 54,72%.
Si utilitzem l'arquitectura AlexNet i apliquem finetuning a totes les capes excepte la última i fem un entrenament de 25.000 iteracions amb el dataset food101 s'ha obtingut un accuracy d'un 67,34% i en el dataset food256 un accuracy de 53,24%.

#### 5.2.2 GoogleNet

Per a l'aprenentatge d'un model utilitzant l'arquitectura GoogleNet i veient el bon funcionament del finetuning en altres arquitectures decidim inicialitzar els pesos del model utilitzant els pesos d'un model entrenat anteriorment. El model utilitzat ha sigut entrenat utilitzant el dataset ImageNet que a diferència dels nostres datasets té 1000 classes. Les modificacions aplicades a la nostra arquitectura respecte la del model utilitzat està en 3 capes on el número de sortides és de 1000 i en el nostre cas l'hem canviat a 101 o 256 depenent del dataset que utilitzem. Al haver-se modificat aquestes 3 capes respecte l'arquitectura del model utilitzat aquestes 3 capes no se'ls aplica finetuning i els pesos són inicialitzats aleatòriament.
GoogleNet a diferència d'altres CNN està composta per tres classificadors FC, cadascún dels classificadors ens dóna l'accuracy que donaria el model si aquest fos l'últim de la nostra xarxa. Els classificadors ens ajuden a calcular el gradient per a fer l'actualització dels pesos, la xarxa és molt profunda i sense els classificadors intermitjos no s'actualitzarien correctament. En les nostres proves sempre ens ha donat un accuracy superior al 3er classificador respecte als altres dos classificadors, i el segon classificador dóna millor accuracy que el primer classificador. Ens podem trobar el cas que el segon classificador doni millor accuracy que el 3er classificador.
Utilitzant una xarxa GoogleNet pre-entrenada amb els pesos d'un model entrenat amb el dataset ImageNet obtenim un accuracy d'un 77,27% en el cas d'utilitzar el dataset food101 i d'un 62,43% utilitzant el dataset food256.

#### 5.2.3 VGG

La xarxa VGG ha estat entrenada fent finetuning amb una xarxa VGG entrenada amb el dataset ImageNet i 1000 clases. La xarxa VGG utilitzada és una xarxa composta per 16 capes de les que 13 són capes convolucionals de 3x3 i finalitza amb 3 capes

| | Food-101 | food-256 |
|---|---|---|
| AlexNet | 67,34% | 53,24% |
| GoogleNet | 77,27% | 62,43% |
| VGG | 78,82% | 63,03% |
| ResNet | 81,59% | 63,21% |

Table 4: Accuracy de les diferents CNN.

| | Food-101 | food-256 |
|---|---|---|
| from Lukas et. al [16] | 77,4% | 54,7% |
| from Lukas et. al [17] | 77,35% | 63,77% |
| GoogleNet CAM | 77,49% | 59,64% |
| ResNet CAM | 82,07% | 63,54% |

Table 5: Accuracy de les diferents CNN aplicant CAM.

FC. Mirant els resultats obtinguts en l'arquitectura AlexNet es decideix que només modifiquem la última capa FC, la que els seus pesos són inicialitzats aleatòriament, i les altres capes fem finetuning. En la última capa canviem el número de sortides i el posem a 101 o 256 depenent del dataset que utilitzem.
En VGG hem obtingut un accuracy d'un 78,82% utilitzant el dataset food101 i 25.000 iteracions i un accuracy d'un 63,03% utilitzant el dataset food256 i 25.000 iteracions.

#### 5.2.4 ResNet

Dels 3 tipus de ResNet que podem trobar (ResNet-50, ResNet-101 i ResNet-152) hem escollit la ResNet-50. La ResNet-101 i la ResNet-152 no les podíem executar per falta de memòria. La ResNet-50 finalitza només amb una capa Fully-Connected. Hem aplicat finetuning utilitzant una xarxa ResNet-50 entrenada amb ImageNet i 1000 sortides, adaptant la ultima capa a 101 o 256 sortides segons el dataset utilitzat i augmentant-li el lr_mult que és el paràmetre que ens permet que la capa aprengui més o menys.
Utilitzant ResNet-50 i aplicant finetuning excepte la última capa, hem obtingut un 81,59% utilitzant el dataset food101 i un 63,21% d'accuracy utilitzant el dataset food256.

### 5.3 CAM (Class Activation Mapping)

Aquest punt proposa una tècnica per explorar en quines parts de la imatge es fixa especialment la CNN. Ens mostra quines zones de la imatge fan que la CNN predigui una classe o una altre[18].
Per a obtenir el CAM s'utilitzen globals average pooling (GAP) en la CNN. El CAM per a una categoria particular indica el discriminant de la regió de la imatge. Utilitzant una xarxa GoogleNet s'ha de fer una modificació, aquesta tracta d'abans de la capa softmax introduir una capa global average pooling, aquest fet ens genera uns mapes de característiques convolucionals que s'utilitzaran per introduir-los a una capa FC que ens donarà la sortida desitjada, indicant a quina el probabilitat que la imatge pertanyi a cada classe. Utilitzant els mapes de característiques obtinguts en la última capa convolucional i els pesos per a cada mapa

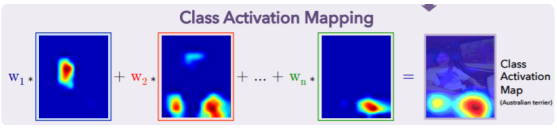


Figure 3: Procés calcul mapa activació

segons la classe generem el mapa d'activació per a la classe indicada com es pot veure en la figura 3.

Els resultats dels diferents models creats utilitzant CAM els podem veure en la taula 5, on algunes precisions són inferiors als models que no utilitzen CAM i d'altres són superiors.
Mirant com es modificava la xarxa GoogleNet s'ha modificat la xarxa ResNet eliminant un conjunt de capes i modificant la última capa pooling de kernel_size 7 a 14. Hem eliminat les capes per a obtenir una sortida de la última capa convolucional amb les mateixes mides que la sortida de la última capa convolucinal de GoogleNet.
Utilitzant els resultats de fer el test de cada imatge i utilitzant la sortida de la última capa i la sortida de la última capa convolucional generem el mapa d'activació per a la classe escollida per a cada imatge. A partir del mapa d'activació generem un box que ens indica en quina zona de la imatge es troba la informació important de la imatge, utilitzem la funció findContours que ens retorna els boxs dels contorns ordenats per mida. D'aquesta sortida sempre escollim el segon, aquesta ens indica exactament la zona desitjada.
En la figura 4 podem visualitzar un exemple de generació de mapa d'activació. Es passa la imatge (a) pel nostre model que ens retorna el mapa d'activació i ens indica quines part de la imatge ens han servit per a classificar que la imatge és Apple_pie, aquestes parts tenen colors més càlids. Finalment, generem el box que ens indica quina zona de la imatge ha mirat principalment el nostre model. En el apèndix en les figures 8 i 9 podem trobar altres imatges amb els seus corresponents mapes d'activació.

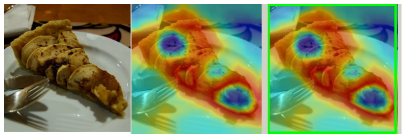

Figure 4: Imatge original, mapa activació i mapa activació + caixa.

### 5.4 Refinant el model

Un cop aplicat data augmentation s'ha intentat veure en quines imatges falla el nostre model i buscar una possible raó. Primerament, després de provar 4 arquitectures diferents i aplicant-los data augmentation hem pogut apreciar que la millor xarxa convolucional és la ResNet-50. Utilitzant la ResNet com es pot veure a la taula 4 hem obtingut un accuracy d'un 82,07% utilitzant el dataset food-101 i d'un 63,54% utilitzant el dataset food-256.
Utilitzant el dataset food-101 i la CNN ResNet hem obtingut que les imatges més ben classificades són les imatges de les classes deviled_eggs amb un 93,6%, hot_and_sour_soup amb un 93,6%, pad_thai amb un 96,4% i edamame amb un 99,2%. I les classes que es reconeixen més malament són steak amb un 42,4%, pork_chop amb un 51,6% i bread_pudding amb un 54,4%. En totes les classes s'ha testejat utilitzant 250 imatges prèviament etiquetades.
Si mirem atentament la classe en la que les seves imatges s'han classificat més malament, podem observar que dels seus 144 errors, 36 dels errors se'ls havia assignat la classe filet_mignon. Quan hem testejat la classe filet_mignon la classe més assignada en els seus errors ha sigut la classe steak. Veient aquesta doble relació ens fa pensar que les imatges són molt similars, aquesta similitud la podem apreciar a la figura 5.

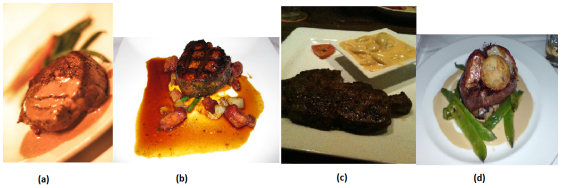


Figure 5: Filet_mignon (a), (b) i steak (c) i (d)

A la figura 10 de l'apèndix podem veure aliments classificats correctament. A la figura 11 de l'apèndix podem veure aliments classificats incorrectament. De les imatges mal classificades si observem la (b) podem veure que el nostre model relaciona la pizza en que ha de se rodona, i que si són triangles els classifica com a tacos. Algunes altres imatges mal classificades són la (j), (k) i (l) que podem veure que són molt diferents a les classificades correctament que pertanyen a la clase falafel. Veient les imatges ben classificades i les mal classificades, dóna a pensar que el dataset utilitzat etiqueta imatges molt diferents en la mateixa classe dificultant el bon funcionament del model.

### 5.5 Conclusions dels resultats

En aquest apartat compararem els resultats obtinguts pels nostres models amb l'estat de l'art. Mirant la taula 5 podem veure que en el dataset food-256 obtenim un 63,54% inferior al 63,77% que es pot visualitzar a la taula 5. Per a obtenir el 63,77% s'ha utilitzat una xarxa pre-entrenada amb 2000 classes i utilitzen Fisher Vector(FV).
En els resultats del food-101, hem obtingut un model molt millor que els de l'estat de l'art. El millor model trobat obtenia una precisió d'un 77,4% bastant inferior al obtingut per nosaltres, 82,07%. La diferència principal és que anteriorment havien utilitzat la CNN GoogleNet i nosaltres hem utilitzat la ResNet. També cal dir que ells van fer un aprenentatge de 250.000 iteracions i nosaltres de 25.000 iteracions, el que fa veure que el nostre model aprèn molt més ràpidament.

## 6 Conclusions

Hem pogut concloure que la CNN que millor s'adapta als nostres datasets és la ResNet. En els dos datasets ha obtingut un accuracy superior a les altres arquitectures. Sorprenentment s'obté millor accuracy en el cas de reduir unes quantes capes la mida de la xarxa.
També podem concloure que és molt beneficiós utilitzar finetuning en quantes més capes millor i que és beneficiós aplicar mirror. Hem comprovat que utilitzar més imatges en l'etapa d'entrenament implica una millora en l'accuracy del model resultant. Una manera d'augmentar el nostre accuracy seria augmentant significativament el número d'imatges utilitzades descarregant-nos més imatges de les xarxes socials. Hem pogut comprovar que en aplicar data augmentation els nostres models no milloren gaire la seva precisió, en alguns casos fins i tot es redueix.
Hem pogut veure que el mapa d'activació té els colors més càlids en les zones on hi ha l'aliment de la classe en la que pertany la imatge.
Comparant amb l'estat de l'art hem superat significativament el resultat trobat en food-101 passant d'un 77,4% en food-101 a un 82,07% i no hem obtingut un millor model al trobat amb food-256 tot i que ens hem quedat molt a prop, obtinguent un 63,54 per un 63,77% de l'estat de l'art.

## Agraïments

Vull donar les gràcies als meus tutors Jordi Gonzàlez i Guillem Cucurull per la seva constant ajuda i els seus consells.

## REFERENCES


[1] Bossard, Lukas and Guillaumin, Matthieu and Van Gool, Luc *Food-101–mining discriminative components with random forests* European Conference on Computer Vision, pages 446-461, 2014.

[2] Kawano, Y. and Yanai, K. *Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation* European Conference on Computer Vision, pages 3-17, 2014.

[3] Rosenblatt, Frank *The perceptron: a probabilistic model for information storage and organization in the brain.* American Psychological Association, 1958.

[4] Rumelhart, David E *Backpropagation: The basic theory*,Backpropagation: Theory, Architectures and Applications, pages 1-34, 1995.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. *ImageNet: A Large-Scale Hierarchical Image Database.* CVPR 2009, pages 248-255, 2009

[6] N Srivastava, GE Hinton, A Krizhevsky *Dropout: a simple way to prevent neural networks from overfitting*,Journal of Machine Learning Research, pages 1929-1958, 2014

[7] C Szegedy, W Liu, Y Jia, P Sermanet *Going deeper with convolutions*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1-9, 2015

[8] Ioffe, Sergey and Szegedy, Christian *Batch normalization: Accelerating deep network training by reducing internal covariate shift* arXiv:1502.03167, 2015.

[9] Ping Dong and Nikolas P. Galatsanos *Affine Transformation Resistant Watermarking Based on Image Normalization* , Image Processing. 2002. Proceedings. 2002 International Conference on, pages 489-492, 2002.

[10] Yangqing Jia *Caffe: Convolutional Architecture for Fast Feature Embedding.*, Proceedings of the 22nd ACM international conference on Multimedia, pages 6755-678, 2014.

[11] Lin, Min and Chen, Qiang and Yan, Shuicheng *Network in network* arXiv:1312.4400, 2013.

[12] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian *Deep residual learning for image recognition* arXiv:1512.03385, 2015.

[13] Szegedy, Christian and Vanhoucke, Vincent and Ioffe, Sergey and Shlens, Jonathon and Wojna, Zbigniew *Rethinking the inception architecture for computer vision* arXiv:1512.00567, 2015.

[14] Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey EKrizhevsky *Imagenet classification with deep convolutional neural networks*,Advances in neural information processing systems, pages 1097–1105, 2012.

[15] Simonyan, Karen and Zisserman, Andrew *Very deep convolutional networks for large-scale image recognition* arXiv:1409.1556, 2014.

[16] Liu, Chang and Cao, Yu and Luo, Yan and Chen, Guanling and Vokkarane, Vinod and Ma, Yunsheng *DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment*, International Conference on Smart Homes and Health Telematics, pages 37-48, 2016.

[17] Yanai, Keiji and Kawano, Yoshiyuki *Food image recognition using deep convolutional network with pre-training and fine-tuning*, International Multimedia Expo Workshops (ICMEW), pages 1-6, 2015.

[18] Zhou, Bolei and Khosla, Aditya and Lapedriza, Agata and Oliva, Aude and Torralba, Antonio *Learning Deep Features for Discriminative Localization* arXiv:1512.04150, 2015.

[19] Lazebnik, Svetlana and Schmid, Cordelia and Ponce, Jean *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*, Computer vision and pattern recognition, pages 2169–2178, 2006.

[20] Sànchez, Jorge and Perronnin, Florent and Mensink, Thomas and Verbeek, Jakob *Image classification with the fisher vector: Theory and practice*, International journal of computer vision, pages 222–245, 2013.

[21] Bosch, Anna and Zisserman, Andrew and Munoz, Xavier *Image classification using random forests and ferns*, ICCV 2007, pages 1-8, 2007.

[22] Moosmann, Frank and Nowak, Eric and Jurie, Frederic *Randomized clustering forests for image classification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1632–1646, 2008.

[23] Singh, Saurabh and Gupta, Abhinav and Efros, Alexei A *Unsupervised discovery of mid-level discriminative patches*, Computer Vision–ECCV, pages 73-86, 2012.

## APÈNDIX

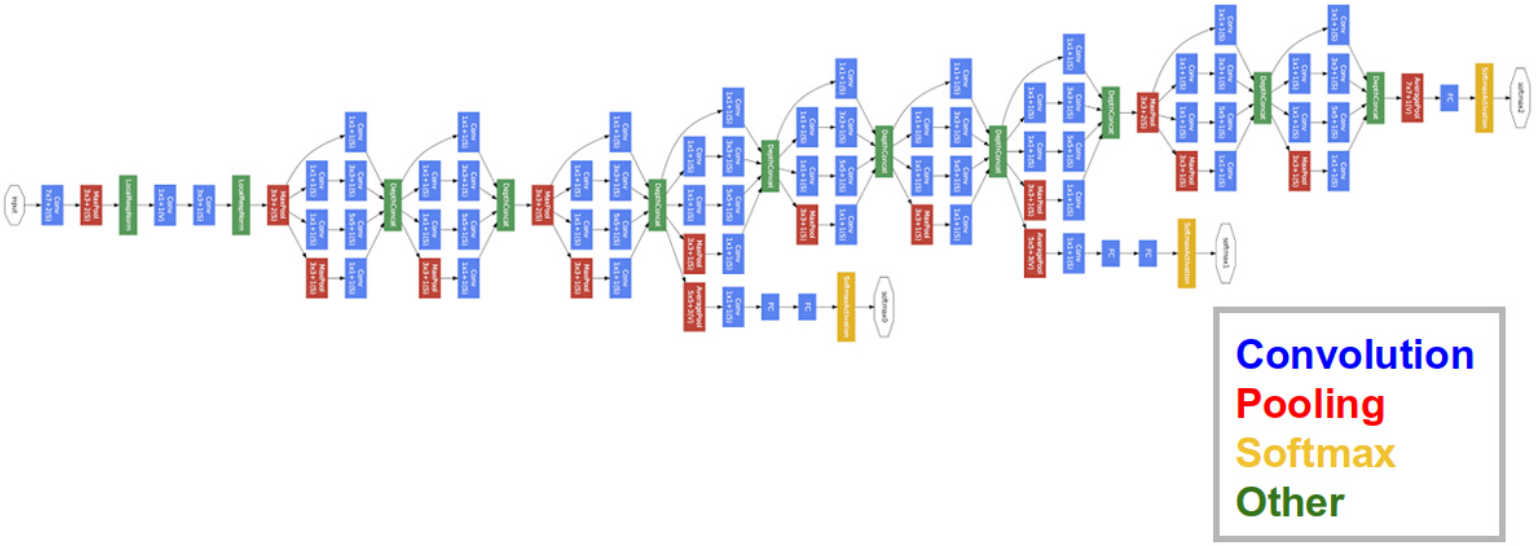


Figure 6: Arquitectura GoogleNet

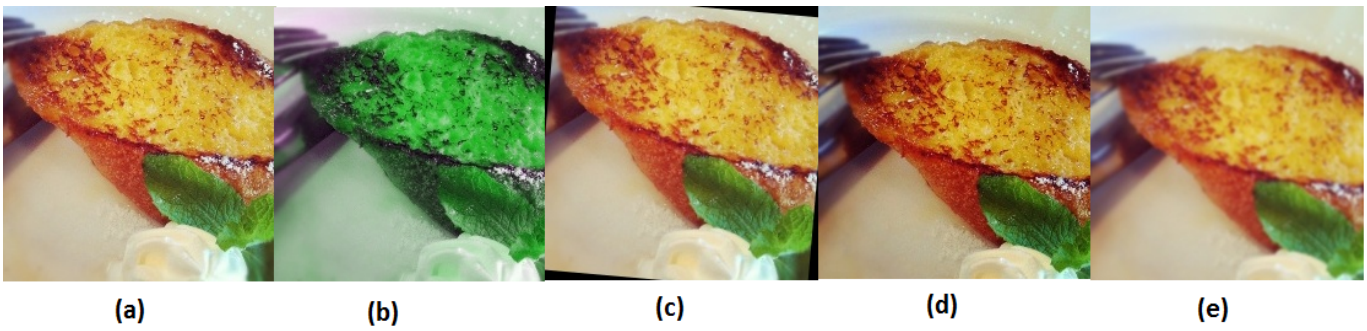


Figure 7: Transformacions imatge: (a)Imatge no modificada, (b)RGB alter, (c) rotació imatge, (d) gamma correction, (e)smoothing

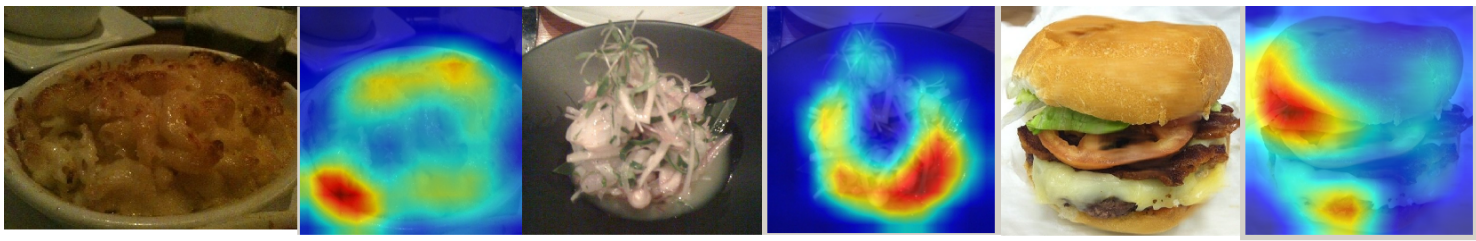

Figure 8: Imatges dataset food-101 amb el seu corresponent mapa d'activació.

Figure 9: Imatges dataset food-101 amb el seu corresponent mapa d'activació.


Figure 10: Imatges classificades correctament del dataset food-101. (a), (b) i (c) són chicken_curry. (d), (e) i (f) són falafel. (g) i (h) són pizza. (i) i (j) són raviole.

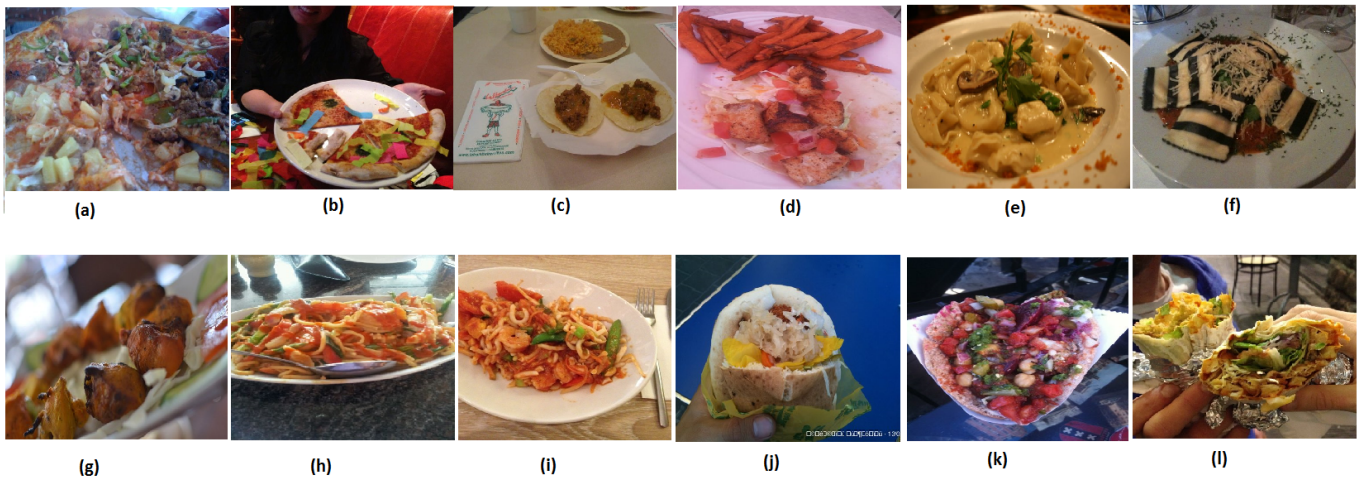


Figure 11: Imatges classificades incorrectament del dataset food-101. (imatge)(correcte, clase clasificada). (a)(pizza, nachos), (b)(pizza,tacos), (c)(tacos, deviled_eggs), (d)(tacos, crab_cakes), (e)(raviole, gnocchi), (f)(raviole, lasagna), (g)(chicken_curry, chicken_wings), (h)(chicken_curry, pad_thi), (i)(chicken_curry, pad_thi), (j)(falafel, strawberry_shortcake), (k)(falafel, guacamole) i (l)(falafel, breakfast_burrito)