

Confecció d'un conjunt ampliat d'entrenament per a l'anàlisi d'emocions bàsiques a partir d'imatges prèvies

Jordi Sors Durán

Enginyeria Informàtica (2n. Cicle) - 2003

Visió per computador

Dr. David Masip Rodó

Juny 2017

© Jordi Sors Durán

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

*Agraeixo el temps que m'ha
regalat la meua família
durant aquesta etapa
d'estudi.*

*No puc oblidar les notes
positives que m'ha anat
deixant el meu fill Èric
sobre el teclat i l'esforç de
l'Anna que m'ha ajudat fins
la correcció final del
treball.*

Jordi

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Confecció d'un conjunt ampliat d'entrenament per a l'anàlisi d'emocions bàsiques a partir d'imatges prèvies</i>
Nom de l'autor:	<i>Jordi Sors Durán</i>
Nom del consultor/a:	<i>Dr. David Masip Rodó</i>
Nom del PRA:	<i>Nom i dos cognoms</i>
Data de lliurament (mm/aaaa):	<i>06/2017</i>
Titulació o programa:	<i>Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>11.109 – PFC-Visió per Computador</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Data Augmentation, TensorFlow, CNNs</i>
Resum del Treball (màxim 250 paraules):	
<p>El projecte consisteix en ampliar els conjunts d'entrenament a partir de conjunts d'imatges classificades existents i validar la hipòtesi per la qual les xarxes neuronals convolucionals (CNN) entrenades amb conjunts ampliat tindran un millor encert en l'anàlisi de les imatges de test.</p>	

Abstract (in English, 250 words or less):

Índex

1. Introducció.....	2
1.1. Context i justificació del Treball.....	2
1.2. Objectius del Treball	2
1.3. Enfocament i mètode seguit.....	2
1.4. Planificació del Treball	2
1.5. Breu sumari de productes obtinguts	5
1.6. Breu descripció dels altres capítols de la memòria.....	5
2. Resum Executiu	6
2.1. Antecedents	6
2.2. Resum de la solució – generació del conjunt ampliat	6
2.3. Anàlisi dels resultats	7
2.4. Conclusions.....	8
3. Estat de l'Art.....	9
3.1. Situació Actual	9
3.2. Ampliació del conjunt d'entrenament	9
3.3. Hipòtesi de treball	10
4. Creació de noves Imatges	11
4.1. Anàlisi morfològica de les imatges.....	11
4.2. Creació de noves imatges.....	11
4.3. Generació i avaluació de les noves imatges.....	12
4.4. Generació massiva	15
4.5. Programa	18
5. Comprovació del conjunt ampliat.....	20
5.1. Antecedents	20
5.2. Instal·lació de les eines.....	20
5.3. Proves prèvies	20
5.4. Primers experiments	21
5.5. Disseny de l'experiment.....	25
5.6. Resultats	27
6. Conclusions.....	39
7. Bibliografia	39
7.1. Referències	39
7.2. Bibliografia	40
8. Annex A: Codi Generació imatges (JAVA).....	41
9. Annex B: Codi Entrenament CNN (PY).....	49

1. Introducció

1.1. Context i justificació del Treball

En el reconeixement de les emocions bàsiques a partir d'imatges facials hi ha dues realitats contrastades en la visió per ordinador, l'anàlisi a partir de imatges tractades, ben enfocades, ben il·luminades, pràcticament seleccionades o fetes a "mida" per aquest propòsit i les imatges de domini lliure, o sigui, les imatges que podem extreure de qualsevol lloc, una pel·lícula per exemple, el que es coneix com l'entorn "in the wild". El percentatge d'encert baixa estrepitosament en aquest segon cas obtenint uns resultats poc acceptables i molt allunyats del reconeixement en el primer cas i molt més allunyat de l'encert que té un humà.

I com es pot millorar aquest percentatge d'encert? Aquest projecte intentarà donar resposta a aquesta pregunta.

1.2. Objectius del Treball

El projecte consisteix en ampliar els conjunts d'entrenament a partir de conjunts d'imatges classificades existents i validar la hipòtesi per la qual les xarxes neuronals convolucionals (CNN) entrenades amb conjunts ampliat tindran un millor encert en l'anàlisi de les imatges de test.

1.3. Enfocament i mètode seguit

Es tracta d'ampliar el conjunt d'imatges original, per tant el primer punt és veure com s'amplia.

Un cop tenim el conjunt ampliat, veure si el nou conjunt de dades entrena millor una xarxa neuronal convolucionals que un conjunt sense ampliar.

Fer el test i comparar-lo per tal de validar la hipòtesi.

1.4. Planificació del Treball

Aquest projecte s'ha de realitzar dins el segon semestre lectiu del curs 2016-2017 amb una data d'entrega prefixada el 31/05.

També hi ha dues fites de control, les PAC's amb data d'entrega el 5/04 i el 5/05, en definida, punts de control mensuals del treball.

Els entregables sempre els penso com l'entrega de la memòria contemplant el treball fet fins el moment. Així com el conjunt d'imatges generats.

El projecte s'ha dividit en 8 grans, tenint en compte el temps, penso que puc definir la següent càrrega de treball i dates previstes:

Tasca	Càrrega	Data inici	Data fi
Presa de contacte – definició	1w	20/02	27/02
Anàlisi Morfològica de les imatges	3w	27/02	19/03
Anàlisi imatges de partida	3w	10/03	26/03
Anàlisi eines i entorn de treball	2w	27/02	19/03
Preparació entron i primeres proves gràfiques	1w	19/03	26/03
Programació i generació imarges		26/03	23/4
Entrega PAC 1 – primeres imatges		12/04	12/04
Refinament i regeneració imarges		24/04	20/05
Metodologia de control – com es fa?	1w	18/04	27/04
Entrenament CNN	?	24/04	20/05
Entrega PAC 2 – primers resultats		15/05	15/05
Preentrega memòria	0d	17/05	26/5
Revisió memòria	1w	27/5	31/5
Entrega memòria	0d	31/05	31/05



1.5. Breu sumari de productes obtinguts

Més que productes, són les eines de treball confeccionades. Un programa per generar imatges a partir d'un conjunt original, desenvolupat en Java i un programa per entrenar CNNs desenvolupat amb Python.

1.6. Breu descripció dels altres capítols de la memòria

Els capítols que componen el projecte són:

- resum executiu
- estat de l'art
- Creació de noves imatges
- Comprovació del conjunt ampliat de dades
- Resultats obtinguts
- Conclusions

2. Resum Executiu

2.1. Antecedents

Hi ha diferents línies de recerca en millorar la identificació de les emocions bàsiques a partir del reconeixement d'imatges facials, entre les quals podem destacar-ne un parell: l'especialització de les xarxes neuronals convolucionals CNNs afegint-hi més nivells "ocults" o fer combinacions de diferents CNNs buscant millorar els resultats.

Però hi ha altres camins, en particular el data augmentation, que consisteix en entrenar les CNN amb conjunts d'entrenament amb més informació, més imatges. En aquest cas es preveu que els sistemes seran millors en el reconeixement. El projecte es centra en aquesta línia de treball.

2.2. Resum de la solució – generació del conjunt ampliat

Amb l'anàlisi fet, es decideix que la millor forma d'ampliar el conjunt d'entrenament és partint d'un conjunt d'imatges classificat i per cada imatge generar-ne de noves fent rotacions per poder entrenar el sistema amb més imatges classificades.

El primer punt del projecte és decidir com es generen les noves imatges a partir del conjunt original, s'opta per l'ús de les llibreries d' Open GL sobre JAVA: JOGL.







































































Un dels principals punts de l'estudi ha estat definir l'increment dels graus de gir entre cadascuna de les noves imatges, els límits en cada eix de gir, analitzar si cal tractar tractar la nova imatge, i finalment, valorar la rellevància de la informació generada, la seva tractabilitat i en definitiva, si la nova informació aporta en el procés.

Es parteix de diferents hipòtesis sobre l'increment de l'angle de gir. Incrementar grau a grau no aporta informació rellevant, les imatges properes són molt semblants. Quan l'increment entre cada nova imatge passa a 5° apareix nova informació, sobre tot quan es combina un gir sobre cadascun dels dos eixos.

Notar que la quantitat d'imatges generades pot arribar a un volum d'informació quasi intractable, per exemple si es volen generar noves imatges amb un gir de +/- 60° en un eix i un gir de +/- 45° en l'altre eix, es generen la no menyspreable quantitat de $25 \cdot 19 \cdot 2 = 950$ noves imatges per cadascuna de les originals existent.

Finalment, s'opta per generar 63 imatges a partir de cada imatge original, el producte de 9 girs sobre l'eix X (+/- 40°) i 7 girs l'eix Y (+/- 30°) amb un increment de 10° en cada nou gir.

En la següent taula es pot comprovar l'evolució de les 63 imatges generades a partir d'una imatge original, notar que s'ha aplicat un zoom a la imatge original per evitar els negres en la imatges girades.

Orig. No Zoom							
X / Y	30	20	10	0	350 (-10)	340 (-20)	330 (-30)
40							
30							
20							
10							
0							
350 (-10)							
340 (-20)							
330 (-30)							
320 (-40)							

2.3. Anàlisi dels resultats

La metodologia utilitzada per comprovar si la hipòtesi és certa o falsa es basa en l'entrenament d'una CNN amb unes imatges originals (IO) i les

imatges del conjunt ampliat de les IO i fer que les dues CNN classifiquin un mateix conjunt de test.

Es tracta de fer un seguit d'experiments segons el mètode explicat, veure que les CNN s'entrenen correctament, que convergeixen i finalment, veure que el percentatge d'encert que té la CNN entrenada amb el conjunt ampliat és millor que la CNN entrenada només amb les imatges originals.

Finalment, per comprovar la hipòtesi, es valida que els intervals de confiança de l'encert en un tipus de xarxa i l'altra no es solapen i per tant es pot assegurar que estadísticament el mètode és correcte i que la hipòtesi es compleix.

Els experiments realitzats s'han fet a partir de les imatges de la base de dades classificada FER fent conjunts disjunts de les diferents imatges de cada tipus d'emoció.

S'han realitzat 10 experiments complets comparant 4 subexperiments amb diferents conjunts d'imatges d'entrenament per cada CNN obtenint el següents intervals de confiança:

Tipus Experiment	Descripció Experiment	INTERVAL CONFIANÇA 95%
Subexperiment 1	700 train - 700 test	[27,64 , 29,12]
Subexperiment 2	44k train - 700 test	[29,62 , 31,18]
Subexperiment 3	700 train - 44k test	[25,24 , 26,20]
	700 imatges originals de test (per comparar amb Subexp. 1)	[28,92 , 30,19]
Subexperiment 4	44k train - 44k test	[30,74 , 32,00]
	700 imatges originals de test (per comparar amb Subexp. 2)	[29,98 , 31,65]

2.4. Conclusions

Amb els resultats obtinguts es pot afirmar que la hipòtesi queda validada: si s'entrenen les xarxes amb un conjunt d'imatges ampliat el percentatge d'encert millora 2 punts amb un interval de confiança del 95%.

3. Estat de l'Art

3.1. Situació Actual

Es pot dir que la informació més rellevant per poder identificar, reconèixer l'emoció d'una persona, és la seva pròpia cara. No en va, la dita popular ens recorda que “la cara és un reflex de l'ànima”.

Quan veiem una persona som capaços d'identificar el seu estat d'ànim i reaccionem en conseqüència: les emocions són importants. Si volem que les màquines interactuin amb nosaltres de manera natural, ens cal que ens entenguin també a nivell emocional. Per això el reconeixement de les emocions bàsiques a partir de les imatges facials s'ha convertit en una àrea rellevant en el món de la computació.

La classificació de les emocions bàsiques segons els estudis de Ekman and Firesen [1], s'identificaven 6 emocions bàsiques, compartides en diferents cultures. Així les anomenen anger, disgust, fear, happiness, sadness and surprise.

Per automatitzar la classificació de les emocions bàsiques, s'han generat un seguit d'algorismes basats en el treball de Ekman et al. [2] anomenat Facial Action Coding System (FACS) el qual descriu 44 Action Units (AUs), i cada AU defineix el conjunt de múscles que identifica una emoció bàsica a partir de l'expressió facial.

Aquest treball no pretén entrar en la classificació de les imatges a partir d'aquesta definició, sinó que parteix d'una base de dades de 28.709 imatges classificades per l'entrenament de CNN: Facial Expression Recognition 2013 database (FER) [3]. Notar que aquesta base de dades incorpora l'estat neutral per aquelles que no hi troben cap tret expressiu que les pugui classificar en cap de les 6 definides per Ekman.

3.2. Ampliació del conjunt d'entrenament

El projecte consisteix en obtenir nous conjunts d'entrenament amb més informació a partir de conjunts d'imatges classificades existents.

En aquest punt sorgeixen algunes preguntes:

- I com ampliar aquests conjunts d'entrenament?
- Amb altres imatges classificades?
- Aquestes noves imatges estaran prou “no preparades” per poder ajudar en entorns in the wild?
- Així, com han de ser les noves imatges?
- Quantes imatges s'han de generar?
- El nou conjunt serà tractable computacionalment?

És tot un repte dimensionar i contestar aquestes preguntes abans d'anar avançant, però són les preguntes que han de guiar el projecte.

Una vegada analitzades les imatges classificades i les imatges de l'entorn in the wild, de seguida es veu que hi ha moltíssimes posicions de la cara que no estan en les imatges d'entrenament, la major part de les imatges són frontals! I en l'entorn in the wild n'hi ha en totes posicions. Per tant, ja tenim la resposta a algunes de les preguntes anteriors. Aquest és el punt de partida del projecte: ampliar el conjunt original amb imatges generades a partir d'aquest conjunt (així tenim les imatges classificades) en diferents posicions, és a dir, girades simulant les diferents posicions en les que podem posar la cara.

3.3. Hipòtesi de treball

Si generem un conjunt ampliat d'imatges per tal d'entrenar una xarxa neuronal convolucional (CNN), el percentatge d'encert en la classificació d'un conjunt de test ha de ser millor amb aquest conjunt que si l'entrenament es fa amb el conjunt original (no ampliat).

4. Creació de noves Imatges

4.1. Anàlisi morfològica de les imatges

A partir de les imatges de partida de la base de dades FER [3] he treballat fonamentalment amb 3 imatges tipus per fer l'anàlisi de com generar el conjunt ampliat: somrient, somrient amb la cara mig tapada (totes dues classificades com a happy) i trist (classificada com a sad).

He escollit tres imatges representatives i que em donàvem prou detall per veure com es comportaria la nova generació d'imatges.

Un cop escollides les imatges de partida, i després de fer diferents proves en la rotació de la imatge, he definit com es generaran les noves imatges:

- Les rotacions es faran des del punt central de la imatge i sense variar el punt de l'observador, així totes les imatges rotades queden projectades sobre el pla generant una imatge nova de la mateixa forma. Intento que el mètode no generi més soroll que el propi de la rotació.
- Notar que tot i que el punt de gir del cap es troba en la part posterior del coll, he intentat estandarditzar la rotació per poder tractar de la mateixa forma la diversitat d'imatges diferents que existeixen.
- Només contemplo gir sobre dos eixos, l'eix x i l'eix y. El gir de l'eix z no el contemplo donat que les imatges que genero cada vegada es degeneraven una mica més i no aportaven nova informació.
- La hipòtesi inicial per la generació de noves imatges és
 - rotació sobre l'eix X amb un pas d'angle de 5° de -45° a 45° entenent que el 0 era la imatge original
 - rotació sobre l'eix Y amb un pas d'angle de 5° de -45° a 45° entenent que el 0 era la imatge original
- Amb aquesta hipòtesi de treball, a partir d'una imatge original generem un total de $19 \cdot 19 - 1$ imatges = 360 imatges noves

4.2. Creació de noves imatges

4.2.1. Eines analitzades

Després d'una cerca de llibreries pel tractament d'imatges vaig decidir treballar amb la implementació de OpenGL per JAVA: JOGL. Hi havia altres opcions, sobretot amb llibreries de generació de jocs en JAVA i gestió de 3D en entorn WEB.

El fet de treballar amb JAVA es deu a que és el llenguatge usat en la carrera per fer algunes pràctiques, i a priori havia de ser més fàcil integrar la gestió d'imatges amb l'OpenGL.

La instal·lació de les llibreries del JOGL no va ser complexa, només incorporar els JAR corresponents i seleccionar els de l'entorn on treballa. Les primeres proves amb l'entorn no van ser tant fàcils com em pensava; la generació de figures, els seus girs i la seva representació no sempre es corresponia amb el resultat esperat. Aprendre el focus, el punt de vista, l'ordre dels vèrtexs van ser els punts importants per tal que finalment es representessin els escenaris esperats.

4.2.2. Creació de les noves imatges – procediment

El següent pas va ser fer-ho amb una imatge. Vaig escollir fer-ho texturant una figura plana, un quadrat, donat que les imatges normalitzades tenien una dimensió de 48x48 píxels.

Un cop fet aquest pas, vaig adonar-me d'alguns problemes: com es deforma i canvia el rati de la imatge original en funció de la dimensió i el rati del frame, independentment de la dimensió del quadrat on s'insertava la imatge com a textura

- per tant el frame ha de tenir una dimensió quadrada en la superfície de treball, s'ha de tenir en compte la barra del frame
- com afecta l'ordre de la introducció de comandes en la pila d'estats de l'openGL, tot i que ho deia el manual, hi ha moltes operacions no commutatives, això ho he anat aprenent a mesura que avançava
- tenir en compte les figures positives i les negatives, en funció de l'ordre en el que es defineixen els vèrtexs la figura, si els vèrtexs van en la direcció de les agulles del rellotge o en contra respectivament
- com influeix la textura en una figura negativa i positiva
- l'error inicial de girar 180° des d'una aresta de la imatge, les imatges quedaven estranyes, de fet feien l'efecte de veure una pel·lícula per la part de darrera de la pantalla.
- el fet que el gir només sigui en sentit de les agulles del rellotge, aquest efecte quedà arreglat passant a girar 355° per representar el gir de -5°
- i finalment, després d'obtenir les imatges, calia desar-les a disc per poder treballar amb elles. Un cop capturat del buffer de representació de la imatge en pantalla, passat a buffer d'imatge convertint els píxels i reordenant-los, es pot gravar el fitxer de tipus imatge.

4.3. Generació i avaluació de les noves imatges

Una vegada analitzada la morfologia de les imatges i saber generar i desar les imatges, s'ha treballat a partir de la hipòtesi inicial de la generació per rotació de la imatge original.

A partir d'aquest punt es treballa amb les imatges de les set classes d'emocions bàsiques.

El resultat obtingut ha semblat curt en el sentit que amb només +/- 45° de gir en cada eix no aportava prou informació, i que els límits de gir del cap humà van més enllà.

S'ha ampliat el rang de proves, fins i tot mirant en quants graus cal incrementar cada gir consecutiu contemplat els següents escenaris:

- 1° d'increment en cada gir, la diferència entre les imatges és molt baixa i no aportem informació addicional
- 5°, és una bona mesura, tot i que els primers salts sembla que no aporten nova informació
- 7°, similar als 5°, no s'aprecia massa diferència
- 10°, ja comença a haver-hi un salt important, sobretot quan es combina el gir dels eixos X i Y. Les imatges ja són molt discretes.
- 15°, és un salt molt alt, les imatges perden semblança i penso que no aportaria en la finalitat que estem buscant

També s'ha ampliat el límit inferior i superior:

- Ampliació de -90° a +90°, clarament a partir de 70° les imatges generades aporten més soroll que informació, la imatge està molt deformada i reduïda.



Imatge 1 gir X_70 Y_70

- Si definim que el nou límit és -70° a +70°, un cop analitzades les combinacions dels dos eixos, clarament una rotació de 60° en cada eix donava una imatge poc visible:



Imatge 2: gir X_60 Y_60

- He treballat diferents angles i combinacions, destacar que el procés es primer girar sobre un eix i llavors girar sobre l'altre eix, és a dir, girem sobre l'eix X 0° (original) i llavors girem sobre l'eix Y de -70° (320°) fins a +70°.
- A continuació mostro una taula amb la projecció sobre el pla de 25 imatges generades per la rotació de la imatge original. He escollit

valors no simètrics per tal de donar més visibilitat de la mostra, entenent que els angles positius i negatius ens estaven donant una simetria respecte cadascun dels eixos.

Y X	-45°	-20°	0°	5°	60°
-45°					
-20°					
0°					
5°					
60°					

4.4. Generació massiva

Després de veure els resultats individuals, s'ha vist que treballar en el rang de -70° a $+70^\circ$ per cada eix aporta molt soroll: hi ha imatges amb molta informació "negre" i amb molt poca informació "facial" tal com es pot veure en la taula anterior.

També és molt important veure el número de fitxers que es generarà per cada imatge original. El final es tracta del producte

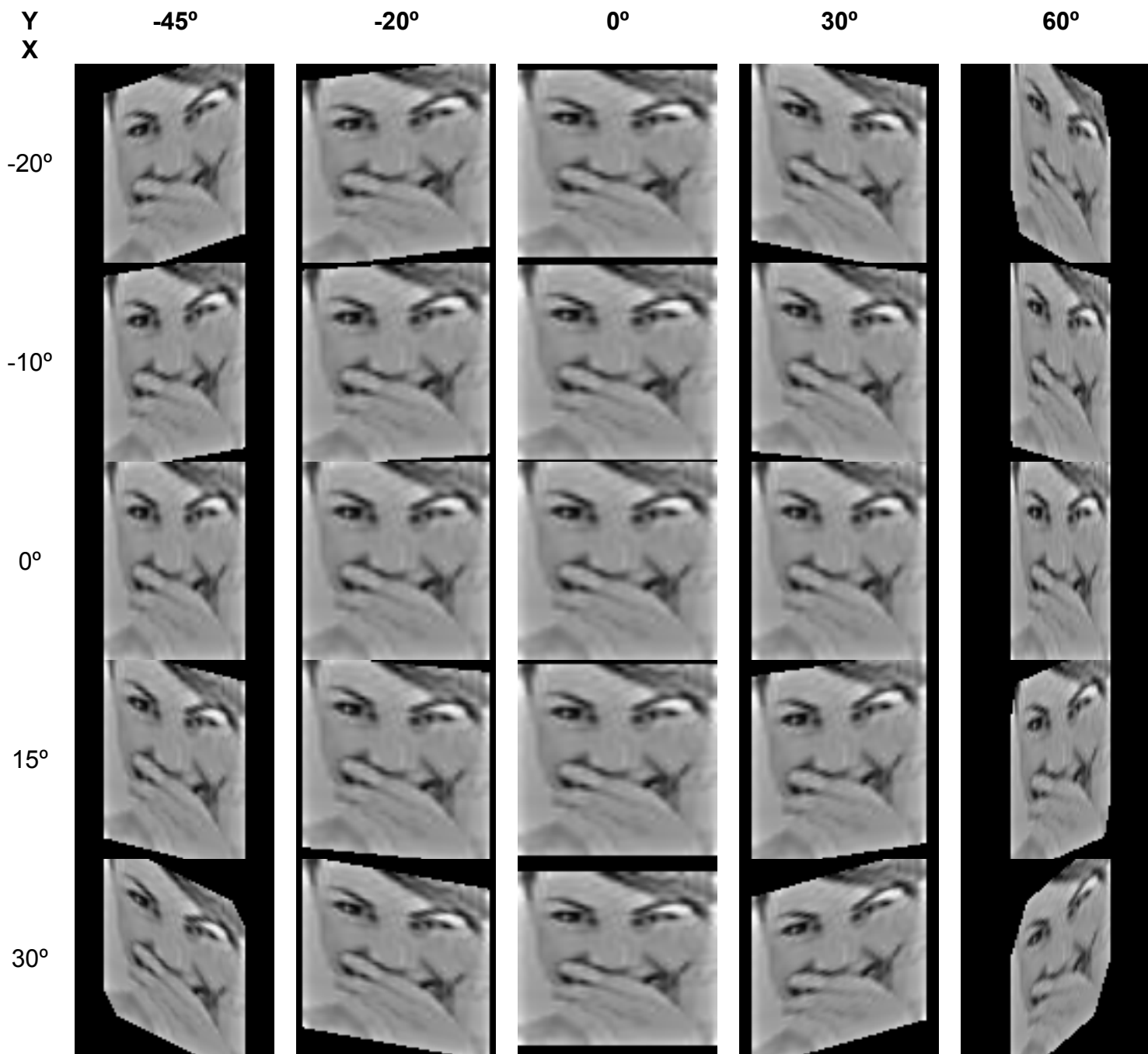
$$\frac{(\text{límit_super_X} - \text{límit_inferior_X})}{\text{angle}} * \frac{(\text{límit_super_Y} - \text{límit_inferior_Y})}{\text{angle}}$$

En el cas particular $\pm 70^\circ$ i 5° , tenim $140/5 * 140/5 = 28 * 28 = 784$ imatges per cadascuna de les originals, si plantegem ampliar la base de dades FER, passaríem de 28K a 22,5 milions d'imatges: el número d'imatges no és tractable i no és computable en un temps raonable.

Tenint en compte l'anàlisi fet i el número de fitxers a generar, partim d'un nou model més reduït generant imatges girades sobre l'eix Y de -60° a $+60^\circ$ i amb la gir sobre l'eix X de -30° a $+30^\circ$.

En aquest supòsit es generaran 324 imatges noves per cada imatge original.

En la següent taula hi ha els extrems de la representació i el punt mig:



En aquest escenari s'ha canviat la hipòtesi inicial de generar imatges amb un gir de -45° a $+45^\circ$ en els eixos X i Y passant a un gir de -60° a $+60^\circ$ respecte l'eix Y i un gir de -30° a $+30^\circ$ respecte l'eix X després d'analitzar el resultat de les imatges obtingudes.

Després de fer els primers entrenaments de xarxes neuronals convolucionals, s'ha detectat que tot i que la xarxa aprèn i convergeix, no classifica correctament. Encara tenim molta informació "negre" que fa que no millorem els resultats.

Igualment, s'ha vist que multiplicar per 325 el número d'imatges provoca que els conjunts d'entrenament es facin molt grans i el temps computacionals siguin elevadíssims.

Davant d'aquesta situació, es refà el sistema de generació d'imatges. Primerament s'ha d'eliminar la NO informació, "els negres", i prendre una determinació per generar menys fitxers. Per tant, es decideix passar d'un increment de 5° en cada nova generació d'imatges a 10°. Així, conservant els límits definits abans, es redueix el número d'imatges noves de 324 a 63.

Un dels requisits de les CNN és que totes les imatges tinguin la mateixa dimensió per tal de poder fer les convolucions i els productes de matrius, així que no podem agafar només el centre de la imatge, hem de de fer-hi un zoom. Per solucionar aquest punt, s'ha ampliat el frame i només s'ha convertit la part central de la imatge a fitxer obtenint molt menys negre en les imatges generades.

La generació massiva dels conjunts ampliat s'ha fet segons el disseny dels experiments explicats en el següent apartat, classificant-los seguint el tipus d'emoció segons la FER[3].

S'han generat un total de 441.000 imatges per l'execució dels experiments finals.

A continuació es pot veure un exemple amb les 63 imatges generades per cada imatge original:

Orig. No Zoom							
X / Y	30	20	10	0	350 (-10)	340 (-20)	330 (-30)
40							
30							
20							
10							
0							
350 (-10)							
340 (-20)							
330 (-30)							
320 (-40)							

4.5. Programa

L'objectiu del projecte no és generar una aplicació informàtica a l'ús, sinó codificar un programa com a eina per a obtenir la informació a analitzar i poder fer els canvis de la forma més eficient i ràpida possible en funció de l'avanç i l'anàlisi de les dades.

Per això s'ha generat un programa que mitjançant bucles i la configuració dels directoris d'entrada on hi ha les imatges originals, es generen les noves imatges en els directoris de sortida i una estadística del conjunt generat.

Aquest codi, fet en JAVA es pot veure en l'annex B, a destacar:

1. la classe Main és el programa principal i genera la classe generacio_imatges per cada fitxer origen que troba.
2. L'entrada del programa és un directori o un fitxer. Estar preparat per fer un bucle de 4 carpetes.
3. El tractament de l'entrada es fa en funció de:
 - a. Si és una carpeta, tracta tots els fitxers de la carpeta.
 - b. És un fitxer jpg, genera les imatges d'aquest fitxer únic
 - c. Si la carpeta conté fitxers de tipus txt, pressuposa que són les etiquetes tal com estan codificades en la FER[3], per cada tipus d'emoció bàsica, busca el fitxer origen en la seva carpeta i genera els fitxers en la carpeta corresponent (angry, sad, fair...)
 - d. Si la carpeta conté fitxers de tipus jpg, tracta tots els fitxers jpg i genera tots els fitxers a partir d'aquest jpg
4. La classe generació_imatges genera un frame i un entorn en openGL a partir de la classe Gir. Controla que s'acabi la generació de tots els fitxers
5. La classe gir, implementa el codi d 'OpenGL, en funció dels paràmetres genera més o menys imatges i per cada imatge generada passa el buffer de pantalla a un buffer d'imatge i acaba gravant la imatge a disc.

5. Comprovació del conjunt ampliat

5.1. Antecedents

En el capítol anterior s'ha analitzat i explicat la metodologia per generar noves imatges a partir d'un conjunt d'imatges existents classificades.

Recordar que el conjunt ampliat de dades està compost per les imatges generades a partir d'una imatge original amb una rotació de 10° en cada increment sobre l'eix Y entre els límits de -40° a $+40^\circ$ i amb la rotació sobre l'eix X de -30° a $+30^\circ$. Per tant es generaran 63 imatges per cada imatge original.

Tot aquest estudi es basa en l'entrenament de xarxes neuronals convolucionals (CNN), i a la pràctica ho hem fet amb l'ús del TENSORFLOW[4]

5.2. Instal·lació de les eines

El TensorFlow és una llibreria que es pot usar com una Api des de Python.

Aquí apareix un nou repte en el projecte. Fins aquest moment no havia treballat mai ni en Python ni amb eines avançades d'intel·ligència artificial. El primer pas ha estat entendre a què m'he d'enfrontar, instal·lar-ho i començar a treballar-hi.

Dir que he tingut dificultats per entendre el procés de Machine Learning i Deep Learning – en els qual es basa principalment l'entrenament de CNN – xarxes neuronals convolucionals, però amb l'ajuda del llibre HelloWorld en Tensor Flow d'en Jordi Torres he pogut seguir pas a pas com treballar amb TensorFlow amb l'exemple del MNIST i altres que m'han introduït en aquest món.

5.3. Proves prèvies

Fer menció que tot el procés de validació es fa a partir d'un programa d'entrenament de CNN codificat pel Dr. Gerard Pons el Juliol de 2016 que m'ha fet arribar el director del projecte. La principal feina ha estat entendre i adaptar aquest programa. En particular les inicialitzacions de les matrius convolucionals, el fet de no usar GPU's i les constants del programa. El programa resultant s'adjunta en l'Annex C.

Per validar que tot l'entorn funciona correctament i que està ben programat, s'han fet proves i desenvolupat eines paral·leles per comprovar que les conversions d'imatge a vector i canvis de forma en els tensors funcionaven correctament, com per exemple instal·lar la llibreria Pillow per poder visualitzar imatges des de Python.

Un cop superats alguns problemes d'instal·lació, de compilació i d'entesa de com funciona el programa he treballat amb la següent metodologia:

- Generació de les imatges de l'experiment i arxiu segons les carpetes per tipus d'imatge que preveia el codi
- Confeció dels conjunts d'imatges per entrenament i test
- Assegurar que les imatges es carreguen bé, que no hi ha problemes amb els tipus d'imatge, jpg, png, etc. Es valida fins el punt que es visualitzen totes les imatges que es tracten.
- Execució dels experiments

La prova definitiva ha estat entrar un CNN on les dades d'entrenament i test són les mateixes o part de les originals i s'han obtingut resultats d'accuracy de 99,6%. Això ens indica que el sistema funciona correctament.

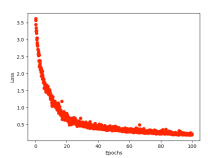
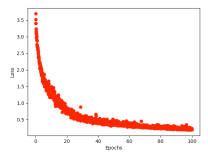
Es configura un primer conjunt d'experiments per poder validar que l'entorn i les dades generades són correctes. Tal com s'ha comentat en el capítol anterior, aquesta fase ens va portar a redefinir els experiments i a generar les noves imatges amb unes característiques diferents.

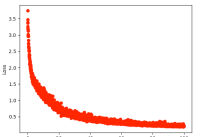
5.4. Primers experiments

El disseny dels primers experiments es va basar en la comparació de l'entrenament de les CNN a partir dels següents conjunts de dades:

- Un subconjunt o el conjunt complet d'imatges de la base de dades FER en el seu format original
- Un conjunt ampliat a partir de 20 imatges per cada tipus d'emoció bàsica. Un total de $7 \cdot 20 \cdot 324 = 45K$ imatges (cj A)
- Un conjunt ampliat a partir de 50 imatges per cada tipus d'emoció bàsica. Un total de $7 \cdot 50 \cdot 324 = 113K$ imatges (cj B)
- Un conjunt de test amb un subconjunt de les imatges originals de la base de dades FER amb 100 (cj C) o 1000 (cj D) imatges per cada emoció bàsica
- Els mateixos conjunts A, B, C i D amb les imatges amb zoom.

Amb aquests conjunts es van dissenyar els experiments que podem veure en la següent taula:

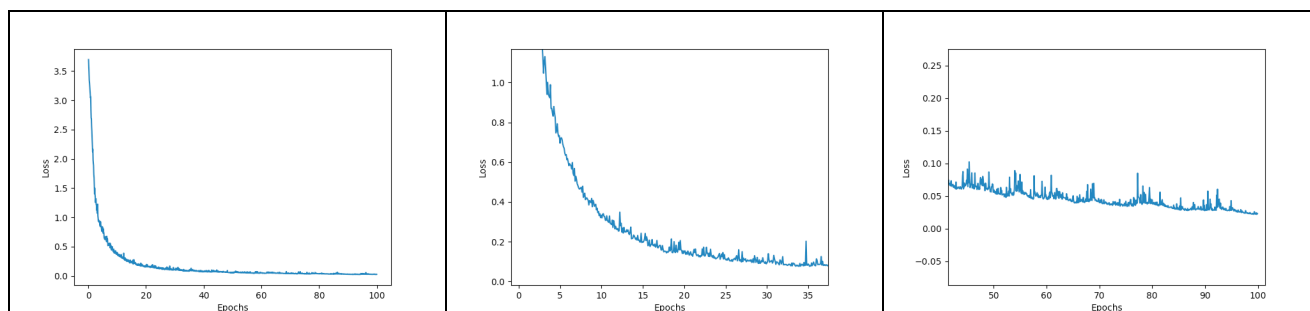
#	Descripció / Resultat	Train	Test	Accuracy	Loss/epochs
1	Imatges de train i test originals. Resultat esperat, encert molt alt.	12.000 imatges aleatòries originals de TRAIN de la base de dades FER	Subconjunt de 100 imatges originals de TRAIN per cada emoció bàsica de la base de dades FER	99,36%	
2	Imatges de train originals i test amb conjunt ampliat. Resultat, és una primera dada, no sabia fins aquest moment que passava amb imatges rotades, tot i provenir d'una imatges coneguda, original.	17.000 imatges aleatòries originals de TRAIN de la base de dades FER	20 imatges de cada classe amb la generació de de 63 imatges per cadascuna, sense zoom	36,74% (100% per les imatges originals)	

3	<p>Experiment com el #1 amb més imatges</p> <p>Resultat esperat, confirmo que el número d'imatges no ha de canviar el resultat, només el temps d'execució.</p>	<p>Totes les imatges originals de TRAIN de de la base de dades FER</p>	<p>Subconjunt de 1000 imatges originals de TRAIN de la base de dades FER</p>	<p>99,68%</p>	
4	<p>Altres combinacions entre els conjunts ampliats de 20 i 50 imatges</p> <p>Resultats diversos</p>			<p>Al voltant del 20%</p>	

Notar que aquests primers experiments van donar resultats no comparables, doncs es treballava amb conjunts de dades diferents, es tenien conjunts ampliats a partir de 20 i 50 imatges originals per cada tipus d'emoció.

Tot i no poder treballar sobre la hipòtesi del treball: el percentatge d'incert, es van poder treure un seguit de conclusions que van ajudar a definir el conjunt d'experiments definitiu.

Donat que els resultats no variaven, l'accuracy es mantenia en un valor del 20%. Penso que és degut a que la funció de cost tendia molt ràpid a zero i que no hi havia prou imatges en cada batch per fer més d'una iteració. A continuació hi ha unes mostres de la funció de cost en funció del número de epochs:



S'han modificat els experiments, posant èmfasi en el número d'imatges i la dimensió del batch, era necessari ampliar el número d'imatges i també diversificar-les més. Per això es decideix:

1. reduir l'angle de pas entre cada generació d'imatges, passant de 5° a 10° . Treballarem amb una quarta part de les imatges proposades.
2. Conservem els límits de gir

Vaig començar amb un experiment simple:

- Classificar totes les imatges de TRAIN de la base de dades FER, obtenint un total de 28718 imatges classificades en les següents categories:

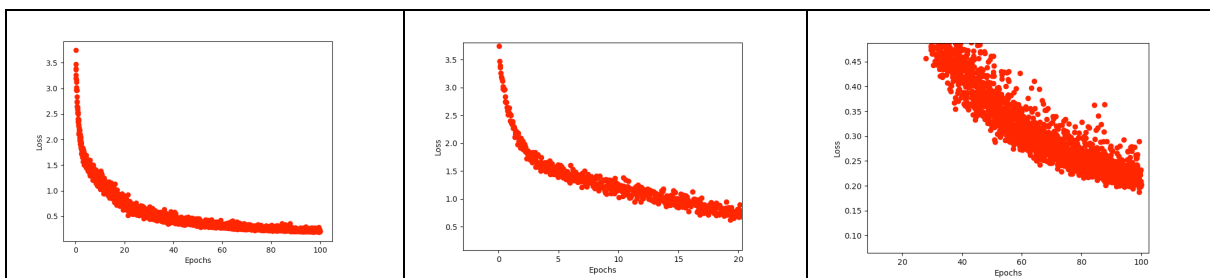
Classe	Quantitat
Angry	3.997
Disgust	436
Fear	4.098
Happy	7.218
Sad	4.831
Surprise	3.173
Neutral	4.965

- Utilitzar aquestes imatges com a conjunt de TRAIN.
- Agafar un subconjunt amb 1000 imatges de cada classe (emoció bàsica), excepte de Disgust, no n'hi ha més i entrar-la com un conjunt de TEST.
- El resultat és l'esperat, un encert molt elevat: 99,36%
- La matriu de confusió és la següent:

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	TOTAL
Angry	996	1	1		1			999
Disgust		432						432
Fear	1		995			1	1	998
Happy				999				999
Sad	2		3	1	999			1.008
Surprise		1	1			999		1.001
Neutral	1	2					996	999
TOTAL	1.000	436	1.000	1.000	1.000	1.000	1.000	6.436

	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	99,6%	1%	1%		1%		
Disgust		92,08%					
Fear	1%		99,5%			1%	1%
Happy				99,9%			
Sad	2%		3%	1%	99,9%		
Surprise		1%	1%			99,9%	
Neutral	1%	2%					99,6%

- Notar que aquest experiment es va repetir amb quantitats menors d'informació, i a partir d'un mínim d'imatges d'entrenament, el resultat és similar, al voltant del 99%.
- La funció de cost en aquest experiment varia respecte a la que hem vist abans, no en la forma, sinó en el número de punts que hi ha per cada epoch, això és degut al canvi de la dimensió del batch i del total d'imatges:



Els següents experiments seguien patrons similars de la funció de cost, en canvi el percentatge d'encert es quedava en un 20% aproximadament. En la mesura que va avançar el número de proves, em vaig adonar que les imatges girades tenen molta informació "no útil", un 50% de la imatge és negra. Aquest tipus d'imatges només aporta confusió. Totes les imatges rotades així es classifiquen igual independentment que la classe a la que pertany:











Imatge 3: gir X_50 Y_340

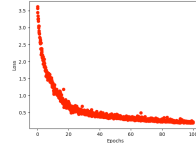
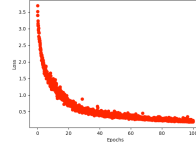
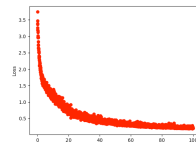
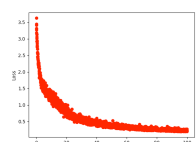
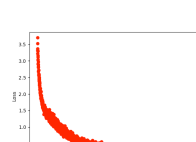
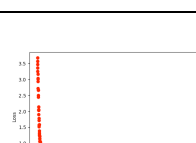
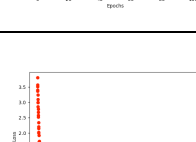
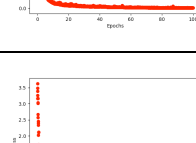
He treballat en l'eliminació de "el negre" de les imatges. Vaig concloure que calia redissenyar la generació d'imatges i buscar un compromís entre la quantitat de negre en la imatge girada i la pèrdua de les imatges poc girades. Seleccionar la part central de les imatges significa perdre informació en les imatges poc rotades en la mateixa proporció que l'eliminació de les parts negres: notar que no canvia la dimensió de la imatge, però si l'enquadrament, estem fent un zoom.

Les imatges definitives es van generar ampliant un 50% la imatge per tal de treure els "negres", una altra opció podia haver estat només ampliar un 30% i treure alguna rotació, descartat perquè encara quedava molt negre a partir d'una rotació.

En la taula següent poso uns exemples de com queden les imatges:

	Abans		Tractada		Abans		tractada
							
							

Amb aquest nou conjunt es van realitzar un seguit d'experiments per tal de verificar que el percentatge d'encert variava i que no s'empitjorava el resultat en els casos trivials. Així, com a pas previ al disseny i a l'execució del experiments finals, es van repetir els experiments anteriors i fer noves proves que es detallen en la següent taula resum:

#	Descripció / Resultat	Train	Test	Accuracy	Loss/epochs
1	Imatges de train i test originals. Resultat esperat, encert molt alt.	12.000 imatges aleatòries originals de TRAIN de de la base de dades FER	Subconjunt de 100 imatges originals de TRAIN per cada emoció bàsica de la base de dades FER	99,36%	
2	Imatges de train originals i test amb conjunt ampliat. Resultat, és una primera dada, no sabia fins aquest moment que passava amb imatges rotades, tot i provenir d'una imatges coneguda, original.	17.000 imatges aleatòries originals de TRAIN de de la base de dades FER	20 imatges de cada classe amb la generació de de 63 imatges per cadascuna, sense zoom	36,74% (100% per les imatges originals)	
3	Experiment com el #1 amb més imatges Resultat esperat, confirmo que el número d'imatges no ha de canviar el resultat, només el temps d'execució.	Totes les imatges originals de TRAIN de de la base de dades FER	Subconjunt de 1000 imatges originals de TRAIN de la base de dades FER	99,68%	
4	Imatges de train originals i imatges de test amb zoom Resultat, és una primera dada, es nota que el fet d'ampliar la imatge fa perdre informació rellevant i tot i que les imatges provenen d'una imatge original el percentatge es redueix.	Totes les imatges originals de TRAIN de de la base de dades FER	Subconjunt de 100 imatges per cada tipus d'emoció bàsica amb zoom aplicat	25,29%	
5	Experiment com el #2 canviant el conjunt de test on totes les imatges són amb zoom Resultat, alineat amb l'experiment anterior, el fet de fer zoom, aquestes imatges es classifiquen pitjor, per tant tenim una reducció del % respecte l'experiment #2.	Totes les imatges originals de TRAIN de de la base de dades FER	20 imatges de cada classe amb la generació de de 63 imatges per cadascuna, amb zoom per totes	28,02%	
6	Experiment invers al #5. Resultat, caic en la temptació de dir que el fet de perdre informació del zoom fa que classifiqui pitjor, no obstant, aquí estem treballant amb un conjunt d'entrenament reduït respecte els experiments anteriors.	50 imatges de cada classe amb la generació de de 63 imatges per cadascuna, amb zoom per totes	100 imatges originals per classe de TRAIN de de la base de dades FER amb zoom	24,28%	
7	Experiment amb imatges de train i test amb zoom i el seus corresponents conjunts ampliat. Resultat alineat amb l'experiment #5. Notar que l'entrenament es fa amb menys diversitat d'imatges	50 imatges de cada classe amb la generació de de 63 imatges per cadascuna, amb zoom per totes	20 imatges de cada classe amb la generació de de 63 imatges per cadascuna, amb zoom per totes – conjunt disjunt	25,01%	
8	Experiment com el #1. Imatges de train i test amb zoom i els seus corresponents conjunts ampliat. Les imatges de test són un subconjunt de les de train. Resultat esperat, encert molt alt.	50 imatges de cada classe amb la generació de de 63 imatges per cadascuna, amb zoom per totes	20 imatges de cada classe amb la generació de de 63 imatges per cadascuna, amb zoom per totes – subconjunt de les 50	100%	

5.5. Disseny de l'experiment

La metodologia de control i valoració de la hipòtesi es basa en el disseny de diversos experiments amb l'objectiu d'entrenar una xarxa neuronal convolucional, CNN, amb un conjunt d'imatges originals i el seu

corresponent conjunt ampliat i poder-ne comparar el percentatge d'encert per un mateix conjunt de test.

Un cop vistes les proves preliminars, els primers experiments, redefinir i regenerar els conjunts ampliat passem a definir l'experiment. Notar que a partir d'aquest moment totes les imatges amb les que es treballa tenen el zoom aplicat. Els conjunts d'imatges de test i d'imatges de train són disjunts, és a dir, no tenen cap imatge en comú en els conjunts.

Anomenem experiment al conjunt de 4 subexperiments coherents entre ells que ens donaran resultats en funció del tipus de dades d'entrenament i de test.

Els 4 subexperiments són amb les mateixes imatges:

1. entrenament i test amb imatges originals
 - a. 100 imatges de train per emoció bàsica: 700 imatges de train
 - b. 100 imatges de test per emoció bàsica: 700 imatges de test
2. entrenament amb conjunt ampliat exp 1 i test amb imatges originals de l'exp 1
 - a. 100 imatges de train per emoció bàsica ampliat (*63): 44K imatges de train
 - b. 100 imatges de test per emoció bàsica (no es repeteixen) : 700 imatges de test
3. entrenament imatges originals idem exp 1 i test amb conjunt ampliat de l'exp 1
 - a. 100 imatges de train per emoció bàsica: 700 imatges de train
 - b. 100 imatges de test per emoció bàsica ampliat (*63): 44K imatges de test
4. entrenament amb conjunt ampliat exp 1 i test amb conjunt ampliat de l'exp 1
 - a. 100 imatges de train per emoció bàsica ampliat (*63): 44K imatges de train
 - b. 100 imatges de test per emoció bàsica ampliat (*63): 44K imatges de test

Aquests experiments s'han repetit 10 vegades amb conjunts de dades d'entrada diferents cada vegada i analitzar els percentatges d'encert obtinguts individualment, avaluar la mitjana d'aquests i confirmar la millora esperada. L'objectiu era veure que els intervals de confiança per un tipus d'entrenament de CNN i per l'altra no es solapaven.

Els experiments 5 i 10 només tenen 36 imatges de la classe disgust i la seva corresponent ampliació 36*63 en lloc de les 100 que tenen les altres classes.

Tots els experiments tenen imatges diferents en totes les classes, excepte en la classe disgust on les imatges dels experiments 1, 2, 3, 4 i 5 es repeteixen en el 6, 7, 8, 9 i 10 respectivament.

La metodologia per generar els conjunts de train i test a partir de la base de dades FER per fer els experiments és la següent.

Donat que hi ha més de 2.000 imatges per cada classe (tipus d'emoció bàsica) excepte en el cas de disgust, i els fitxers estan numerats de l'1 al 28.718, per construir els conjunts pels 10 experiments he escollit 100 imatges de cada classe de forma ascendent per formar el conjunt de train. En canvi, la selecció d'imatges pel conjunt de test s'ha fet de forma descendent, assegurant que no hi ha cap imatge repetida en cap classe. Recordar que les imatges de la base de dades FER estan repartides de la següent manera:

Classe	Quantitat
Angry	3.997
Disgust	436
Fear	4.098
Happy	7.218
Sad	4.831
Surprise	3.173
Neutral	4.965

Així, el disseny final de l'experiment consisteix en executar l'entrenament de CNN 40 vegades segons els diferents conjunts de dades de train i test definits anteriorment.

El percentatge d'encert es mira amb el nivell de certesa que classifica les imatges de TEST (accuracy). En cada experiment he confeccionat la matriu de confusió per poder comprovar el % d'encert i també per veure si hi ha algun comportament similar.

5.6. Resultats

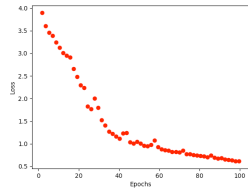
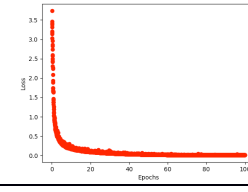
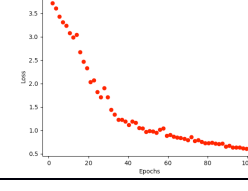
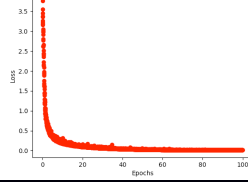
Un cop dissenyats els experiments i tenir les dades preparades, només cal entrenar la CNN i fer el procés de test.

Per cada subexperiment he recollit la informació de les imatges classificades correctament i les que no per poder fer les matrius de confusió per cada experiment. He volgut controlar el cas particular de les imatges originals dins dels conjunts ampliat per veure si es mantenia el mateix percentatge d'encert que sense el conjunt ampliat. Notar que els percentatges es mantenen en aquest cas.

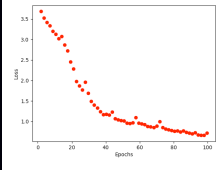
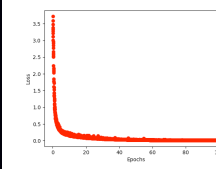
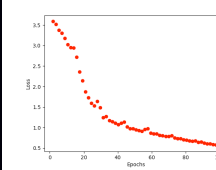
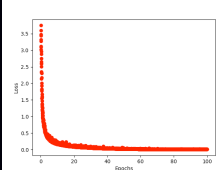
En general, com es pot veure en els resultats dels 10 experiments, l'encert en el reconeixement millora en el cas de les xarxes entrenades amb els conjunts ampliat. No obstant, es pot comprovar que no hi ha una correlació entre els percentatges d'encert per cadascuna de les classes entre la CNN entrenada amb conjunt ampliat o conjunt original.

La funció de cost convergeix molt més ràpidament en els conjunts ampliat tal com es pot veure en les gràfiques de cada subexperiment- experiment.

5.6.1. EXPERIMENT 1

#	Descripció	Matriu de confusió	Accuracy	Loss/epochs																																																																
1	entrenament i test amb imatges originals (700 imatges)	<table border="1"> <thead> <tr> <th></th> <th>Angry</th> <th>Disgust</th> <th>Fear</th> <th>Happy</th> <th>Sad</th> <th>Surprise</th> <th>Neutral</th> </tr> </thead> <tbody> <tr> <th>Angry</th> <td>22%</td> <td>9%</td> <td>13%</td> <td>11%</td> <td>16%</td> <td>8%</td> <td>14%</td> </tr> <tr> <th>Disgust</th> <td>12%</td> <td>32%</td> <td>6%</td> <td>7%</td> <td>8%</td> <td>6%</td> <td>10%</td> </tr> <tr> <th>Fear</th> <td>22%</td> <td>16%</td> <td>14%</td> <td>7%</td> <td>15%</td> <td>12%</td> <td>15%</td> </tr> <tr> <th>Happy</th> <td>6%</td> <td>12%</td> <td>16%</td> <td>44%</td> <td>15%</td> <td>9%</td> <td>22%</td> </tr> <tr> <th>Sad</th> <td>23%</td> <td>15%</td> <td>19%</td> <td>13%</td> <td>24%</td> <td>6%</td> <td>11%</td> </tr> <tr> <th>Surprise</th> <td>7%</td> <td>6%</td> <td>22%</td> <td>5%</td> <td>5%</td> <td>51%</td> <td>13%</td> </tr> <tr> <th>Neutral</th> <td>8%</td> <td>10%</td> <td>10%</td> <td>13%</td> <td>17%</td> <td>8%</td> <td>15%</td> </tr> </tbody> </table>		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	Angry	22%	9%	13%	11%	16%	8%	14%	Disgust	12%	32%	6%	7%	8%	6%	10%	Fear	22%	16%	14%	7%	15%	12%	15%	Happy	6%	12%	16%	44%	15%	9%	22%	Sad	23%	15%	19%	13%	24%	6%	11%	Surprise	7%	6%	22%	5%	5%	51%	13%	Neutral	8%	10%	10%	13%	17%	8%	15%	28,86%	
	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral																																																													
Angry	22%	9%	13%	11%	16%	8%	14%																																																													
Disgust	12%	32%	6%	7%	8%	6%	10%																																																													
Fear	22%	16%	14%	7%	15%	12%	15%																																																													
Happy	6%	12%	16%	44%	15%	9%	22%																																																													
Sad	23%	15%	19%	13%	24%	6%	11%																																																													
Surprise	7%	6%	22%	5%	5%	51%	13%																																																													
Neutral	8%	10%	10%	13%	17%	8%	15%																																																													
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)	<table border="1"> <thead> <tr> <th></th> <th>Angry</th> <th>Disgust</th> <th>Fear</th> <th>Happy</th> <th>Sad</th> <th>Surprise</th> <th>Neutral</th> </tr> </thead> <tbody> <tr> <th>Angry</th> <td>15%</td> <td>6%</td> <td>11%</td> <td>10%</td> <td>14%</td> <td>6%</td> <td>12%</td> </tr> <tr> <th>Disgust</th> <td>20%</td> <td>48%</td> <td>19%</td> <td>15%</td> <td>10%</td> <td>5%</td> <td>12%</td> </tr> <tr> <th>Fear</th> <td>8%</td> <td>12%</td> <td>16%</td> <td>9%</td> <td>15%</td> <td>14%</td> <td>13%</td> </tr> <tr> <th>Happy</th> <td>15%</td> <td>10%</td> <td>8%</td> <td>42%</td> <td>12%</td> <td>2%</td> <td>15%</td> </tr> <tr> <th>Sad</th> <td>10%</td> <td>8%</td> <td>13%</td> <td>11%</td> <td>23%</td> <td>8%</td> <td>13%</td> </tr> <tr> <th>Surprise</th> <td>12%</td> <td>6%</td> <td>20%</td> <td>6%</td> <td>10%</td> <td>55%</td> <td>12%</td> </tr> <tr> <th>Neutral</th> <td>20%</td> <td>10%</td> <td>13%</td> <td>7%</td> <td>16%</td> <td>10%</td> <td>23%</td> </tr> </tbody> </table>		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	Angry	15%	6%	11%	10%	14%	6%	12%	Disgust	20%	48%	19%	15%	10%	5%	12%	Fear	8%	12%	16%	9%	15%	14%	13%	Happy	15%	10%	8%	42%	12%	2%	15%	Sad	10%	8%	13%	11%	23%	8%	13%	Surprise	12%	6%	20%	6%	10%	55%	12%	Neutral	20%	10%	13%	7%	16%	10%	23%	31,71%	
	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral																																																													
Angry	15%	6%	11%	10%	14%	6%	12%																																																													
Disgust	20%	48%	19%	15%	10%	5%	12%																																																													
Fear	8%	12%	16%	9%	15%	14%	13%																																																													
Happy	15%	10%	8%	42%	12%	2%	15%																																																													
Sad	10%	8%	13%	11%	23%	8%	13%																																																													
Surprise	12%	6%	20%	6%	10%	55%	12%																																																													
Neutral	20%	10%	13%	7%	16%	10%	23%																																																													
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)	<table border="1"> <thead> <tr> <th></th> <th>Angry</th> <th>Disgust</th> <th>Fear</th> <th>Happy</th> <th>Sad</th> <th>Surprise</th> <th>Neutral</th> </tr> </thead> <tbody> <tr> <th>Angry</th> <td>28,13%</td> <td>24,76%</td> <td>22,11%</td> <td>23,30%</td> <td>24,76%</td> <td>15,35%</td> <td>19,95%</td> </tr> <tr> <th>Disgust</th> <td>8,75%</td> <td>21,41%</td> <td>5,98%</td> <td>6,78%</td> <td>6,62%</td> <td>5,19%</td> <td>5,62%</td> </tr> <tr> <th>Fear</th> <td>13,75%</td> <td>11,68%</td> <td>12,10%</td> <td>8,95%</td> <td>12,22%</td> <td>13,30%</td> <td>12,95%</td> </tr> <tr> <th>Happy</th> <td>13,57%</td> <td>14,48%</td> <td>22,41%</td> <td>39,83%</td> <td>19,70%</td> <td>13,03%</td> <td>26,94%</td> </tr> <tr> <th>Sad</th> <td>17,27%</td> <td>10,67%</td> <td>9,19%</td> <td>7,60%</td> <td>15,51%</td> <td>5,06%</td> <td>10,95%</td> </tr> <tr> <th>Surprise</th> <td>9,78%</td> <td>10,30%</td> <td>16,30%</td> <td>6,24%</td> <td>9,46%</td> <td>40,06%</td> <td>12,05%</td> </tr> <tr> <th>Neutral</th> <td>8,76%</td> <td>6,70%</td> <td>11,90%</td> <td>7,30%</td> <td>11,73%</td> <td>8,00%</td> <td>11,54%</td> </tr> </tbody> </table>		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	Angry	28,13%	24,76%	22,11%	23,30%	24,76%	15,35%	19,95%	Disgust	8,75%	21,41%	5,98%	6,78%	6,62%	5,19%	5,62%	Fear	13,75%	11,68%	12,10%	8,95%	12,22%	13,30%	12,95%	Happy	13,57%	14,48%	22,41%	39,83%	19,70%	13,03%	26,94%	Sad	17,27%	10,67%	9,19%	7,60%	15,51%	5,06%	10,95%	Surprise	9,78%	10,30%	16,30%	6,24%	9,46%	40,06%	12,05%	Neutral	8,76%	6,70%	11,90%	7,30%	11,73%	8,00%	11,54%	24,08% (originals 28,85%)	
	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral																																																													
Angry	28,13%	24,76%	22,11%	23,30%	24,76%	15,35%	19,95%																																																													
Disgust	8,75%	21,41%	5,98%	6,78%	6,62%	5,19%	5,62%																																																													
Fear	13,75%	11,68%	12,10%	8,95%	12,22%	13,30%	12,95%																																																													
Happy	13,57%	14,48%	22,41%	39,83%	19,70%	13,03%	26,94%																																																													
Sad	17,27%	10,67%	9,19%	7,60%	15,51%	5,06%	10,95%																																																													
Surprise	9,78%	10,30%	16,30%	6,24%	9,46%	40,06%	12,05%																																																													
Neutral	8,76%	6,70%	11,90%	7,30%	11,73%	8,00%	11,54%																																																													
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)	<table border="1"> <thead> <tr> <th></th> <th>Angry</th> <th>Disgust</th> <th>Fear</th> <th>Happy</th> <th>Sad</th> <th>Surprise</th> <th>Neutral</th> </tr> </thead> <tbody> <tr> <th>Angry</th> <td>16,87%</td> <td>6,90%</td> <td>10,90%</td> <td>10,40%</td> <td>16,65%</td> <td>6,62%</td> <td>10,08%</td> </tr> <tr> <th>Disgust</th> <td>16,35%</td> <td>39,59%</td> <td>11,22%</td> <td>12,21%</td> <td>10,10%</td> <td>5,13%</td> <td>7,63%</td> </tr> <tr> <th>Fear</th> <td>14,11%</td> <td>12,70%</td> <td>16,95%</td> <td>8,17%</td> <td>11,43%</td> <td>18,03%</td> <td>16,63%</td> </tr> <tr> <th>Happy</th> <td>11,49%</td> <td>10,87%</td> <td>9,71%</td> <td>43,13%</td> <td>12,48%</td> <td>6,35%</td> <td>14,97%</td> </tr> <tr> <th>Sad</th> <td>18,24%</td> <td>13,83%</td> <td>17,62%</td> <td>11,37%</td> <td>29,03%</td> <td>8,29%</td> <td>16,94%</td> </tr> <tr> <th>Surprise</th> <td>8,11%</td> <td>6,67%</td> <td>21,57%</td> <td>6,73%</td> <td>6,70%</td> <td>47,25%</td> <td>11,06%</td> </tr> <tr> <th>Neutral</th> <td>14,83%</td> <td>9,44%</td> <td>12,02%</td> <td>8,00%</td> <td>13,62%</td> <td>8,33%</td> <td>22,68%</td> </tr> </tbody> </table>		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	Angry	16,87%	6,90%	10,90%	10,40%	16,65%	6,62%	10,08%	Disgust	16,35%	39,59%	11,22%	12,21%	10,10%	5,13%	7,63%	Fear	14,11%	12,70%	16,95%	8,17%	11,43%	18,03%	16,63%	Happy	11,49%	10,87%	9,71%	43,13%	12,48%	6,35%	14,97%	Sad	18,24%	13,83%	17,62%	11,37%	29,03%	8,29%	16,94%	Surprise	8,11%	6,67%	21,57%	6,73%	6,70%	47,25%	11,06%	Neutral	14,83%	9,44%	12,02%	8,00%	13,62%	8,33%	22,68%	30,79% (originals 31%)	
	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral																																																													
Angry	16,87%	6,90%	10,90%	10,40%	16,65%	6,62%	10,08%																																																													
Disgust	16,35%	39,59%	11,22%	12,21%	10,10%	5,13%	7,63%																																																													
Fear	14,11%	12,70%	16,95%	8,17%	11,43%	18,03%	16,63%																																																													
Happy	11,49%	10,87%	9,71%	43,13%	12,48%	6,35%	14,97%																																																													
Sad	18,24%	13,83%	17,62%	11,37%	29,03%	8,29%	16,94%																																																													
Surprise	8,11%	6,67%	21,57%	6,73%	6,70%	47,25%	11,06%																																																													
Neutral	14,83%	9,44%	12,02%	8,00%	13,62%	8,33%	22,68%																																																													

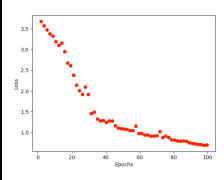
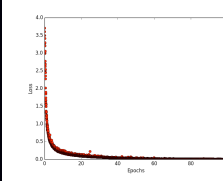
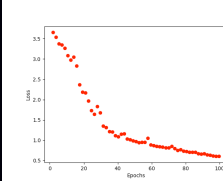
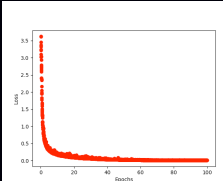
5.6.2. EXPERIMENT 2

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
			Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral		
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	30,64%	
		Angry	25,51%	13,00%	21,21%	12,24%	15,15%	15,31%	11,00%		
		Disgust	18,37%	43,00%	18,18%	10,20%	14,14%	8,16%	15,00%		
		Fear	5,10%	8,00%	12,12%	6,12%	8,08%	11,22%	7,00%		
		Happy	16,33%	8,00%	17,17%	36,73%	11,11%	7,14%	12,00%		
		Sad	21,43%	11,00%	12,12%	15,31%	32,32%	7,14%	23,00%		
		Surprise	8,16%	11,00%	11,11%	13,27%	9,09%	45,92%	13,00%		
		Neutral	5,10%	6,00%	8,08%	6,12%	10,10%	5,10%	19,00%		
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	29,62%	
		Angry	17,35%	12,00%	13,13%	14,29%	9,09%	10,20%	14,00%		
		Disgust	20,41%	51,00%	14,14%	10,20%	12,12%	4,08%	14,00%		
		Fear	19,39%	6,00%	21,21%	13,27%	16,16%	21,43%	12,00%		
		Happy	11,22%	6,00%	13,13%	27,55%	13,13%	2,04%	8,00%		
		Sad	11,22%	7,00%	16,16%	12,24%	23,23%	4,08%	16,00%		
		Surprise	8,16%	6,00%	11,11%	8,16%	10,10%	41,84%	11,00%		
		Neutral	12,24%	12,00%	11,11%	14,29%	16,16%	16,33%	25,00%		
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	26,70% (originals 30,64%)	
		Angry	24,73%	18,33%	19,26%	22,43%	14,41%	14,35%	18,52%		
		Disgust	7,82%	23,89%	6,30%	3,69%	6,56%	3,63%	4,75%		
		Fear	9,20%	14,68%	19,50%	9,36%	14,94%	16,99%	8,83%		
		Happy	20,36%	15,11%	16,74%	37,59%	22,83%	9,85%	20,81%		
		Sad	20,44%	14,19%	23,87%	15,47%	25,54%	9,30%	24,08%		
		Surprise	8,96%	6,73%	9,38%	5,09%	7,84%	40,12%	7,10%		
		Neutral	8,49%	7,06%	4,95%	6,37%	7,87%	5,77%	15,92%		
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	30,479% (originals 29%)	
		Angry	22,09%	9,81%	12,94%	14,32%	13,76%	15,42%	16,29%		
		Disgust	15,57%	50,24%	12,38%	6,49%	8,67%	3,97%	10,84%		
		Fear	16,65%	6,44%	18,26%	14,84%	18,87%	17,57%	8,22%		
		Happy	12,08%	6,92%	8,96%	31,81%	11,42%	4,37%	13,86%		
		Sad	15,65%	9,62%	14,03%	13,09%	21,26%	3,17%	15,19%		
		Surprise	9,43%	7,08%	18,04%	9,65%	11,82%	46,84%	12,83%		
		Neutral	8,54%	9,89%	15,39%	9,80%	14,21%	8,65%	22,78%		

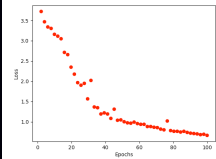
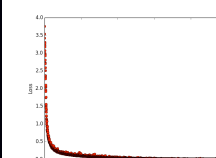
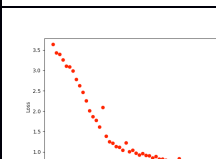
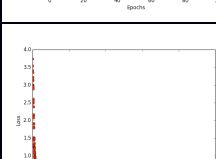
5.6.3. EXPERIMENT 3

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
			Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral		
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	31,33%	
		Angry	17,00%	11,00%	21,00%	12,12%	22,00%	11,00%	10,00%		
		Disgust	29,00%	58,00%	12,00%	10,10%	9,00%	4,00%	13,00%		
		Fear	11,00%	4,00%	11,00%	9,09%	12,00%	13,00%	13,00%		
		Happy	14,00%	7,00%	12,00%	38,38%	11,00%	8,00%	15,00%		
		Sad	14,00%	10,00%	18,00%	12,12%	28,00%	7,00%	16,00%		
		Surprise	13,00%	5,00%	20,00%	9,09%	11,00%	47,00%	13,00%		
Neutral	2,00%	5,00%	6,00%	9,09%	7,00%	10,00%	20,00%				
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	34,91%	
		Angry	27,00%	3,00%	17,00%	17,17%	19,00%	4,00%	14,00%		
		Disgust	19,00%	69,00%	15,00%	8,08%	14,00%	6,00%	12,00%		
		Fear	12,00%	6,00%	9,00%	7,07%	11,00%	11,00%	10,00%		
		Happy	9,00%	3,00%	4,00%	35,35%	6,00%	1,00%	14,00%		
		Sad	7,00%	7,00%	17,00%	11,11%	23,00%	10,00%	12,00%		
		Surprise	17,00%	3,00%	23,00%	12,12%	10,00%	51,00%	8,00%		
Neutral	9,00%	9,00%	15,00%	9,09%	17,00%	17,00%	30,00%				
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	27,36% (originals 32,19%)	
		Angry	29,21%	22,59%	30,63%	19,48%	26,10%	15,02%	21,11%		
		Disgust	14,06%	36,00%	5,44%	6,09%	8,86%	3,83%	6,94%		
		Fear	7,83%	5,57%	9,03%	5,77%	6,83%	17,16%	4,57%		
		Happy	20,57%	16,05%	21,29%	45,15%	23,92%	16,00%	26,19%		
		Sad	9,79%	11,37%	13,75%	8,48%	17,13%	9,11%	13,24%		
		Surprise	8,25%	2,51%	10,60%	4,91%	6,90%	33,33%	6,13%		
Neutral	10,29%	5,92%	9,25%	10,12%	10,27%	5,56%	21,83%				
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	34,10% (originals 34,48%)	
		Angry	24,25%	12,60%	17,56%	11,78%	15,46%	5,30%	17,51%		
		Disgust	20,19%	63,41%	10,43%	9,33%	9,73%	4,21%	7,32%		
		Fear	9,51%	6,56%	11,97%	7,52%	12,48%	7,25%	12,60%		
		Happy	11,62%	3,21%	11,38%	35,96%	12,95%	7,86%	14,95%		
		Sad	8,59%	5,54%	14,29%	10,15%	22,13%	5,90%	11,22%		
		Surprise	15,19%	1,11%	20,14%	12,17%	9,48%	54,41%	9,81%		
Neutral	10,65%	7,57%	14,24%	13,08%	17,78%	15,06%	26,59%				

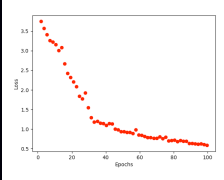
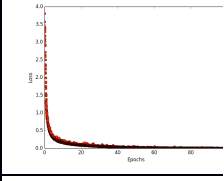
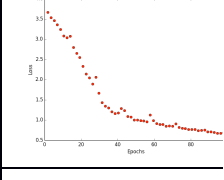
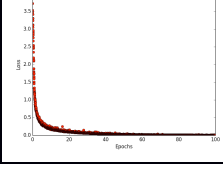
5.6.4. EXPERIMENT 4

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	29,57%	
		Angry	25,00%	17,00%	14,00%	15,00%	20,00%	8,00%	14,00%		
		Disgust	8,00%	31,00%	7,00%	8,00%	5,00%	6,00%	15,00%		
		Fear	12,00%	25,00%	23,00%	14,00%	8,00%	17,00%	12,00%		
		Happy	10,00%	5,00%	3,00%	28,00%	9,00%	2,00%	9,00%		
		Sad	14,00%	6,00%	22,00%	18,00%	28,00%	9,00%	13,00%		
		Surprise	8,00%	5,00%	11,00%	5,00%	6,00%	41,00%	6,00%		
		Neutral	23,00%	11,00%	20,00%	12,00%	24,00%	17,00%	31,00%		
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	31,00%	
		Angry	22,00%	13,00%	11,00%	11,00%	16,00%	10,00%	14,00%		
		Disgust	18,00%	35,00%	7,00%	10,00%	12,00%	10,00%	13,00%		
		Fear	16,00%	15,00%	31,00%	14,00%	5,00%	14,00%	8,00%		
		Happy	10,00%	11,00%	6,00%	37,00%	12,00%	4,00%	16,00%		
		Sad	10,00%	11,00%	21,00%	12,00%	27,00%	9,00%	17,00%		
		Surprise	8,00%	9,00%	14,00%	9,00%	12,00%	43,00%	10,00%		
		Neutral	16,00%	6,00%	10,00%	7,00%	16,00%	10,00%	22,00%		
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	27,17% (originals 31,71%)	
		Angry	17,71%	12,05%	15,97%	8,86%	14,56%	11,84%	13,65%		
		Disgust	16,40%	34,32%	14,67%	11,49%	11,05%	10,81%	20,87%		
		Fear	4,84%	6,37%	12,62%	7,68%	9,22%	8,37%	6,87%		
		Happy	22,54%	19,92%	18,21%	42,95%	21,75%	11,10%	19,67%		
		Sad	12,86%	9,21%	18,05%	14,32%	25,95%	10,49%	14,00%		
		Surprise	6,57%	8,38%	10,13%	6,68%	5,19%	38,33%	6,67%		
		Neutral	19,08%	9,76%	10,37%	8,02%	12,29%	9,06%	18,27%		
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	32,02% (originals 32,71%)	
		Angry	18,65%	11,84%	13,70%	9,25%	14,62%	10,51%	12,81%		
		Disgust	14,89%	38,02%	9,24%	8,27%	10,97%	5,94%	14,95%		
		Fear	13,13%	17,30%	20,89%	11,73%	7,89%	11,32%	12,44%		
		Happy	16,70%	11,10%	10,17%	44,10%	11,84%	5,38%	12,51%		
		Sad	10,44%	9,35%	21,83%	11,92%	31,92%	10,48%	12,11%		
		Surprise	10,02%	6,05%	14,52%	5,98%	7,90%	48,32%	12,92%		
		Neutral	16,17%	6,35%	9,65%	8,75%	14,86%	8,06%	22,25%		

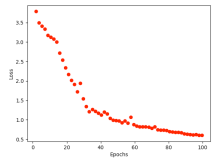
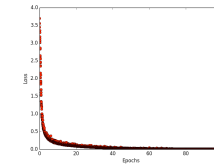
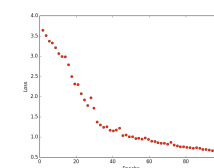
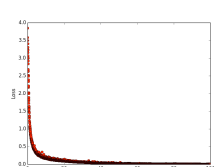
5.6.5. EXPERIMENT 5

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
			Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral		
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	23,90%	
		Angry	23,00%	22,22%	23,00%	21,00%	25,00%	12,00%	25,00%		
		Disgust	9,00%	8,33%	3,00%	2,00%	3,00%	0,00%	4,00%		
		Fear	12,00%	11,11%	16,00%	13,00%	13,00%	10,00%	8,00%		
		Happy	19,00%	11,11%	13,00%	29,00%	19,00%	13,00%	20,00%		
		Sad	13,00%	22,22%	20,00%	8,00%	13,00%	9,00%	12,00%		
		Surprise	10,00%	8,33%	14,00%	6,00%	7,00%	45,00%	8,00%		
		Neutral	14,00%	16,67%	11,00%	21,00%	20,00%	11,00%	23,00%		
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	27,67%	
		Angry	22,00%	13,89%	14,00%	11,00%	11,00%	9,00%	17,00%		
		Disgust	2,00%	13,89%	6,00%	3,00%	4,00%	3,00%	2,00%		
		Fear	10,00%	5,56%	12,00%	6,00%	11,00%	11,00%	9,00%		
		Happy	22,00%	13,89%	16,00%	34,00%	22,00%	6,00%	17,00%		
		Sad	11,00%	22,22%	20,00%	15,00%	27,00%	5,00%	17,00%		
		Surprise	15,00%	11,11%	16,00%	20,00%	12,00%	53,00%	15,00%		
		Neutral	18,00%	19,44%	16,00%	11,00%	13,00%	13,00%	23,00%		
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	26,02% (originals 30,19%)	
		Angry	22,29%	21,52%	15,75%	11,68%	17,46%	8,40%	14,08%		
		Disgust	2,32%	5,69%	2,05%	1,30%	2,84%	1,05%	1,25%		
		Fear	8,29%	13,18%	14,90%	14,37%	12,02%	9,97%	8,78%		
		Happy	15,43%	5,16%	10,46%	29,46%	14,08%	10,76%	12,02%		
		Sad	13,30%	19,00%	18,25%	15,51%	22,03%	8,98%	17,43%		
		Surprise	18,54%	17,33%	23,14%	12,29%	11,40%	40,63%	12,32%		
		Neutral	19,84%	18,12%	15,44%	15,40%	20,17%	20,21%	34,13%		
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	28,55% (originals 27,67%)	
		Angry	23,21%	16,09%	16,71%	11,90%	14,60%	8,52%	12,63%		
		Disgust	2,65%	11,20%	2,65%	2,38%	2,54%	1,52%	2,57%		
		Fear	17,44%	13,18%	22,86%	15,22%	17,71%	16,84%	13,24%		
		Happy	18,83%	10,98%	17,17%	35,90%	18,56%	9,19%	14,41%		
		Sad	12,02%	22,27%	14,90%	11,86%	22,21%	5,33%	16,86%		
		Surprise	12,06%	9,79%	11,38%	11,46%	7,32%	47,56%	14,46%		
		Neutral	13,79%	16,49%	14,32%	11,27%	17,06%	11,03%	25,83%		

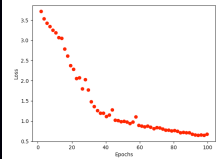
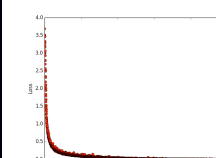
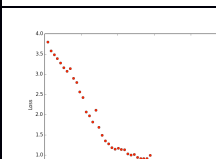
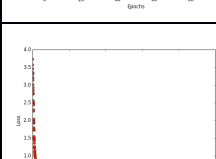
5.6.6. EXPERIMENT 6

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
			Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral		
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	27,14%	
		Angry	8,00%	5,00%	12,00%	6,00%	12,00%	4,00%	10,00%		
		Disgust	7,00%	33,00%	5,00%	4,00%	6,00%	6,00%	10,00%		
		Fear	11,00%	8,00%	9,00%	12,00%	11,00%	10,00%	7,00%		
		Happy	11,00%	14,00%	11,00%	37,00%	8,00%	5,00%	10,00%		
		Sad	20,00%	12,00%	20,00%	14,00%	22,00%	13,00%	17,00%		
		Surprise	12,00%	15,00%	16,00%	8,00%	17,00%	43,00%	8,00%		
		Neutral	31,00%	13,00%	27,00%	19,00%	24,00%	19,00%	38,00%		
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	26,43%	
		Angry	11,00%	7,00%	12,00%	13,00%	17,00%	5,00%	14,00%		
		Disgust	21,00%	45,00%	12,00%	13,00%	12,00%	5,00%	13,00%		
		Fear	16,00%	10,00%	17,00%	11,00%	14,00%	16,00%	9,00%		
		Happy	7,00%	9,00%	15,00%	29,00%	8,00%	6,00%	10,00%		
		Sad	14,00%	16,00%	13,00%	14,00%	12,00%	8,00%	16,00%		
		Surprise	11,00%	5,00%	16,00%	9,00%	18,00%	45,00%	12,00%		
		Neutral	20,00%	8,00%	15,00%	11,00%	19,00%	15,00%	26,00%		
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	24,46% (originals 27,43%)	
		Angry	12,90%	11,65%	9,94%	14,86%	12,60%	6,83%	12,73%		
		Disgust	8,08%	23,62%	9,08%	4,32%	7,25%	3,54%	10,02%		
		Fear	15,41%	12,83%	16,54%	12,10%	15,57%	12,38%	9,48%		
		Happy	21,41%	20,37%	23,90%	36,83%	17,51%	14,17%	26,24%		
		Sad	9,10%	6,19%	7,49%	9,52%	12,32%	6,16%	6,75%		
		Surprise	11,71%	9,22%	14,32%	5,71%	12,92%	42,17%	7,97%		
		Neutral	21,38%	16,13%	18,73%	16,67%	21,83%	14,75%	26,83%		
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	30,16% (originals 28%)	
		Angry	17,79%	11,73%	10,10%	9,90%	12,33%	6,48%	11,87%		
		Disgust	15,30%	43,60%	10,94%	11,49%	11,98%	6,40%	8,35%		
		Fear	15,52%	6,73%	22,24%	10,98%	13,30%	19,63%	10,00%		
		Happy	8,41%	11,48%	14,25%	33,81%	10,02%	6,37%	12,06%		
		Sad	12,63%	11,95%	12,95%	14,16%	18,52%	6,02%	15,86%		
		Surprise	11,84%	8,02%	17,52%	9,73%	13,94%	43,30%	10,02%		
		Neutral	18,49%	6,49%	12,00%	9,92%	19,90%	11,81%	31,84%		

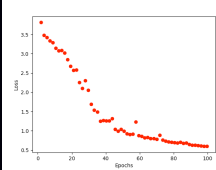
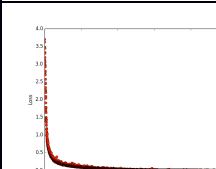
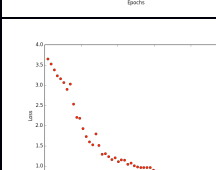
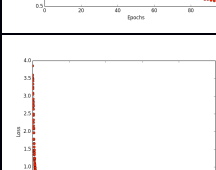
5.6.7. EXPERIMENT 7

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
			Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral		
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	26,86%	
		Angry	19,00%	10,00%	17,00%	13,00%	20,00%	16,00%	21,00%		
		Disgust	10,00%	41,00%	5,00%	8,00%	13,00%	4,00%	9,00%		
		Fear	10,00%	7,00%	12,00%	12,00%	14,00%	14,00%	14,00%		
		Happy	16,00%	11,00%	11,00%	33,00%	12,00%	14,00%	12,00%		
		Sad	30,00%	15,00%	30,00%	14,00%	25,00%	10,00%	14,00%		
		Surprise	2,00%	4,00%	9,00%	7,00%	7,00%	32,00%	4,00%		
Neutral	13,00%	12,00%	16,00%	13,00%	9,00%	10,00%	26,00%				
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	30,42%	
		Angry	24,00%	10,00%	18,00%	17,00%	19,00%	8,00%	16,00%		
		Disgust	22,00%	44,00%	12,00%	15,00%	18,00%	4,00%	11,00%		
		Fear	5,00%	7,00%	11,00%	6,00%	10,00%	5,00%	8,00%		
		Happy	16,00%	10,00%	6,00%	31,00%	10,00%	6,00%	9,00%		
		Sad	15,00%	14,00%	20,00%	8,00%	21,00%	10,00%	14,00%		
		Surprise	7,00%	6,00%	16,00%	8,00%	8,00%	51,00%	11,00%		
Neutral	11,00%	9,00%	17,00%	15,00%	14,00%	16,00%	31,00%				
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	27,95% (originals 30,42%)	
		Angry	17,60%	12,60%	11,62%	11,86%	16,10%	8,81%	14,44%		
		Disgust	8,56%	23,46%	8,54%	7,06%	7,62%	2,94%	4,68%		
		Fear	9,73%	10,25%	15,52%	12,24%	13,00%	8,86%	12,78%		
		Happy	13,02%	16,02%	12,60%	28,11%	7,48%	8,10%	8,33%		
		Sad	25,97%	18,54%	28,05%	22,40%	34,02%	13,57%	18,62%		
		Surprise	10,11%	7,95%	11,13%	9,62%	8,86%	45,52%	9,75%		
Neutral	15,02%	11,17%	12,54%	8,71%	12,94%	12,21%	31,40%				
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	33,19% (originals 31,57%)	
		Angry	21,27%	9,30%	13,32%	9,98%	22,95%	7,76%	12,60%		
		Disgust	16,89%	50,89%	12,38%	13,38%	14,24%	5,29%	13,03%		
		Fear	15,68%	5,95%	17,11%	7,43%	13,17%	8,92%	10,11%		
		Happy	11,13%	9,67%	8,63%	40,24%	11,75%	8,75%	10,17%		
		Sad	18,52%	11,43%	21,32%	9,59%	20,30%	8,94%	12,48%		
		Surprise	8,60%	3,63%	14,97%	6,73%	7,46%	49,68%	8,79%		
Neutral	7,90%	9,13%	12,27%	12,65%	10,13%	10,67%	32,81%				

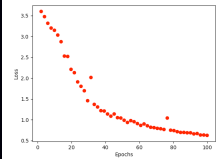
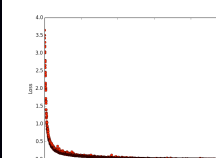
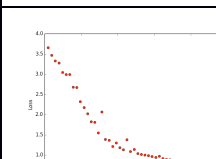
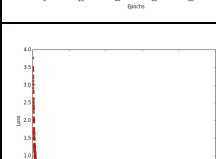
5.6.8. EXPERIMENT 8

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
			Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral		
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	31,29%	
		Angry	17,00%	7,00%	12,00%	9,00%	15,00%	11,00%	19,00%		
		Disgust	19,00%	63,00%	17,00%	10,00%	16,00%	11,00%	14,00%		
		Fear	7,00%	6,00%	21,00%	13,00%	8,00%	10,00%	8,00%		
		Happy	13,00%	8,00%	14,00%	25,00%	9,00%	9,00%	8,00%		
		Sad	17,00%	9,00%	9,00%	9,00%	19,00%	6,00%	7,00%		
		Surprise	8,00%	0,00%	9,00%	12,00%	10,00%	36,00%	6,00%		
Neutral	19,00%	7,00%	18,00%	22,00%	23,00%	17,00%	38,00%				
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	34,43%	
		Angry	15,00%	6,00%	10,00%	3,00%	15,00%	6,00%	12,00%		
		Disgust	20,00%	64,00%	12,00%	8,00%	10,00%	3,00%	16,00%		
		Fear	12,00%	7,00%	17,00%	9,00%	10,00%	12,00%	11,00%		
		Happy	13,00%	3,00%	19,00%	43,00%	12,00%	11,00%	11,00%		
		Sad	11,00%	9,00%	12,00%	11,00%	22,00%	8,00%	12,00%		
		Surprise	17,00%	5,00%	17,00%	8,00%	12,00%	50,00%	8,00%		
Neutral	12,00%	6,00%	13,00%	18,00%	19,00%	10,00%	30,00%				
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	26,29% (originals 31,14%)	
		Angry	22,06%	15,17%	11,11%	10,24%	15,54%	6,25%	20,65%		
		Disgust	9,83%	31,16%	8,11%	3,98%	9,14%	7,84%	5,60%		
		Fear	11,56%	8,08%	14,70%	15,29%	11,95%	13,29%	11,56%		
		Happy	15,78%	12,32%	17,35%	33,38%	12,97%	13,17%	14,67%		
		Sad	16,56%	16,10%	17,16%	14,75%	23,21%	6,52%	16,32%		
		Surprise	12,70%	5,73%	18,17%	10,48%	10,87%	38,67%	10,32%		
Neutral	11,52%	11,44%	13,40%	11,89%	16,32%	14,25%	20,89%				
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	35,18% (originals 35,81%)	
		Angry	17,71%	8,11%	10,14%	2,89%	11,25%	2,83%	14,73%		
		Disgust	19,06%	67,54%	10,57%	12,94%	13,30%	6,92%	14,78%		
		Fear	12,71%	4,11%	18,95%	13,02%	9,46%	11,79%	7,08%		
		Happy	14,46%	4,43%	12,97%	33,30%	10,13%	9,81%	9,89%		
		Sad	14,65%	5,43%	17,87%	12,71%	28,33%	7,14%	15,86%		
		Surprise	10,90%	4,10%	19,65%	8,59%	8,21%	50,94%	8,17%		
Neutral	10,49%	6,29%	9,84%	16,56%	19,32%	10,57%	29,49%				

5.6.9. EXPERIMENT 9

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
			Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral		
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	29,29%	
		Angry	22,00%	13,00%	14,00%	15,00%	12,00%	11,00%	14,00%		
		Disgust	10,00%	36,00%	11,00%	14,00%	10,00%	2,00%	10,00%		
		Fear	7,00%	5,00%	12,00%	9,00%	10,00%	12,00%	7,00%		
		Happy	11,00%	9,00%	16,00%	25,00%	11,00%	7,00%	10,00%		
		Sad	23,00%	20,00%	14,00%	11,00%	29,00%	12,00%	16,00%		
		Surprise	9,00%	2,00%	13,00%	9,00%	10,00%	44,00%	6,00%		
		Neutral	18,00%	15,00%	20,00%	17,00%	18,00%	12,00%	37,00%		
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	29,00%	
		Angry	26,00%	19,00%	20,00%	18,00%	19,00%	7,00%	11,00%		
		Disgust	13,00%	40,00%	10,00%	9,00%	13,00%	6,00%	15,00%		
		Fear	9,00%	12,00%	18,00%	16,00%	22,00%	18,00%	15,00%		
		Happy	9,00%	5,00%	8,00%	23,00%	8,00%	4,00%	7,00%		
		Sad	17,00%	9,00%	15,00%	11,00%	18,00%	4,00%	14,00%		
		Surprise	11,00%	6,00%	18,00%	13,00%	8,00%	55,00%	15,00%		
		Neutral	15,00%	9,00%	11,00%	10,00%	12,00%	6,00%	23,00%		
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	24,41% (originals 28,00%)	
		Angry	18,00%	16,44%	12,37%	15,52%	14,67%	7,21%	10,76%		
		Disgust	9,89%	25,90%	8,73%	7,05%	9,05%	5,03%	8,22%		
		Fear	16,87%	13,57%	15,33%	15,63%	20,68%	17,56%	20,22%		
		Happy	15,17%	14,00%	16,97%	27,68%	15,29%	11,48%	14,78%		
		Sad	9,83%	12,98%	14,25%	10,52%	18,24%	6,22%	9,89%		
		Surprise	14,65%	8,44%	16,95%	9,30%	9,17%	42,86%	13,29%		
		Neutral	15,59%	8,65%	15,40%	14,29%	12,90%	9,65%	22,84%		
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	30,61% (originals 30,71%)	
		Angry	19,38%	11,90%	12,62%	10,81%	17,90%	8,19%	6,84%		
		Disgust	16,43%	46,32%	14,30%	15,11%	15,79%	4,17%	13,27%		
		Fear	11,16%	12,86%	15,73%	14,54%	16,37%	10,21%	12,13%		
		Happy	9,43%	5,67%	11,05%	23,90%	7,86%	7,41%	13,30%		
		Sad	17,94%	11,33%	13,89%	12,08%	22,75%	5,02%	16,41%		
		Surprise	8,40%	2,62%	16,89%	6,94%	5,84%	57,10%	8,89%		
		Neutral	17,27%	9,30%	15,52%	16,62%	13,49%	7,90%	29,16%		

5.6.10. EXPERIMENT 10

#	Descripció	Matriu de confusió								Accuracy	Loss/epochs
			Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral		
1	entrenament i test amb imatges originals (700 imatges)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	24,84%	
		Angry	18,00%	11,11%	18,00%	22,00%	21,00%	13,00%	24,00%		
		Disgust	5,00%	8,33%	3,00%	2,00%	5,00%	0,00%	2,00%		
		Fear	20,00%	27,78%	22,00%	15,00%	21,00%	26,00%	11,00%		
		Happy	9,00%	2,78%	8,00%	30,00%	8,00%	5,00%	6,00%		
		Sad	14,00%	16,67%	12,00%	10,00%	16,00%	7,00%	20,00%		
		Surprise	10,00%	2,78%	19,00%	6,00%	10,00%	42,00%	10,00%		
Neutral	24,00%	30,56%	18,00%	15,00%	19,00%	7,00%	27,00%				
2	entrenament amb conjunt ampliat exp 1 (44K) i test amb imatges originals de l'exp 1 (700)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	28,77%	
		Angry	14,00%	13,89%	8,00%	7,00%	10,00%	6,00%	9,00%		
		Disgust	7,00%	13,89%	5,00%	2,00%	4,00%	2,00%	4,00%		
		Fear	9,00%	11,11%	16,00%	9,00%	17,00%	17,00%	12,00%		
		Happy	18,00%	13,89%	9,00%	39,00%	19,00%	7,00%	11,00%		
		Sad	18,00%	11,11%	16,00%	12,00%	22,00%	6,00%	21,00%		
		Surprise	8,00%	8,33%	22,00%	14,00%	10,00%	52,00%	8,00%		
Neutral	26,00%	27,78%	24,00%	17,00%	18,00%	10,00%	35,00%				
3	entrenament imatges originals (700 imatges) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	22,79% (originals 25,00%)	
		Angry	15,16%	20,37%	15,95%	9,78%	14,73%	15,29%	11,10%		
		Disgust	2,25%	5,60%	1,52%	1,94%	3,51%	0,57%	0,98%		
		Fear	17,65%	17,20%	17,08%	19,32%	14,70%	18,94%	17,35%		
		Happy	25,78%	15,17%	18,03%	36,63%	21,32%	12,08%	18,44%		
		Sad	15,27%	13,32%	14,21%	11,06%	19,57%	11,86%	23,19%		
		Surprise	7,78%	7,85%	17,14%	7,30%	9,95%	36,48%	10,94%		
Neutral	16,11%	20,50%	16,06%	13,97%	16,22%	4,79%	18,00%				
4	E entrenament amb conjunt ampliat exp 1 (44K) i test amb conjunt ampliat test exp 1 (44K)		Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral	28,64% (originals 27,20%)	
		Angry	19,65%	23,99%	13,81%	9,51%	15,98%	8,65%	14,27%		
		Disgust	4,81%	15,74%	4,43%	2,37%	5,76%	0,52%	2,78%		
		Fear	11,97%	1,90%	17,89%	10,46%	16,19%	17,40%	14,27%		
		Happy	17,35%	8,20%	14,49%	42,52%	19,17%	8,68%	11,35%		
		Sad	19,75%	11,86%	12,13%	13,14%	16,76%	4,81%	22,02%		
		Surprise	9,76%	12,17%	22,59%	9,54%	8,40%	52,46%	8,10%		
Neutral	16,71%	26,15%	14,67%	12,46%	17,73%	7,48%	27,22%				

En la següent taula agrupem tots els resultats obtinguts en els 10 experiments anteriors, en calculem la mitjana, la desviació estadística i també l'interval de confiança amb una probabilitat del 95%.

	Subexperiment 1	Subexperiment 2	Subexperiment 3		Subexperiment 4	
	100 train - 100 test	44k train - 100 test	100 train - 44k test	700 imatges originals en espic	44k train - 44k test	700 imatges originals en expedí
exp 1	28,86	31,71	24,08	28,85	30,79	31
exp 2	30,64	29,62	26,7	30,63	30,47	29
exp 3	31,33	34,91	27,36	32,18	34,1	34,47
exp 4	29,57	31	27,17	31,71	32,02	32,71
exp 5	23,99	27,67	26,02	30,18	28,55	27,67
exp 6	27,14	26,43	24,46	27,42	30,16	28
exp 7	26,86	30,43	27,95	30,43	33,18	31,57
exp 8	31,29	34,43	26,29	31,14	35,18	35,85
exp 9	29,29	29	24,41	28	30,62	30,71
exp 10	24,84	28,77	22,79	25	28,64	27,2
Mitjana	28,381	30,397	25,723	29,554	31,371	30,818
desv	2,457732492	2,590309055	1,603696044	2,111398589	2,103470703	2,776583512
$=0,95 * desv / (10)^{0,5}$	0,738343093	0,778171263	0,481776556	0,634298716	0,631917049	0,834131161
Límit inf. Interv. Confiança	27,64265691	29,61882874	25,24122344	28,91970128	30,73908295	29,98386884
Límit sup. Interv. Confiança	29,11934309	31,17517126	26,20477656	30,18829872	32,00291705	31,65213116
INTERVAL CONFIANÇA 95%	[27,64 , 29,12]	[29,62 , 31,18]	[25,24 , 26,20]	[28,92 , 30,19]	[30,74 , 32,00]	[29,98 , 31,65]

Resumint tenim:

Tipus Experiment	Descripció Experiment	INTERVAL CONFIANÇA 95%
Subexperiment 1	700 train - 700 test	[27,64 , 29,12]
Subexperiment 2	44k train - 700 test	[29,62 , 31,18]
Subexperiment 3	700 train - 44k test	[25,24 , 26,20]
	700 imatges originals de test (per comparar amb Subexp. 1)	[28,92 , 30,19]
Subexperiment 4	44k train - 44k test	[30,74 , 32,00]
	700 imatges originals de test (per comparar amb Subexp. 2)	[29,98 , 31,65]

6. Conclusions

Un cop analitzats i vistos els resultats en l'apartat anterior, podem destacar els següents punts:

- Tal com es veu en les matrius de confusió, no existeix correlació en el percentatge de classificació de les diferents classes entre una CNN entrenada amb conjunt ampliat i una CNN entrenada amb un conjunt original. És a dir, el percentatge d'encert de la classe angry varia del 14% al 25%.
- El fet que hi hagi una diferència entre el número d'imatges d'una classe i l'altre fa que el resultat també variï, veure els casos de la classe disgust en els experiments 5 i 10 on hi havia menys imatges d'aquest tipus. Això indica que si hi ha una informació molt predominant a l'entrenament, la CNN es pot esbiaixar.
- Notar que els resultats obtinguts en les CNNs entrenades amb conjunts ampliats sempre estan 2 punts per sobre de les CNNs entrenades amb els conjunts originals. La millora és estadísticament significativa.
- Els intervals de confiança per experiments similars no es solapen.

La CONCLUSIÓ final és que la hipòtesi queda validada, les CNN entrenades amb conjunts ampliats – data augmentation – classifiquen millor que les xarxes entrenades amb menys informació.

7. Bibliografia

7.1. Referències

[1] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion." *Journal of personality and social psychology*, vol. 17, no. 2, p. 124, 1971.

[2] — —, "Facial action coding system: A technique for the measurement of facial movement. palo alto," *CA: Consulting Psychologists Press. Ellsworth, PC, & Smith, CA (1988). From appraisal to emotion: Differences among unpleasant feelings. Motivation and Emotion*, vol. 12, pp. 271–302, 1978.

[3] "Facial Expression Recognition 2013 database," <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>, (última consulta a la web 28/05/2017)

[4] <https://www.tensorflow.org> (última consulta a la web 28/05/2017)

7.2. Bibliografia

<http://www.opengl-tutorial.org/beginners-tutorials/tutorial-5-a-textured-cube/>

<http://www.cl.cam.ac.uk/research/dtg/>

<http://stackoverflow.com/questions>

<https://www.pythonmania.net/es/>

https://www.tensorflow.org/api_docs/python/

JOGL, java input output Package. Tutorialspoint. www.tutorialspoint.com

Jorge García, “*Curso de introducción a OpenGL (v1.0)*”, 2003

Oscar García, Alex Guevara, “*Introducción a la Programación Gráfica con OpenGL*”, Escola Tècnica Superior d’Enginyeria Electrònica i Informàtica La Salle, Enero 2004

Jordi Torres, “Hello world en TensorFlow, para iniciarse en la programación del Deep Learning”, Watch this spaces collection, BSC & UPC Barcelona Tech, Barcelona, 2016. Ed. Undertitle, ISBN 978-1-326-56702-6

8. Annex A: Codi Generació imatges (JAVA)

```
// fitxer Main.java
package openGL101;

import java.awt.Container;
import java.awt.Frame;
import java.awt.Graphics2D;
import java.awt.Transparency;
import java.awt.geom.AffineTransform;
import java.awt.image.BufferedImage;
import java.awt.image.ColorModel;
import java.awt.image.ComponentColorModel;
import java.awt.image.DataBuffer;
import java.awt.image.DataBufferByte;
import java.awt.image.Raster;
import java.awt.image.WritableRaster;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.FloatBuffer;

import javax.imageio.ImageIO;
import javax.swing.JFrame;

import com.jogamp.common.nioBuffers;
import com.jogamp.opengl.GL;
import com.jogamp.opengl.GL2;
import com.jogamp.opengl.GLAutoDrawable;
import com.jogamp.opengl.GLCapabilities;
import com.jogamp.opengl.GLEventListener;
import com.jogamp.opengl.GLProfile;
import com.jogamp.opengl.awt.GLCanvas;
import com.jogamp.opengl.util.FPSAnimator;
import com.jogamp.opengl.util.texture.TextureData.ColorSpace;

public class MAin {

    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub

        String path, path2, ima_path, label_path;
        String [] folderContent;
        File folder;
        int i;
        int num_0 = 0;
        int num_1 = 0;
        int num_2 = 0;
        int num_3 = 0;
        int num_4 = 0;
        int num_5 = 0;
        int num_6 = 0;
        int num_7 = 0;
        int num_alt = 0;
        int gnum_0 = 0;
        int gnum_1 = 0;
        int gnum_2 = 0;
        int gnum_3 = 0;
        int gnum_4 = 0;
        int gnum_5 = 0;
        int gnum_6 = 0;
        int gnum_7 = 0;
        int gnum_alt = 0;
        int tot = 0;
        int gtot = 0;
        long n_g = 0L;
        int limit = 1000;
    }
}
```

```

int inici = 0;

long time_start, time_end;

if (args[0] == null)
    path = = "/Users/jordisors/Documents/workspace/src/img/xxxx.jpg";
else
    path = args[0];

String wrk_path[] =
{"Users/jordisors/Documents/workspace/py/xxtrain_orig/",
"/Users/jordisors/Documents/workspace/py/xxtest_orig/",
"/Users/jordisors/Documents/workspace/py/xxtrain_tract/",
"/Users/jordisors/Documents/workspace/py/xxtest_tract/"};

boolean ampliat[] = {false, false, false, false};
boolean copia[] = {true, true, false, false};
int incr[] = {1, -1, 1, -1};

for (int xx = 0; xx<wrk_path.length; xx++ )
{
    time_start = System.currentTimeMillis();
    ima_path = wrk_path [xx] ;
    label_path = wrk_path[xx];

    folder = new File (path);
    num_0 = 0;
    num_1 = 0;
    num_2 = 0;
    num_3 = 0;
    num_4 = 0;
    num_5 = 0;
    num_6 = 0;
    num_7 = 0;
    num_alt = 0;
    gnum_0 = 0;
    gnum_1 = 0;
    gnum_2 = 0;
    gnum_3 = 0;
    gnum_4 = 0;
    gnum_5 = 0;
    gnum_6 = 0;
    gnum_7 = 0;
    gnum_alt = 0;
    tot = 0;
    gtot = 0;
    n_g = 0L;
    path = folder.getAbsolutePath() + "/";
    folderContent = folder.list();
    BufferedWriter out =
    new BufferedWriter (new FileWriter(wrk_path[xx]+"gravats.txt"));
    if (xx % 2 == 0){
        inici = 1;
    }else {
        inici = folderContent.length-1;
    }
    for (i = inici; (i< folderContent.length) && (i>0); i+=incr[xx]){
        if ((folderContent[i].endsWith(".jpg")) ||
(folderContent[i].endsWith(".JPG"))){
            if ((folderContent[i].charAt(6) == '.') &&
(folderContent[i].charAt(0) == '0') &&
(folderContent[i].charAt(1) == '0') &&
(folderContent[i].charAt(2) == '0') &&
(folderContent[i].charAt(3) == '0')){
                File fileToGenerate = new File (path+folderContent[i]);
                if (fileToGenerate.exists()) {
                    String str = "x";
                    generacio_imatges genIm = new generacio_imatges
(fileToGenerate.getAbsolutePath(), ima_path,
label_path, str,ampliat[xx], copia[xx]);
                    genIm.run();
                    genIm.stop();
                } else
                    System.out.println("fitxer INEXISTENT :"+
+ folderContent[i]);
            }
        }
    }
}

```

```

}
}else {
if ((folderContent[i].endsWith(".txt")) ||
(folderContent[i].endsWith(".TXT"))){
BufferedReader in = new BufferedReader(
new FileReader(path+folderContent[i]));
String str = new String();
char c = ' ';
if ((str = in.readLine()) != null){
System.out.println(folderContent[i]+str +
" fitxer número: " + i);
in.close();
in = null;
c = str.charAt(0);
File fileToGenerate = new File (path2 + "/"
+folderContent[i].substring(0, 6) + ".jpg");
if (fileToGenerate.exists()) {
switch (c){
case '0': num_0++; tot = num_0;
break;
case '1': num_1++; tot = num_1;
break;
case '2': num_2++; tot = num_2;
break;
case '3': num_3++; tot = num_3;
break;
case '4': num_4++; tot = num_4;
break;
case '5': num_5++; tot = num_5;
break;
case '6': num_6++; tot = num_6;
break;
case '7': num_7++; tot = num_7;
break;
default: num_alt++;tot = num_alt;
break;
}
if (tot <= limit){
n_g += 63;
switch (c){
case '0': gnum_0++; break;
case '1': gnum_1++; break;
case '2': gnum_2++; break;
case '3': gnum_3++; break;
case '4': gnum_4++; break;
case '5': gnum_5++; break;
case '6': gnum_6++; break;
case '7': gnum_7++; break;
default: gnum_alt++; break;
}
gtot++;
int ioff = (((tot-1) / 100) + 1) *
out.write("generat el conjunt : " +
+
+
+folderContent[i].substring(0,6)
+ ".jpg" + " fitxer número: " + i
+ " tipus emocio: " + str.charAt(0)
+ " num fit tipus: " + tot
+ " num gravats: " + n_g + "\n");
generacio_imatges genIm = new
(fileToGenerate.getAbsolutePath(),
ima_path+ioff+"/ima/",
label_path+ioff+"/labels/",
str,ampliat[xx], copia[xx]);
genIm.run();
genIm.stop();
} else {
out.write ("!!!!!!! fitxer inexistent : "
+
+
+folderContent[i].substring(0,6) +
".jpg" + " fitxer número: " + i +

```



```

        " tipus emocio: " + str.charAt(0) +
        " num fit tipus: " + tot +
        " num gravats: " + n_g + "\n");
    }
    }
} else
    System.out.println("fitxer INEXISTENT : " +
        folderContent[i]);
}
}
time_end = System.currentTimeMillis();
out.write("the task has taken "+
    ( time_end - time_start ) + " milliseconds"+ "\n");
out.write("el numero de fitxer de tipus 0 es "+ num_0 +
    " num gravats: " + gnum_0+ "\n");
out.write("el numero de fitxer de tipus 1 es "
    + num_1 + " num gravats: " + gnum_1 + "\n");
out.write("el numero de fitxer de tipus 2 es "+ num_2
    + " num gravats: " + gnum_2 + "\n");
out.write("el numero de fitxer de tipus 3 es "+ num_3
    + " num gravats: " + gnum_3 + "\n");
out.write("el numero de fitxer de tipus 4 es "+ num_4
    + " num gravats: " + gnum_4 + "\n");
out.write("el numero de fitxer de tipus 5 es "+ num_5
    + " num gravats: " + gnum_5 + "\n");
out.write("el numero de fitxer de tipus 6 es "+ num_6
    + " num gravats: " + gnum_6 + "\n");
out.write("el numero de fitxer de tipus 7 es "+ num_7
    + " num gravats: " + gnum_7 + "\n");
out.write("el numero de fitxer de tipus alt es "
    + num_alt + " num gravats: " + gnum_alt + "\n");
out.write("el numero total de fitxers gravats és "+ gtot + "\n");
out.close();
out = null;

} else{
    generacio_imatges genIm = new generacio_imatges
        (folder.getAbsolutePath(),"ima/",
        "labels","x",ampliat[xx], copia[xx]);
    genIm.run();
    genIm.stop();
}
} else {
    System.out.println("fitxer inexistent " + path);
}
}
}

// fitxer ggeneracio_imatges.java

package openGL101;

import javax.swing.JFrame;

import com.jogamp.opengl.GLCapabilities;
import com.jogamp.opengl.GLProfile;
import com.jogamp.opengl.awt.GLCanvas;
import com.jogamp.opengl.util.FPSAnimator;

public class generacio_imatges {

    private String FileName;
    private String image_path;
    private String label_path;
    private String emotion_type;
    private boolean ampliat = false;
    private boolean copia = false;
    private Gir gir;
    private JFrame frame = new JFrame("Rotació d'imatges" + FileName);
    private FPSAnimator animator;

    public generacio_imatges (String InputFileName, String ima_path, String lbl_path,
        String em_type, boolean ampli, boolean copy) {

        FileName = InputFileName;
        image_path = ima_path;
        label_path = lbl_path;

```

```

        emotion_type = em_type;
        ampliati = ampli;
        copia = copy;
    }

    public void run(){
        final int width = 96;
        final int height = 118;
        final GLProfile profile = GLProfile.get(GLProfile.GL2);
        GLCapabilities capabilities = new GLCapabilities(profile);

        gir = new Gir(fileName, image_path, label_path, emotion_type,
            ampliati, copia);

        final GLCanvas canvas = new GLCanvas(capabilities);
        canvas.addGLEventListener(gir);
        frame.getContentPane().add(canvas);
        frame.setSize(width, height);
        frame.setLocationRelativeTo(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
        frame.setResizable(false);
        canvas.requestFocusInWindow();

        animator = new FPSAnimator (canvas, 20, true);
        do {
            animator.start();
            animator.stop();
        }while (!gir.isFinished());
    }

    public void stop() {
        int i=0;
        while (!gir.isFinished()){
            i++;
        }
        animator.stop();
        frame.dispose();
    }
}

// fiter Gir.java
package openGL101;

import java.awt.image.BufferedImage;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.nio.FloatBuffer;

import com.jogamp.common.nioBuffers;
import com.jogamp.opengl.GL;
import com.jogamp.opengl.GL2;
import com.jogamp.opengl.GLAutoDrawable;
import com.jogamp.opengl.GLCapabilities;
import com.jogamp.opengl.GLEventListener;
import com.jogamp.opengl.GLProfile;
import com.jogamp.opengl.awt.GLCanvas;
import com.jogamp.opengl.util.FPSAnimator;
import com.jogamp.opengl.util.texture.Texture;
import com.jogamp.opengl.util.texture.TextureIO;
import com.jogamp.opengl.glu.GLU;

import javax.imageio.ImageIO;
import javax.swing.JFrame;
import java.lang.Character;

public class Gir implements GLEventListener {
    private static final long serialVersionUID = 1L;
    private static String fileName;
    private String image_path;
    private String label_path;
    private String emotion_type;
    private String categories[] = {"Angry", "Disgust", "Fear", "Happy",
        "Sad", "Surprise", "Neutral"};

```

```

private float punt = 0.0f;
private int rtrX = 0;
private int rtrY = 0;
private int texture;
private static boolean finished = false;
private GLU glu = new GLU();
private boolean ampliati = false;
private boolean copia = false;
private int max_radX = 40;
private int min_radX = 320;
private int inc_radX = 10;
private int max_radY = 30;
private int min_radY = 330;
private int inc_radY = 10;
private int inc_grausX = inc_radX;
private int inc_grausY = inc_radY;

public Gir (String FN, String ima_path, String lbl_path, String em_type,
           boolean ampli, boolean copia) {
    FileName = FN;
    image_path = ima_path;
    label_path = lbl_path;
    emotion_type = em_type;
    ampliati = ampli;
    copia = copia;

    if (copia == false) {
        max_radX = 40;
        min_radX = 320;
        inc_radX = 10;
        max_radY = 30;
        min_radY = 330;
        inc_radY = 10;
    } else {
        max_radX = 5;
        min_radX = 355;
        inc_radX = 10;
        max_radY = 5;
        min_radY = 355;
        inc_radY = 10;
    }
}

public boolean isFinished() {
    return this.finished;
}

private void writeBufferToFile(GL2 gl, GLAutoDrawable drawable) {
    int width = drawable.getSurfaceWidth()/2;
    int height = drawable.getSurfaceHeight()/2;
    if ((rtrX <= max_radX) || (rtrX >= min_radX)) {
        if ((rtrY <= max_radY) || (rtrY >= min_radY)) {
            ByteBuffer pixelsRGB =
                Buffers.newDirectByteBuffer(width * height * 3);
            this.finished = false;
            gl.glReadBuffer(GL.GL_BACK);
            gl.glPixelStorei(GL.GL_PACK_ALIGNMENT, 1);
            gl.glReadPixels(24, 24, width, height,
                           GL.GL_RGB, GL.GL_UNSIGNED_BYTE, pixelsRGB);
            int[] pixels = new int[width * height];
            int firstByte = width * height * 3;
            int sourceIndex;
            int targetIndex = 0;
            int rowBytesNumber = width * 3;
            for (int row = 0; row < height; row++) {
                firstByte -= rowBytesNumber;
                sourceIndex = firstByte;
                for (int col = 0; col < width; col++) {
                    int iR = pixelsRGB.get(sourceIndex++);
                    int iG = pixelsRGB.get(sourceIndex++);
                    int iB = pixelsRGB.get(sourceIndex++);
                    pixels[targetIndex++] = 0xFF000000
                        | ((iR & 0x000000FF) << 16)
                        | ((iG & 0x000000FF) << 8)
                        | (iB & 0x000000FF);
                }
            }
        }
    }
}

```

```

        BufferedImage bufferedImage = new BufferedImage
            (width, height, BufferedImage.TYPE_INT_RGB);
        bufferedImage.setRGB(0, 0, width, height, pixels, 0,
width);

        try {
            int i = 0;
            char c = '0';
            c= emotion_type.charAt(0);
            i = java.lang.Character.getNumericValue(c);
            String fn, fntxt;
            if (copia){
                fn = new String(image_path +categories[i]
+ "/" +
                FileName.substring(FileName.length()-10,
                    FileName.length()));
                fntxt = new String(label_path+
                    FileName.substring(FileName.length()
-10,
                    FileName.length()-4)+".txt");
            }else{
                fn = new String(image_path +categories[i]
+ "/" +
                FileName.substring(FileName.length()
-10,
                    FileName.length()-4)+"_eixX_" + rtrX
+
                    "_eixY_" + rtrY + ".jpg");
                fntxt = new String (label_path+
                    FileName.substring(FileName.length()
-10,
                    FileName.length()-4)+"_eixX_" + rtrX
+
                    "_eixY_" + rtrY + ".txt");
            }
            ImageIO.write(bufferedImage, "JPG", new File(fn));
            BufferedWritter out = new
                BufferedWritter (new FileWriter(fntxt));
            out.write(emotion_type);
            out.close();
            out = null;
        } catch (Exception e) {
            e.printStackTrace();
        }
        rtrY += inc_grausY;           //assigning the angle
    } else {
        if (inc_grausY > 0){
            rtrY = 360 - inc_radY;
            inc_grausY = -inc_radY;
        }else {
            rtrY = 0;
            inc_grausY = inc_radY;
            rtrX += inc_grausX; //assigning the angle
        }
    }
} else {
    if (inc_grausX > 0) {
        rtrX = 360 - inc_radX;
        inc_grausX = -inc_radX;
        rtrY = 0;
        inc_grausY = inc_radY;
    }
    else {
        System.out.println("acabem el fixer");
        this.finished = true;
    }
}
}

@Override

public void display(GLAutoDrawable drawable) {
    final GL2 gl = drawable.getGL().getGL2();
    gl.glClear(GL2.GL_COLOR_BUFFER_BIT | GL2.GL_DEPTH_BUFFER_BIT);

```

```

// clear the screen and the depth buffer

gl.glLoadIdentity();
gl.glBindTexture(GL2.GL_TEXTURE_2D, texture);
gl.glRotatef((float)rtrX, 1.0f, 0.0f, 0.0f);
gl.glRotatef((float)rtrY, 0.0f, 1.0f, 0.0f);
gl.glBegin(GL2.GL_QUADS);
if (ampliat == true){
    punt = 0.75f;
}else{
    punt = 0.50f;
}
gl.glTexCoord2f(1.0f, 0.0f); gl.glVertex3f(punt, -punt, 0.0f);
gl.glTexCoord2f(1.0f, 1.0f); gl.glVertex3f(punt, punt, 0.0f);
gl.glTexCoord2f(0.0f, 1.0f); gl.glVertex3f(-punt, punt, 0.0f);
gl.glTexCoord2f(0.0f, 0.0f); gl.glVertex3f(-punt, -punt, 0.0f);
gl.glEnd();

gl.glFlush();
writeBufferToFile(gl, drawable);
}

@Override
public void dispose(GLAutoDrawable drawable) {
    this.finished = true;
    this.FileName = "";
    this.rtrX = 0;
    this.rtrY = 0;
    this.inc_grausX = inc_radX;
    this.inc_grausY = inc_radY;
    final GL2 gl = drawable.getGL().getGL2();
    Texture t = new Texture(texture);
    t.destroy(gl);
}

@Override
public void init(GLAutoDrawable drawable) {
    final GL2 gl = drawable.getGL().getGL2();
    gl.glShadeModel(GL2.GL_SMOOTH);
    gl.glClearColor(0.392f, 0.584f, 0.929f, 1.0f);
    gl.glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
    gl.glClearDepth(1.0f);
    gl.glEnable(GL2.GL_DEPTH_TEST);
    gl.glDepthFunc(GL2.GL_LEQUAL);
    gl.glHint(GL2.GL_PERSPECTIVE_CORRECTION_HINT, GL2.GL_NICEST);

    gl.glEnable(GL2.GL_TEXTURE_2D);

    try{
        File im = new File(FileName);
        Texture t = TextureIO.newTexture(im, true);
        texture= t.getTextureObject(gl);
    }catch(IOException e){
        e.printStackTrace();
    }
}

@Override
public void reshape(GLAutoDrawable drawable, int x, int y, int width,
    int height) {

    final GL2 gl = drawable.getGL().getGL2();

    if (height <=0)
        height = 1;

    final float h = (float) width / (float) height;

    gl.glViewport(0, 0, width, height);

    gl.glMatrixMode(GL2.GL_PROJECTION);
    gl.glLoadIdentity();
    gl.glMatrixMode(GL2.GL_MODELVIEW);
    gl.glLoadIdentity();
}
}

```

9. Annex B: Codi Entrenament CNN (PY)

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jul 26 16:00:05 2016
Modified on 2017, April-May by JS

@author: gerard - adapted by jordi sors
"""

from __future__ import absolute_import
from __future__ import division
from __future__ import print_function

import sys, time, glob

from datetime import datetime

import numpy as np
from PIL import Image
from six.moves import xrange # pylint: disable=redefined-builtin
import tensorflow as tf
from tensorflow import nn

IMAGE_SIZE_W = 48
IMAGE_SIZE_H = 48
NUM_CHANNELS = 1
PIXEL_DEPTH = 255
NUM_LABELS = 7
SEED = 66478 # Set to None for random seed.
VALIDATION_IMAGES = 100
BATCH_SIZE = 105 # 105 #35
NUM_EPOCHS = 100 # abans era 100 - minim
EVAL_BATCH_SIZE = 100 #80
EVAL_FREQUENCY = 10 # Number of steps between evaluations.
CHECKPOINT_DIR = './models/'
BASE_LEARNING_RATE = 0.001
NEUR = 512 #Number of neurons in the fc layer
FILT_SZ = 5 #Size of the conv filter

tf.app.flags.DEFINE_boolean("train", True, "True if running a training.")
FLAGS = tf.app.flags.FLAGS

from utility_functions import *

f = open('./output.txt','w')
f2 = open('results.csv', 'w')

def generateDatabaseSFEW():
    """Return the list of images for training and testing."""
    # Change path
    dirSFEW = '/data/Workspace/Data/SFEW_2'
    dirSFEW = '/Users/jordisors/Documents/workspace/py/'

    categories = ['Angry', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprise', 'Neutral']
    imagesTrain = []
    imagesTest = []
    for cat in categories:
        imagesTrain.append(sorted(glob.glob(dirSFEW + 'xxtrain_tract/100/ima/' + cat +
        '/*.jpg')))
        imagesTest.append(sorted(glob.glob(dirSFEW + 'xctest_orig/100/ima/' + cat +
        '/*.jpg')))

    return imagesTrain, imagesTest

def accuracy_rate(predictions, labels):
    """Return the accuracy based on dense predictions and sparse labels."""
    return np.sum(np.argmax(predictions, 1) == labels) / predictions.shape[0]

def error_rate(predictions, labels):
    """Return the error rate based on dense predictions and sparse labels."""
```

```

return 100.0 - (
    100.0 *
    np.sum(np.argmax(predictions, 1) == labels) /
    predictions.shape[0])

def print_test_labels(predictions, batch_labels_test, testimages):
    """Return the error rate based on dense predictions and sparse labels."""
    for i, img in enumerate(testimages):
        pred = predictions[i]
        lbl = batch_labels_test[i]
        f2.write( '%s; %d; %d \n' % (img, np.argmax(pred, 0), lbl))

def weight_variable(shape):
    initial = tf.truncated_normal(shape, stddev=0.1)
    return tf.Variable(initial)

def main(argv=None):

    # Read the images from disk
    imagesTrain, imagesTest = generateDatabaseSFEW()

    num_epochs = NUM_EPOCHS

    # Creates a new graph
    graph = tf.Graph()
    with graph.as_default():
        # This is where training samples and labels are fed to the graph.
        # These placeholder nodes will be fed a batch of training data at each
        # training step using the {feed_dict} argument to the Run() call below.
        train_data_node = tf.placeholder(tf.float32, shape=(BATCH_SIZE, IMAGE_SIZE_W,
IMAGE_SIZE_H, NUM_CHANNELS))
        train_labels_node = tf.placeholder(tf.int64, shape=(BATCH_SIZE,))
        eval_data = tf.placeholder(tf.float32, shape=(EVAL_BATCH_SIZE, IMAGE_SIZE_W,
IMAGE_SIZE_H, NUM_CHANNELS))

        # This placeholder have size None in order to be fed with any shape of batch for
        test
        test_data = tf.placeholder(tf.float32, shape=(None, IMAGE_SIZE_W, IMAGE_SIZE_H,
NUM_CHANNELS))

        # The variables below hold all the trainable weights. They are passed an
        # initial value which will be assigned when we call:
        # {tf.initialize_all_variables().run()}
        #conv1_weights = tf.get_variable("conv1_weights", shape=[FILT_SZ, FILT_SZ,
NUM_CHANNELS, 6],
        #    initializer=tf.contrib.layers.xavier_initializer())
        #conv1_biases = tf.Variable(tf.zeros([6], dtype=tf.float32))

        #conv2_weights = tf.get_variable("conv2_weights", shape=[FILT_SZ, FILT_SZ, 6, 12],
        #    initializer=tf.contrib.layers.xavier_initializer())
        #conv2_biases = tf.Variable(tf.constant(0.1, shape=[12], dtype=tf.float32))

        #fc1_weights = tf.get_variable("fc1_weights", shape=[12 * 12 * 12, NEUR],
        #    initializer=tf.contrib.layers.xavier_initializer())
        #fc1_biases = tf.Variable(tf.constant(0.1, shape=[NEUR], dtype=tf.float32))

        #fc2_weights = tf.get_variable("fc2_weights", shape=[NEUR, NUM_LABELS],
        #    initializer=tf.contrib.layers.xavier_initializer())
        #fc2_biases = tf.Variable(tf.constant(
        #    0.1, shape=[NUM_LABELS], dtype=tf.float32))

        conv1_weights = weight_variable(shape=[FILT_SZ, FILT_SZ, NUM_CHANNELS, 6])
        conv1_biases = tf.Variable(tf.zeros([6], dtype=tf.float32))

        conv2_weights = weight_variable(shape=[FILT_SZ, FILT_SZ, 6, 12])
        conv2_biases = tf.Variable(tf.constant(0.1, shape=[12], dtype=tf.float32))

        fc1_weights = weight_variable(shape=[12 * 12 * 12, NEUR])
        fc1_biases = tf.Variable(tf.constant(0.1, shape=[NEUR], dtype=tf.float32))

        fc2_weights = weight_variable(shape=[NEUR, NUM_LABELS])
        fc2_biases = tf.Variable(tf.constant(
            0.1, shape=[NUM_LABELS], dtype=tf.float32))

    # We will replicate the model structure for the training subgraph, as well
    # as the evaluation subgraphs, while sharing the trainable parameters.

```

```

def model(data, train=False):
    """The Model definition."""
    # 2D convolution, with 'SAME' padding (i.e. the output feature map has
    # the same size as the input). Note that {strides} is a 4D array whose
    # shape matches the data layout: [image index, y, x, depth].
    conv = tf.nn.conv2d(data,
                        conv1_weights,
                        strides=[1, 1, 1, 1],
                        padding='SAME')
    # Bias and rectified linear non-linearity.
    relu = tf.nn.relu(tf.nn.bias_add(conv, conv1_biases))
    # Max pooling. The kernel size spec {ksize} also follows the layout of
    # the data. Here we have a pooling window of 2, and a stride of 2.
    pool = tf.nn.max_pool(relu,
                          ksize=[1, 2, 2, 1],
                          strides=[1, 2, 2, 1],
                          padding='SAME')
    conv = tf.nn.conv2d(pool,
                        conv2_weights,
                        strides=[1, 1, 1, 1],
                        padding='SAME')
    relu = tf.nn.relu(tf.nn.bias_add(conv, conv2_biases))
    pool = tf.nn.max_pool(relu,
                          ksize=[1, 2, 2, 1],
                          strides=[1, 2, 2, 1],
                          padding='SAME')
    # Reshape the feature map cuboid into a 2D matrix to feed it to the
    # fully connected layers.
    pool_shape = pool.get_shape().as_list()
    reshape = tf.reshape(
        pool,
        [-1, pool_shape[1] * pool_shape[2] * pool_shape[3]])
    # Fully connected layer. Note that the '+' operation automatically
    # broadcasts the biases.
    hidden = tf.nn.relu(tf.matmul(reshape, fc1_weights) + fc1_biases)
    # Add a 50% dropout during training only. Dropout also scales
    # activations such that no rescaling is needed at evaluation time.
    if train:
        hidden = tf.nn.dropout(hidden, 0.5, seed=SEED)
    return tf.matmul(hidden, fc2_weights) + fc2_biases

# Training computation: logits + cross-entropy loss.
logits = model(train_data_node, True)
labels = train_labels_node
# loss = tf.reduce_mean(tf.nn.sparse_softmax_cross_entropy_with_logits(logits,
train_labels_node))

tx = tf.nn.sparse_softmax_cross_entropy_with_logits(labels=labels, logits=logits)

loss = tf.reduce_mean(tx)

# L2 regularization for the fully connected parameters.
regularizers = (tf.nn.l2_loss(fc1_weights) + tf.nn.l2_loss(fc1_biases) +
                tf.nn.l2_loss(fc2_weights) + tf.nn.l2_loss(fc2_biases))
# Add the regularization term to the loss.
loss += 5e-4 * regularizers

# Optimizer: set up a variable that's incremented once per batch and
# controls the learning rate decay.
batch = tf.Variable(0, dtype=tf.float32)
# Decay once per epoch, using an exponential schedule starting at 0.01.
learning_rate = tf.train.exponential_decay(BASE_LEARNING_RATE, batch * BATCH_SIZE,
                                           100000, 0.9, staircase=True)

optimizer = tf.train.AdamOptimizer(learning_rate).minimize(loss,
global_step=batch)

# Predictions for the current training minibatch.
train_prediction = tf.nn.softmax(logits)

# Predictions for the validation, which we'll compute less often.
eval_prediction = tf.nn.softmax(model(eval_data))

# Predictions for the test, which we'll compute less often.
test_prediction = tf.nn.softmax(model(test_data))

# Small utility function to evaluate a dataset by feeding batches of data to

```



```

# {eval_data} and pulling the results from {eval_predictions}.
# Saves memory and enables this to run on smaller GPUs.
def eval_in_batches(data, sess):
    """Get all predictions for a dataset by running it in small batches."""
    size = data.shape[0]
    if size < EVAL_BATCH_SIZE:
        raise ValueError("batch size for evals larger than dataset: %d" % size)
    predictions = np.ndarray(shape=(size, NUM_LABELS), dtype=np.float32)
    for begin in xrange(0, size, EVAL_BATCH_SIZE):
        end = begin + EVAL_BATCH_SIZE
        if end <= size:
            predictions[begin:end, :] = sess.run(
                eval_prediction,
                feed_dict={eval_data: data[begin:end, ...]})
        else:
            batch_predictions = sess.run(
                eval_prediction,
                feed_dict={eval_data: data[-EVAL_BATCH_SIZE:, ...]})
            predictions[begin:, :] = batch_predictions[begin - size:, :]
    return predictions

# Same as eval_in_batches but for testing
def test_in_batches(data, sess):
    predictions = sess.run(
        test_prediction,
        feed_dict={test_data: data})

    return predictions

# Create a local session to run the training.
start_time = time.time()
losses = []
epochs = []
saver = tf.train.Saver()

# Define the fraction of GPU memory
# gpu_options = tf.GPUOptions(per_process_gpu_memory_fraction=0.9)
# Launch the graph
# with tf.Session(graph=graph, config=tf.ConfigProto(gpu_options=gpu_options)) as
sess:
with tf.Session(graph=graph) as sess:

    if FLAGS.train:
        # Deletes the previous saved models
        rmdir(CHECKPOINT_DIR)
        mkdir(CHECKPOINT_DIR)

        # Run all the initializers to prepare the trainable parameters.
        tf.initialize_all_variables().run()
        #tf.global_variables_initializer

        # Prepare data
        train_images = [x for y in imagesTrain for x in y]
        train_images = np.array(train_images)
        train_labels = [np.float32(np.zeros(shape=(len(x))+1) for i,x in
enumerate(imagesTrain))]
        train_labels = [x for y in train_labels for x in y]
        train_labels = np.array(train_labels)
        # Randomise training set
        perm = np.arange(len(train_images))
        np.random.shuffle(perm)
        train_images = train_images[perm]
        train_labels = train_labels[perm]

        # Select randomly the validation set
        perm = np.arange(len(train_images))
        np.random.shuffle(perm)
        sel = perm[:VALIDATION_IMAGES]
        validationimages = train_images[sel]
        validationlabels = train_labels[sel]
        mask = np.ones(len(train_images), dtype=bool) # all elements included/True.
        mask[sel] = False
        train_im = train_images[mask]
        train_lab = train_labels[mask]

        """
        Extract the images into a 4D tensor [image index, y, x, channels].

```

```

Values are rescaled from [0, 255] down to [-0.5, 0.5].
"""
batch_im_train =
np.float32(np.zeros(shape=(len(train_im), IMAGE_SIZE_W, IMAGE_SIZE_H, NUM_CHANNELS)))
batch_labels_train = np.zeros(shape=(len(train_lab)))
for j, im_str in enumerate(train_im):
    im = Image.open(im_str).convert('L')
    # print ('index ', j)
    # print ('imatge que es tracte' + im_str)
    # if (j%25 == 0):
    # im.show()
    im = np.reshape(im, (IMAGE_SIZE_W, IMAGE_SIZE_H))
    im = (im - (PIXEL_DEPTH / 2.0)) / PIXEL_DEPTH
    # if (j%25 == 0):
    # print ('imatge passada a vector')
    # print (im.shape)
    # print (im)
    batch_im_train[j, :, :, 0] = im
    label = train_lab[j]
    batch_labels_train[j] = label
    # print ('xxxxxxx label de la imatge: ' + im_str)
    # print (label)

    batch_im_validation =
np.float32(np.zeros(shape=(len(validationimages), IMAGE_SIZE_W, IMAGE_SIZE_H, NUM_CHANNELS)
))

batch_labels_validation = np.zeros(shape=(len(validationimages)))
for j, im_str in enumerate(validationimages):
    im = Image.open(im_str).convert('L')
    im = np.reshape(im, (IMAGE_SIZE_W, IMAGE_SIZE_H))
    im = (im - (PIXEL_DEPTH / 2.0)) / PIXEL_DEPTH
    batch_im_validation[j, :, :, 0] = im
    label = validationlabels[j]
    batch_labels_validation[j] = label

# Generate test batch
testimages = []
testlabels = []
for i, testSet in enumerate(imagesTest):
    testimages = np.hstack((testimages, testSet))
    testSet_lab = np.float32(np.zeros(shape=(len(testSet))))+i
    testlabels = np.hstack((testlabels, testSet_lab))

batch_im_test =
np.float32(np.zeros(shape=(len(testimages), IMAGE_SIZE_W, IMAGE_SIZE_H, NUM_CHANNELS)))
batch_labels_test = np.zeros(shape=(len(testimages)))
for j, im_str in enumerate(testimages):
    im = Image.open(im_str).convert('L')
    im = np.reshape(im, (IMAGE_SIZE_W, IMAGE_SIZE_H))
    im = (im - (PIXEL_DEPTH / 2.0)) / PIXEL_DEPTH
    batch_im_test[j, :, :, 0] = im
    label = testlabels[j]
    batch_labels_test[j] = label

step = 1
train_size = len(train_im)
# Loop through training steps. Note: // is floor division
while step * BATCH_SIZE < num_epochs * train_size:
    # Compute the offset of the current minibatch in the data.
    # Note that we could use better randomization across epochs.
    epoch = int(step*BATCH_SIZE/train_size) + 1
    offset = (step * BATCH_SIZE) % (train_size - BATCH_SIZE)
    batch_data = batch_im_train[offset:(offset + BATCH_SIZE), ...]
    batch_labels = batch_labels_train[offset:(offset + BATCH_SIZE)]

    # This dictionary maps the batch data (as a numpy array) to the
    # node in the graph it should be fed to.
    feed_dict = {train_data_node: batch_data,
                  train_labels_node: batch_labels}

    # Run the graph and fetch some of the nodes.
    _, l, lr, predictions = sess.run(
        [optimizer, loss, learning_rate, train_prediction],
        feed_dict=feed_dict)

    epoch = float(step) * BATCH_SIZE / train_size

```

```

    if step % EVAL_FREQUENCY == 0:
        acc_batch = accuracy_rate(predictions, batch_labels)
        eval_pred = eval_in_batches(batch_im_validation, sess)
        val_acc = accuracy_rate(eval_pred, batch_labels_validation)
        format_str = ('%s: Epoch %d, LR: %.8f, Minibatch Loss = %.5f, Training
Accuracy=%.5f, Val_Acc=%.5f')
        format_strf = ('%s: Epoch %d, LR: %.8f, Minibatch Loss = %.5f, Training
Accuracy=%.5f, Val_Acc=%.5f\n')
        print (format_str % (datetime.now(), epoch, lr, l, acc_batch, val_acc))
        f.write (format_strf % (datetime.now(), epoch, lr, l, acc_batch, val_acc))
        losses.append(l)
        epochs.append(epoch)
        step += 1

# Save the model
saver.save(sess, CHECKPOINT_DIR + 'model.ckpt', global_step=step)

# Test
predictions_test = test_in_batches(batch_im_test, sess)
print_test_labels(predictions_test, batch_labels_test, testimages)

test_acc = accuracy_rate(predictions_test, batch_labels_test) * 100.0
print('Test accuracy: %.2f%%' % test_acc)
f.write('Test accuracy: %.2f%%\n' % test_acc)

f.close()
f2.close()

# Plot learning curve
import matplotlib.pyplot as plt
plt.plot(epochs, losses, 'ro', label= 'losses by epochs')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.show()

else:
    ckpt = tf.train.get_checkpoint_state(CHECKPOINT_DIR)
    if ckpt and ckpt.model_checkpoint_path:
        saver.restore(sess, ckpt.model_checkpoint_path)
        print('Model restored correctly')
    else:
        print('no checkpoint found...')
    predictions_test = test_in_batches(batch_im_test, sess)
    test_error = error_rate(predictions_test, batch_labels_test)
    print_test_labels(predictions_test, batch_labels_test, testimages)
    print('Test error: %.1f%%' % test_error)
    f.write('Test error: %.1f%%\n' % test_error)

if __name__ == '__main__':
    tf.app.run()

```