Reconeixement d'Emocions en Temps Real fent servir Xarxes Neuronals Convolucionals

Informe de Progrés I

1 Introducció

Aquest informe de progrés té com a objectiu documentar el desenvolupament setmanal del projecte de reconeixement d'emocions en temps real mitjançant xarxes neuronals convolucionals (CNN). El projecte està estructurat seguint una metodologia basada en cicles de treball setmanals, en què es defineixen els objectius i es fa una avaluació del progrés. Així com dels problemes o desafiaments que han sorgit durant cada setmana. Això permet ajustar el procés de desenvolupament segons les necessitats que sorgeixin en cada etapa.

El projecte busca implementar un sistema que pugui detectar rostres en temps real a partir de vídeos i, a continuació, classificar les emocions mitjançant una CNN. Aquest informe cobreix la introducció de les eines emprades, la configuració inicial del projecte, el progrés setmanal i els propers passos previstos per aconseguir els objectius del Treball de Fi de Grau.

2 Eines i Llibreries Utilitzades

En aquest projecte s'han utilitzat les següents eines i biblioteques per a la implementació i entrenament del model:

- **Python 3.9**: Llenguatge de programació principal per a tot el desenvolupament.

Python és àmpliament utilitzat en machine learning i deep learning per la seva simplicitat i per la gran quantitat de biblioteques especialitzades, com PyTorch, TensorFlow i Keras. És altament accessible i té una gran comunitat de suport, cosa que facilita trobar exemples, documentació i suport tècnic. A més, Python és conegut per la seva eficiència en treballar amb dades, que és crucial en projectes de xarxes neuronals convolucionals.

- **PyTorch i Torchvision**: Paquet principal per a crear i entrenar les xarxes neuronals. Es va seleccionar ResNet-18 com a base del model CNN, que s'ha modificat per a la classificació d'emocions.

  Alternatives: **TensorFlow/Keras** i **MXNet** són altres opcions populars per a implementar i entrenar xarxes neuronals.

PyTorch destaca per la seva simplicitat i flexibilitat gràcies a la seva naturalesa de gràfics dinàmics, que fa que sigui intuïtiu per a investigadors i estudiants. Torchvision és també molt convenient, ja que inclou models preentrenats i transformacions de dades especialment dissenyades per a visió per computador. TensorFlow, per exemple, pot ser una mica més complex en termes de definició de gràfics, mentre que PyTorch és més semblant a un codi Python estàndard, cosa que fa que sigui més fàcil de depurar i adaptar durant l'entrenament.

- **OpenCV**: Biblioteca per al processament de vídeo i la detecció de rostres en temps real, fonamental per obtenir els rostres sobre els quals s'aplicarà el model d'emocions.

  Alternatives: **Dlib** i **MediaPipe** són altres biblioteques conegudes per al processament d'imatges i detecció facial en temps real.

OpenCV és molt versàtil i optimitzat per al processament de vídeo i imatges en temps real, amb una àmplia gamma de funcions que cobreixen des de la detecció de rostres fins a filtres d'imatge i transformacions geomètriques. Dlib, tot i que és més senzill per a la detecció de rostres, té menys funcionalitats per al processament d'imatges a gran escala. MediaPipe és una bona alternativa per a aplicacions avançades de detecció de rostres i seguiment, però OpenCV és preferible en aquest cas pel seu suport extensiu i per la facilitat amb què es pot integrar amb altres eines de Python.

- **Pandas i Matplotlib**: Per a la manipulació de dades i visualització de resultats durant el procés de desenvolupament i avaluació del model.

  Alternatives: **NumPy** o **Seaborn** són alternatives útils en anàlisi de dades i visualització.

Pandas és una biblioteca molt potent per a la manipulació de dades, especialment en la gestió de grans conjunts de dades, com ara el que utilitzes per entrenar la xarxa. Matplotlib, per la seva banda, és molt flexible per crear gràfics detallats, i té una integració perfecta amb Pandas per visualitzar fàcilment estadístiques d'entrenament o resultats d'evaluació. NumPy és més limitat per a la manipulació complexa de dades, i Seaborn, tot i ser molt bo per a gràfics avançats, depèn de Matplotlib i està orientat principalment a l'anàlisi estadística.

- **TQDM**: Per monitoritzar el progrés durant l'entrenament del model, la qual cosa és especialment útil per a bucles llargs.

  Alternatives: **Rich** o **Progress** són altres opcions per monitoritzar el progrés en processos de Python.

TQDM és extremadament lleuger i fàcil d'usar amb només una línia de codi, i és molt eficaç per fer seguiment del progrés d'entrenament de models en bucles llargs, la qual cosa és útil en aquest projecte. A més, té una àmplia compatibilitat amb altres biblioteques de Python com PyTorch i Pandas. Tot i que Rich ofereix opcions visuals més avançades, TQDM és més

senzill i es concentra exclusivament en el seguiment del progrés, fet que el fa molt eficient en entorns de desenvolupament i entrenament de models.

3 Estructura General del Projecte

Per crear un sistema de reconeixement d'emocions en temps real utilitzant xarxes neuronals convolucionals (CNN), seguim un procés que combina la preparació de dades, la construcció d'un model d'intel·ligència artificial i el seu entrenament, i finalment, la implementació en un entorn de vídeo en temps real. A continuació, t'explico de manera detallada cada pas d'aquest procés, utilitzant les eines descrites en l'informe de progrés.

Preparació del Conjunt de Dades d'Emocions (FERPlus)

Per entrenar el model perquè reconegui emocions facials, és fonamental disposar d'un conjunt de dades amb imatges de rostres anotats amb emocions. El conjunt de dades escollit és **FERPlus**, una ampliació del conegut conjunt FER2013, que corregeix i millora la qualitat de les anotacions emocionals per augmentar la precisió en estudis de reconeixement d'emocions facials. FERPlus inclou aproximadament 35.887 imatges anotades amb 8 emocions bàsiques i dues categories addicionals: "Unknown" (emoció desconeguda) i "Non-Face" (sense rostre).

Altres Conjunts de Dades de Reconeixement d'Emocions

1. **AffectNet**: Un dels conjunts més extensos, amb més d'un milió d'imatges. Inclou 450.000 anotacions emocionals amb una diversitat àmplia. Tanmateix, requereix permisos específics per accedir-hi i és més complex de manipular.
2. **CK+ (Extended Cohn-Kanade Dataset)**: CK+ és un conjunt de dades molt utilitzat en estudis de reconeixement d'emocions, especialment per a l'anàlisi d'expressions facials en seqüències de vídeo. Tot i així, és relativament petit, amb menys d'un miler de seqüències d'emocions, i per tant pot no ser suficient per entrenar un model robust que generalitzi bé.
3. **JAFFE (Japanese Female Facial Expression)**: Conté imatges de cares femenines japoneses amb diferents expressions. Tot i ser molt conegut, és limitat perquè només inclou cares de dones d'una sola procedència cultural, la qual cosa pot limitar la capacitat del model de generalitzar.
4. **SAVEE (Surrey Audio-Visual Expressed Emotion)**: És un conjunt de dades que inclou seqüències de vídeo i àudio, amb diverses emocions representades per persones de gènere masculí. És útil per combinar informació de vídeo i àudio, però és limitat en diversitat i volum d'imatges.

Per què FERPlus és una Millor Opció?

FERPlus ofereix avantatges significatius enfront d'altres conjunts:

- **Correcció d'Anotacions**: Les anotacions d'emocions han estat revisades manualment per assegurar una classificació més precisa, reduint errors d'etiquetatge del FER2013 original.
- **Emocions Equilibrades**: Inclou categories emocionals àmpliament reconegudes en estudis psicològics, facilitant la generalització del model.
- **Accessibilitat i Simplicitat**: FERPlus és públicament accessible i menys complex de manipular, ideal per iteracions ràpides i experiments. Aquestes característiques fan que sigui un dels models més utilitzats i per tant amb més dades i suport.

Les 8 Emocions Utilitzades

En aquest projecte, es treballa amb les següents 8 emocions, que es consideren bàsiques per la seva reconeixibilitat en expressions facials:

Train	
Emoció	**Quantitat d'imatges etiquetades**
Neutre	10.294
Alegria	7.526
Tristesa	3.530
Sorpresa	3.557
Por	655
Disgust	191
Enuig	2463
Desig	168
Unknown	171
Non-Face	2
	Total: 28.557

Validació	
Emoció	**Quantitat d'imatges etiquetades**
Neutre	1.328
Alegria	899
Tristesa	415
Sorpresa	458
Por	71
Disgust	32
Enuig	325
Desig	26
Unknown	23
Non-Face	1
	Total: 3.578

Test	
Emoció	**Quantitat d'imatges etiquetades**
Neutre	1.258
Alegria	928
Tristesa	446
Sorpresa	447
Por	97
Disgust	20
Enuig	321
Desig	27
Unknown	28
Non-Face	0
	Total: 3.572

Aquestes emocions han estat triades perquè cobreixen un espectre ampli de les expressions emocionals més comunes i recognoscibles a través de les expressions facials. A més, són categories que es troben amb alta precisió a **FERPlus**, cosa que permet que el model aprengui amb exemples diversos i equilibrats per a cada emoció.

Sistema de Votació a FERPlus

El conjunt de dades **FERPlus** es basa en un sistema d'anotació col·laboratiu on **10 persones** revisen cada imatge i voten per l'emoció que creuen que està més representada. Això proporciona una etiqueta consensuada per a cada imatge, millorant significativament la qualitat de les dades.

Format del CSV a FERPlus

El conjunt de dades es troba en format CSV, amb les següents columnes:

1. **Nom de la Imatge**: El nom del fitxer d'imatge (ex. fer0032220.png).
2. **Bounding Box**: Coordenades de la caixa delimitadora del rostre (ex. (0, 0, 48, 48)).
3. **Vots per Emoció**: 10 columnes que indiquen el nombre de vots per cada emoció, en aquest ordre:
    - Neutral
    - Happiness
    - Surprise
    - Sadness
    - Anger
    - Disgust
    - Fear
    - Contempt
    - Unknown
    - Non-Face

Exemple d'una fila del CSV:

fer0032220.png, "(0, 0, 48, 48)", 0, 0, 0, 0, 2, 1, 0, 7, 0, 0

En aquest exemple:

- **fer0032220.png**: Nom de l'arxiu d'imatge.
- **(0, 0, 48, 48)**: Bounding box del rostre dins de la imatge.
- **0, 0, 0, 0, 2, 1, 0, 7, 0, 0**: Vots per emoció. L'emoció amb més vots és **Contempt** (7 vots).

Raons per Treballar amb aquestes 8 Emocions:

1. **Alegria, Tristesa, Enuig, Por, Sorpresa i Disgust**: Són emocions bàsiques reconegudes per la majoria dels estudis psicològics, amb expressions facials úniques i fàcilment identificables.
2. **Neutre**: Essencial per distingir rostres sense emoció aparent d'expressions emocionals.
3. **Desig**: Sovint subtil, aporta complexitat i profunditat a l'anàlisi emocional.

Categories Addicionals:

- **Unknown**: Assignada a imatges amb emocions ambigües o sense consens clar.
- **Non-Face**: Imatges sense cap rostre identificable.

Beneficis de FERPlus

Aquest conjunt de dades destaca per:

- **Qualitat en les Anotacions**: El sistema de votació consensuat assegura etiquetes fiables i consistents.
- **Accés Públic**: FERPlus és obert per a investigadors, cosa que facilita el seu ús en experiments.
- **Diversitat i Equilibri**: Ofereix un ventall ampli de situacions emocionals i expressions facials.

Aquest sistema d'anotació i votació fa que **FERPlus** sigui una opció sòlida per entrenar models robusts en reconeixement d'emocions, preparant-los per identificar patrons emocionals tant evidents com subtils en contextos variats.

Preprocessament de les Imatges

Abans d'entrenar el model, cal que les imatges del conjunt de dades estiguin en un format que la xarxa pugui processar. En aquest projecte, utilitzem l'arxiu normalize.py per realitzar el preprocessament de les imatges, que inclou:

1. **Escalat**: Totes les imatges es redimensionen a una mida estàndard de 224x224 píxels, que és la mida d'entrada del model ResNet-18.
2. **Transformació a Tensors**: Les imatges es converteixen en tensors, una estructura de dades numèrica que PyTorch pot utilitzar per entrenar el model.
3. **Normalització**: Els valors de color de les imatges es normalitzen perquè estiguin dins d'un rang específic. Això millora la velocitat d'aprenentatge i ajuda el model a convergir més ràpidament.

A més, normalize.py inclou una funció per visualitzar un lot d'imatges preprocessades. Això permet comprovar que les imatges s'estan carregant i processant correctament abans d'iniciar l'entrenament.

Construcció del Model de Xarxa Neuronal (ResNet)

Una xarxa neuronal és un sistema de processament d'informació format per diverses **neurones** organitzades en capes. Cada neurona rep dades, les transforma i les transmet a altres neurones, com si fos una cadena de missatges. Aquest flux permet que la informació es vagi modificant a través de la xarxa, cosa que facilita l'aprenentatge.

Quan volem que una xarxa neuronal identifiqui alguna cosa, com ara una emoció en un vídeo, primer la "entrenem" mostrant-li moltes imatges de rostres amb diferents emocions i indicant-li quina emoció correspon a cada imatge. Al principi, comet molts errors, però, amb cada correcció, ajusta les seves connexions per reconèixer millor els patrons.

ResNet (Residual Network)

ResNet (Residual Network) és una arquitectura de xarxa neuronal convolucional profunda desenvolupada per Microsoft Research, especialment popular per la seva capacitat d'entrenar xarxes molt profundes de manera eficient. La principal innovació de ResNet és l'ús de **connexions residuals** o **connexions de salt** ("skip connections"), que ajuden a evitar el problema de la desaparició de gradients en xarxes molt profundes. Aquestes connexions residuals permeten que el model aprengui més profundament, ja que faciliten el flux de la informació i els gradients entre les capes.

ResNet-18

- **Estructura**: ResNet-18 és una versió de la xarxa amb 18 capes (capes d'aprenentatge profund, que inclouen convolucions i capes completament connectades). És una de les arquitectures més lleugeres de la família ResNet i requereix menys poder computacional i memòria, cosa que la fa adequada per a aplicacions on els recursos de càlcul són limitats.
- **Aplicació**: Ideal per a tasques de classificació amb conjunts de dades de mida moderada i aplicacions que requereixin una inferència ràpida. La seva menor profunditat significa que pot ser entrenada més ràpidament i és menys susceptible a l'overfitting en conjunts de dades relativament petits o moderats.

ResNet-50

- **Estructura**: ResNet-50 és una versió molt més profunda amb 50 capes, on es fan servir capes de blocs residuals més complexos (coneguts com a "bottleneck blocks") amb tres convolucions en comptes de dues. Aquesta profunditat afegeix més capacitat d'aprenentatge, permetent al model capturar patrons i característiques més complexes en les imatges.
- **Aplicació**: ResNet-50 és més adequat per a tasques on es requereix una major precisió i es disposa d'un conjunt de dades gran. La seva major profunditat li permet captar detalls més complexos en les imatges, però també requereix més recursos computacionals i pot ser més susceptible a l'overfitting en conjunts de dades petits.

Diferències entre ResNet-18 i ResNet-50

1. **Nombre de Capes**: ResNet-18 té 18 capes, mentre que ResNet-50 en té 50. Això significa que ResNet-50 pot capturar patrons més complexos, però també implica un cost computacional més alt.
2. **Blocs Residuals**: ResNet-18 utilitza un tipus de bloc residual més senzill amb dues capes de convolució, mentre que ResNet-50 utilitza "bottleneck blocks" amb tres capes de convolució per bloc, fet que augmenta la capacitat de representació.
3. **Rendiment i Precisió**: ResNet-50 tendeix a oferir una millor precisió en tasques de classificació, especialment en conjunts de dades grans i complexos. ResNet-18, tot i ser menys precís, és més ràpid d'entrenar i consumeix menys memòria, cosa que el fa més pràctic per a aplicacions amb recursos limitats.
4. **Aplicabilitat**: ResNet-18 és ideal per a tasques amb restriccions de temps o potència de càlcul (per exemple, inferència en dispositius mòbils o temps real). ResNet-50, en canvi, és més adequat per a aplicacions d'alta precisió en entorns on es disposa de més recursos de processament, com en servidors o sistemes amb potents GPU.

Per a aquest projecte, hem escollit utilitzar la xarxa neuronal **ResNet-18** com a punt de partida pel seu menor cost computacional, la qual cosa ens permet entrenar el model ràpidament i obtenir resultats inicials.

Tot i així, estem considerant realitzar proves amb una xarxa més profunda, com **ResNet-50**, que compta amb 50 capes. Aquesta major profunditat li dona més capacitat per capturar detalls complexos en les imatges, fet que podria convertir-la en una opció més robusta per a reconèixer emocions en contextos variats. Encara que ResNet-50 requereix més temps d'entrenament i més recursos de càlcul, podria millorar la precisió i la capacitat de generalització del model, especialment en aplicacions en temps real.

L'objectiu de provar amb ResNet-50 és comparar el rendiment amb ResNet-18 i determinar quina opció ens dona un model més robust i capaç d'interpretar amb precisió una àmplia gamma d'emocions en temps real.

Entrenament del Model

L'entrenament del model és un procés fonamental en aquest projecte, ja que determina la capacitat del model de reconèixer emocions facials amb precisió. En aquest cas, utilitzem el conjunt de dades **FERPlus**, que inclou imatges etiquetades amb 10 categories emocionals i implementa un criteri avançat per determinar quina emoció predomina en cada imatge. Aquest criteri assegura que les etiquetes siguin robustes i fiables, la qual cosa és essencial per a un aprenentatge eficaç.

L'entrenament implica passar iterativament per les dades, ajustar els pesos del model i validar-ne el rendiment per assegurar que aprèn de manera òptima. Aquí expliquem detalladament cada pas i les tècniques aplicades per millorar el rendiment i prevenir problemes com l'overfitting.

Preparació del Conjunt de Dades

El conjunt de dades **FERPlus** inclou imatges etiquetades per 10 persones, i cada imatge està anotada amb **10 categories emocionals**: Neutral, Happiness, Surprise, Sadness, Anger, Disgust, Fear, Contempt, Unknown i Non-Face.

Per assignar una emoció dominant a una imatge, es segueix la següent regla:

1. Es compara el nombre de vots entre les dues emocions més votades.
2. Si l'emoció amb més vots supera la segona per almenys un **30% (3 vots més)**, aquesta es considera la predominant.
3. Si cap emoció compleix aquest criteri, la imatge es classifica com a Unknown.

Aquest enfocament evita ambigüitats i assegura que les etiquetes reflecteixin patrons consistents en les dades, millorant així la qualitat de l'aprenentatge del model.

Passos d'Entrenament

L'entrenament del model segueix una sèrie de passos que asseguren que aquest aprengui de manera eficient i robusta:

1. **Propagació Endavant (Forward Pass):**
   - Per a cada lot d'imatges (batch), les imatges es passen a través de les capes del model.
   - Les operacions com convolucions i funcions d'activació extreuen característiques de les imatges fins arribar a la capa final.
   - La sortida és una probabilitat associada a cada emoció per cada imatge.
2. **Càlcul de l'Error (Pèrdua):**
   - Amb les prediccions obtingudes, es calcula la discrepància respecte a les etiquetes reals utilitzant la funció de pèrdua CrossEntropyLoss.
   - Aquesta funció genera un valor numèric que representa l'error. Un valor més alt indica que les prediccions són menys precises.
3. **Retropropagació (Backward Pass):**
   - A partir de l'error, es calculen els gradients respecte als pesos del model.
   - Aquest procés és essencial per ajustar els pesos de les connexions neuronals i millorar les prediccions futures.
4. **Actualització dels Pesos:**
   - S'utilitza l'optimitzador **Adam**, que adapta automàticament el ritme d'aprenentatge per a cada pes.
   - Això permet que el model convergeixi més ràpidament i amb més precisió, ajustant els pesos de manera òptima.
5. **Validació i Monitorització:**
   - Després de cada època d'entrenament, es valida el model amb un conjunt de dades separat.
   - Es registren la pèrdua i la precisió de validació per assegurar que el model generalitza bé i no s'ajusta massa a les dades d'entrenament.

Proves amb Diferents Paràmetres

Per assegurar el millor rendiment possible, s'experimenta amb diverses configuracions de paràmetres:

1. **Nombre d'Èpoques:**
    - Determinem el nombre òptim d'èpoques per evitar tant el subentrenament (poques èpoques) com l'overfitting (massa èpoques).
    - Les proves actuals es fan amb un màxim de **50 èpoques**, però es monitoritza constantment l'evolució per ajustar aquest valor.
2. **Batch Size:**
    - Els lots més grans (64 o 128 imatges) permeten entrenar més ràpidament, però requereixen més memòria GPU.
    - Lots més petits (32 imatges) poden donar lloc a una millora de la precisió, però l'entrenament triga més temps.
3. **Learning Rate:**
    - Comencem amb un valor inicial de **0.001** i experimentem amb valors més baixos (0.0005) i més alts (0.005).
    - Això ens ajuda a trobar l'equilibri entre una convergència ràpida i una precisió òptima.
4. **Arquitectura del Model:**
    - S'avaluen **ResNet-18** (ràpida i eficient) i **ResNet-50** (més precisa però més lenta).
    - Aquesta comparació permet trobar un compromís entre precisió i eficiència segons les necessitats del projecte.
5. **Regularització amb Dropout:**
    - S'experimenta amb taxes de dropout de **0.2**, **0.3** i **0.5** per reduir l'overfitting i millorar la generalització.

Prevenció de l'Overfitting

Per assegurar que el model no s'ajusti massa a les dades d'entrenament:

- **Early Stopping:** Es deté l'entrenament si la pèrdua de validació no millora durant diverses èpoques consecutives.
- **Augmentació de Dades:** S'apliquen transformacions com rotacions, flips horitzontals i ajustos de lluminositat per generar variacions en les dades d'entrenament.
- **Validació Contínua:** Es compara constantment la pèrdua i la precisió d'entrenament amb les de validació per detectar signes d'overfitting.

Resultats Esperats i Seguiment del Rendiment

Amb aquest enfocament rigorós, esperem obtenir un model que:

1. **Reconegui emocions amb alta precisió** fins i tot en imatges no vistes durant l'entrenament.
2. **Generalitzi bé** per a aplicacions en temps real, sense ser massa sensible a variacions en les imatges.

3. **Sigui eficient** i balancegi precisió i temps d'entrenament, optimitzant l'ús dels recursos disponibles.

Els resultats de cada configuració es registren en un arxiu CSV per avaluar el rendiment de cada prova i seleccionar els paràmetres més adequats. Aquest procés iteratiu ens permet ajustar contínuament el model fins aconseguir el millor rendiment possible per a una aplicació real de reconeixement d'emocions.

Implementació del Reconeixement en Temps Real amb Vídeo

Un cop el model està entrenat i ha après a reconèixer emocions amb precisió, el podem aplicar a imatges en temps real. En aquest cas, utilitzem **OpenCV** per capturar vídeo i detectar les cares de les persones a cada fotograma. El procés és el següent:

1. **Captura de Vídeo**: OpenCV captura cada fotograma del vídeo en temps real. Això és com agafar una foto ràpida cada vegada que hi ha moviment.
2. **Detecció de la Cara**: En cada fotograma, OpenCV busca i identifica la cara de la persona. Aquesta detecció de la cara ajuda a centrar el reconeixement en la cara, evitant altres parts de la imatge que podrien distreure el model.
3. **Predicció d'Emocions en Temps Real**: Cada vegada que OpenCV detecta una cara, envia la imatge d'aquesta cara al model entrenat. El model analitza la cara i dona una predicció de l'emoció en temps real. Aquest procés es repeteix ràpidament, de manera que el sistema pot actualitzar la predicció d'emoció constantment.

4 Progressió Setmanal del Desenvolupament

En la planificació d'aquesta primera fase del projecte, es va tenir en compte la limitació de temps disponible degut a les dates en què ens trobem, coincidint amb exàmens i pràctiques d'altres assignatures. Això redueix significativament el temps que puc dedicar al projecte. Per aquest motiu, la duració assignada a cada tasca no reflecteix únicament el temps necessari per completar-la, sinó el temps màxim disponible per assegurar que les tasques essencials es completin dins del termini previst. Aquesta planificació ha permès ajustar els objectius de manera realista segons les circumstàncies.

Tasca completada dins del termini estipulat	Tasca en curs	Tasca pendent d'inici	Tasca descartada del projecte

Lliurament	10/11/2024		*Informe Progrés 1*
Setmana	**Dates**	**Duració**	**Tasques**
4	07/10-10/10	4 dies	Selecció del dataset

- Es va realitzar una anàlisi exhaustiva dels conjunts de dades més coneguts per al reconeixement d'emocions, incloent AffectNet, FER+, CK+, JAFFE i SAVEE.

La selecció del conjunt de dades és una etapa crucial en el desenvolupament del projecte, ja que requereix una recerca exhaustiva dels conjunts de dades més utilitzats per al **Reconeixement d'Emocions en Temps Real mitjançant Xarxes Neuronals Convolucionals** o altres aplicacions similars. Aquesta investigació no es limita només a identificar els datasets disponibles, sinó també a analitzar les seves característiques, com el volum d'imatges, les anotacions d'emocions, la diversitat cultural i demogràfica, i les seves limitacions.

-Es va escollir **FERPlus** com a conjunt de dades principal per diverses raons:

- La seva qualitat d'anotacions, ja que les emocions de cada imatge han estat votades per 10 persones, aplicant un criteri avançat per determinar l'emoció dominant.
- La diversitat d'emocions que cobreix, incloent 10 categories emocionals com Neutral, Happiness, Surprise, Sadness, Anger, Disgust, Fear, Contempt, Unknown i Non-Face, fet que permet un entrenament complet i equilibrat.
- La presència d'un volum considerable d'imatges en els subconjunts d'entrenament, validació i prova, suficient per entrenar i validar models amb precisió.
- La seva accessibilitat i la quantitat de dades i suport que hi ha.

- Problemes trobats:

El **dataset FERPlus** està etiquetat per 10 persones, però no sempre hi ha consens sobre l'emoció predominant. Per decidir quina emoció assignar a cada imatge, s'ha implementat una regla:

1. **Criteri de dominància**: Una emoció es considera predominant si té **3 vots més** que la segona emoció més votada (un marge del 30%).
2. **Assignació de Unknown**: Si cap emoció compleix aquest criteri, l'etiqueta de l'imatge és Unknown.

Aquesta regla ha augmentat significativament la quantitat d'imatges etiquetades com a Unknown, ja que moltes imatges no tenen un consens clar. Tot i reduir el volum d'imatges amb etiquetes concretes, millora la qualitat de les etiquetes utilitzades en l'entrenament del model.

Aquesta tasca es va iniciar durant la preparació de l'informe inicial, en el marc de la recerca per a l'estat de l'art. En aquell moment, es va fer una anàlisi teòrica dels datasets més rellevants, destacant-ne les seves aplicacions i avantatges. Inicialment, es va utilitzar un dataset que es pensava que era l'AffectNet, però més tard es va descobrir que era una aparent adaptació feta per un particular, amb moltes menys dades i qualitat.

Com que per utilitzar l'AffectNet original cal demanar accés i seguir una sèrie de normes, es va decidir canviar al dataset FERPlus, que és públic i proporciona etiquetes precises anotades per 10 persones. Aquest canvi va requerir ajustar la metodologia per treballar amb les dades disponibles.

Tot i el canvi, encara quedava pendent el procés pràctic de descarregar el dataset seleccionat i estudiar-lo detalladament. Aquest estudi inclou entendre com està estructurat (per exemple, com estan organitzades les imatges i les etiquetes), identificar possibles problemes o manques, i preparar-lo per ser manipulat eficientment durant el preprocessament i l'entrenament del model. Aquest aprofundiment és necessari per assegurar que el conjunt de dades s'ajusti perfectament als requeriments del projecte i per planificar com es gestionaran les dades durant el flux de treball.

4	11/10-13/10	3 dies	Normalització d'imatges

- Es van implementar transformacions per preparar les imatges, incloent redimensionar-les, convertir-les a tensors i aplicar normalitzacions. També es va desenvolupar una funció de visualització per comprovar el preprocessament.

- Problemes trobats:

Al principi, vaig cometre un error de principiant al intentar carregar les imatges una a una sense utilitzar lots (batch). Aquest procés va provocar que el meu ordinador es quedés sense memòria RAM, i el sistema s'apagués. Després de buscar informació, vaig comprendre que la manera correcta de treballar amb xarxes neuronals és carregar les imatges en lots, utilitzant data loaders per gestionar les dades de manera eficient.

Els **dataloaders** són una eina fonamental a l'hora de treballar amb xarxes neuronals i grans conjunts de dades. La seva funció principal és carregar les dades en lots o **batches** de manera eficient, evitant que el sistema es sobrecarregui amb massa informació alhora i permetent un entrenament més ràpid i robust.

Aquesta tasca, si tens experiència prèvia en realitzar processos similars, no hauria de requerir més d'una tarda per completar-la. Tot i això, si mai has treballat amb aquest tipus

d'activitat, és possible que necessitis fins a dos dies per familiaritzar-te amb les eines, entendre els passos necessaris i resoldre qualsevol problema o dubte que pugui sorgir durant el procés, això és el que em va passar a mi.

5	14/10-20/10	7 dies	Preparació dels conjunts de dades (validació, entrenament, prova)

- Com que **FERPlus** ja inclou aquesta divisió, no ha estat necessari repartir manualment les dades entre entrenament, validació i prova. Aquesta divisió garanteix una separació coherent i uniforme entre els diferents conjunts, assegurant que:

- El conjunt d'entrenament conté suficients dades per aprendre.
- El conjunt de validació permet mesurar el rendiment en dades no vistes durant l'entrenament.
- El conjunt de prova proporciona una avaluació independent del model un cop finalitzat l'entrenament.

Aquesta divisió facilita un flux de treball consistent i estandarditzat, ideal per obtenir resultats fiables.

Tant **FERPlus** com **AffectNet** ja venien amb les dades prèviament dividides en els conjunts de **validació**, **entrenament** i **prova**.

En lloc d'invertir els 7 dies estipulats en aquesta tasca, només va ser necessari dedicar-hi una tarda per familiaritzar-me amb el dataset. Durant aquest temps es van explorar aspectes com:

- El format dels fitxers CSV associats a cada conjunt.
- La distribució de les imatges entre les diferents emocions i conjunts.
- Les característiques específiques de cada dataset, com ara el nombre d'imatges disponibles per a cada emoció i les anotacions per votació.

6-8	21/10-10/11	21 dies	Disseny i implementació de l'arquitectura de la CNN. Definició de les capes i configuració dels hiperparàmetres. Entrenament inicial del model

- Es va utilitzar ResNet-18 per a la classificació en 8 emocions, ajustant la capa de sortida. Es van definir hiperparàmetres inicials (batch size, èpoques, learning rate, dropout rate) i es va iniciar l'entrenament.
- **Problemes trobats**:

Durant el procés de **disseny i implementació de l'arquitectura de la CNN, definició de capes i configuració dels hiperparàmetres, i entrenament inicial del model**, s'han identificat diversos reptes i àrees de millora:

1. **Configuració inicial amb ResNet-18**:
   Es va utilitzar **ResNet-18** com a punt de partida, configurant 10 èpoques i un batch size de 32. Amb aquesta configuració inicial, es va observar un problema significatiu d'**overfitting**: mentre que la pèrdua d'entrenament disminuïa de

manera consistent, la pèrdua de validació augmentava dràsticament després de les primeres èpoques. Això indica que el model aprenia massa bé els patrons específics de les dades d'entrenament, però no era capaç de generalitzar a dades noves.
A més, aquest primer entrenament va requerir aproximadament **5 hores**, un temps que em va semblar molt elevat, especialment sent la primera vegada que entrenava un model d'aquest tipus. Aquesta durada em va portar a buscar ajustos per reduir el temps d'entrenament.

2. **Primers ajustos per reduir el temps**:
   Per intentar disminuir la durada de l'entrenament, es van fer els següents canvis:
   - Augmentar el **batch size** a 64, cosa que permet processar més imatges per iteració i, en teoria, reduir el temps total.
   - Reduir el nombre d'èpoques.
   - Utilitzar un subconjunt més petit d'imatges per a l'entrenament.
3. Tot i que aquestes modificacions van aconseguir reduir significativament el temps d'entrenament, el problema d'overfitting va persistir. Amb menys dades i èpoques, el model continuava sense generalitzar bé, i els resultats seguien sent insuficients.
4. **Desconeixement inicial sobre el nombre d'èpoques adequat**:
   Durant les primeres proves, no tenia clar que **10 èpoques eren insuficients** per extreure conclusions sòlides, especialment amb un conjunt de dades gran i complex com AffectNet. Això em va portar a subestimar la importància d'entrenar el model durant més èpoques (idealment **100 o més**) per obtenir un aprenentatge més complet.
5. Després d'iniciar les proves amb un dataset que es creia que era **AffectNet**, es va descobrir que aquest era una **versió reduïda no oficial** creada per un particular. Davant la dificultat d'obtenir accés al dataset oficial d'AffectNet, es va decidir canviar a **FERPlus**, un conjunt de dades públic i accessible.
6. Aquest canvi va requerir l'adaptació del codi per implementar una **nova regla per determinar l'emoció predominant** en funció dels vots. Aquesta regla també va augmentar considerablement les imatges classificades com a **"Unknown"**, ja que es va exigir que l'emoció predominant tingués almenys **3 vots més** que la segona emoció més votada.
7. Un cop es va decidir el canvi de dataset a **FERPlus**, es va identificar un problema amb l'ús de la GPU. Inicialment, es creia que el model estava aprofitant la GPU, però es va descobrir que l'entrenament es feia exclusivament amb la **CPU**, cosa que augmentava dràsticament el temps necessari per època. Per solucionar aquest problema, es va haver de reinstal·lar **PyTorch** amb suport per CUDA, la qual cosa va permetre fer ús de la GPU.

Aquest canvi va tenir un impacte significatiu en el temps d'entrenament:

- **Temps amb CPU:** Entre 20 i 30 minuts per època.
- **Temps amb GPU:** Aproximadament **2 minuts per època**.

Aquesta reducció dràstica va obrir la porta a entrenar el model durant un nombre molt més gran d'èpoques, passant de proves inicials amb 10-20 èpoques a entrenaments amb **100 o més èpoques**. Això va permetre generar dades més consistents i extreure informació més

rellevant sobre el rendiment del model, evitant conclusions precipitades a causa d'entrenaments curts.

Amb l'ús de FERPlus i la GPU, es van introduir tècniques addicionals per millorar el rendiment i abordar problemes com l'**overfitting**:

1. **Dropout:**
   - Es va afegir una capa de **Dropout** amb una probabilitat de desconnexió configurable (0.2, 0.3, 0.5) abans de la capa de sortida del model.
   - Aquesta tècnica ajuda a prevenir l'overfitting en desconectar de manera aleatòria algunes connexions durant l'entrenament, obligant el model a no dependre excessivament de determinades característiques.
2. **Augmentació de dades:**
   - Es van utilitzar tècniques com **rotacions aleatòries**, canvis de **brillantor**, i **flip horitzontal** per augmentar la diversitat del conjunt d'entrenament.
   - Això permet que el model aprengui a generalitzar millor en situacions i contextos variats.
3. **Optimització del ritme d'aprenentatge (learning rate):**
   - Es van realitzar proves amb valors com **0.001** i **0.0005** per identificar la configuració òptima per minimitzar la pèrdua i maximitzar la precisió.
4. **Validació contínua i monitorització:**
   - Es van monitoritzar les mètriques de pèrdua i precisió tant en entrenament com en validació per identificar moments de **saturació** i ajustar l'estratègia en conseqüència.
   - Es va considerar la implementació de **Early Stopping** per evitar entrenaments innecessaris i prevenir l'overfitting.

8. Un cop implementades les millores actuals i amb una base d'entrenament més robusta gràcies a l'ús del dataset **FERPlus** i la GPU, el següent pas previst és explorar l'ús de **ResNet-50** per comparar els resultats amb **ResNet-18**.

Per què provar ResNet-50?

- **Capacitat més alta:** ResNet-50, amb 50 capes en comparació amb les 18 de ResNet-18, té una major capacitat per capturar patrons més complexos en les dades, cosa que pot ser especialment útil per reconèixer emocions subtils.
- **Rendiment potencialment superior:** Es preveu que ResNet-50 pugui oferir millors resultats en termes de precisió i generalització, gràcies a la seva arquitectura més profunda.
- **Impacte en el temps d'entrenament:** Tot i que ResNet-50 requereix més recursos computacionals i temps d'entrenament, la recent optimització de temps amb la GPU permet realitzar aquestes proves sense grans limitacions.

Objectiu de la Comparació

- **Avaluar l'equilibri entre precisió i eficiència:** La comparació entre ResNet-18 i ResNet-50 es farà no només basant-se en la precisió del model, sinó també en el temps d'entrenament i en la complexitat dels recursos necessaris.
- **Decidir l'arquitectura final:** Segons els resultats, es determinarà quina arquitectura ofereix una millor combinació de rendiment i eficiència per crear un model **robust** i **generalitzable**.

Amb aquesta comparació, es busca assegurar que el model final sigui capaç de reconèixer emocions amb alta precisió i que també sigui pràctic per a una implementació futura, especialment en aplicacions en temps real o en entorns amb recursos limitats. Aquesta fase serà clau per establir la base definitiva del projecte i garantir-ne l'èxit.

Aquesta tasca, si és la primera vegada que l'afrontes, pot resultar molt confusa, ja que no saps exactament quins problemes poden sorgir ni quina és la millor manera de començar. Això pot dificultar molt el procés d'organització i execució, especialment quan no tens experiència prèvia en aquest tipus de projectes.

Algú amb experiència prèvia en una tasca similar i ben documentat probablement podria completar aquesta feina en uns dies o, com a molt, en una setmana. Tanmateix, la realitat ha estat força diferent en aquest cas. A dia d'avui, aquesta tasca encara no s'ha completat del tot, tot i que ja ha passat gairebé 1 mes des del seu inici.

Aquesta demora es deu principalment al fet que vaig començar-la en plena fase d'exàmens i pràctiques d'altres assignatures, la qual cosa va reduir significativament el temps que podia dedicar al projecte. Durant aquests 21 dies, no he pogut treballar-hi amb la constància ni la intensitat necessàries per assolir-la completament abans de la data d'entrega de l'**Informe de Progrés I**.

Malgrat aquesta situació, he aprofitat aquest temps per identificar millor els problemes principals i entendre les solucions potencials, cosa que em permetrà abordar la tasca amb una estratègia més clara i efectiva en les properes setmanes.

5 Rendiment

Durant el desenvolupament, es van dur a terme proves de rendiment del model utilitzant el conjunt de validació, amb l'objectiu de mesurar la precisió i la pèrdua en cada època d'entrenament. A continuació, es mostren les mètriques obtingudes durant les proves inicials:

Èpoques (Epochs): Nombre de vegades que el model veu tot el conjunt d'entrenament.

Batch Size: Quantitat de mostres processades abans d'actualitzar els pesos del model.

Train Loss: Error del model en les dades d'entrenament; mesura com de bé està aprenent el model.

Validation Loss: Error del model en dades noves (no vistes), que mostra com de bé generalitza.

Accuracy: Percentatge de prediccions correctes fetes pel model, indicant el seu rendiment en la classificació.

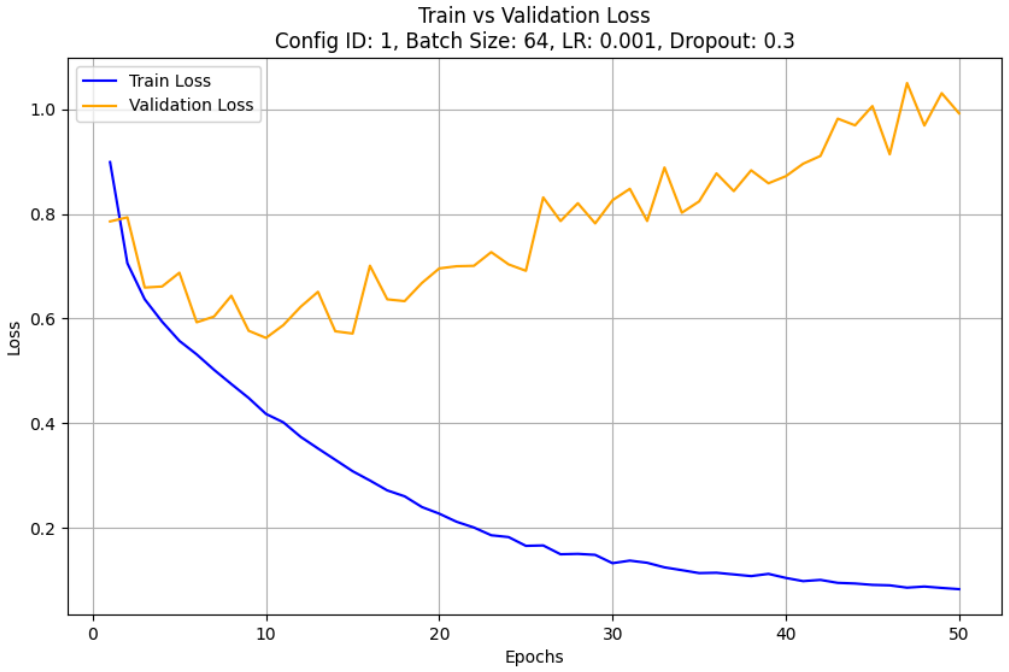
Dropout Rate: Percentatge de neurones desconnectades aleatòriament per prevenir overfitting.

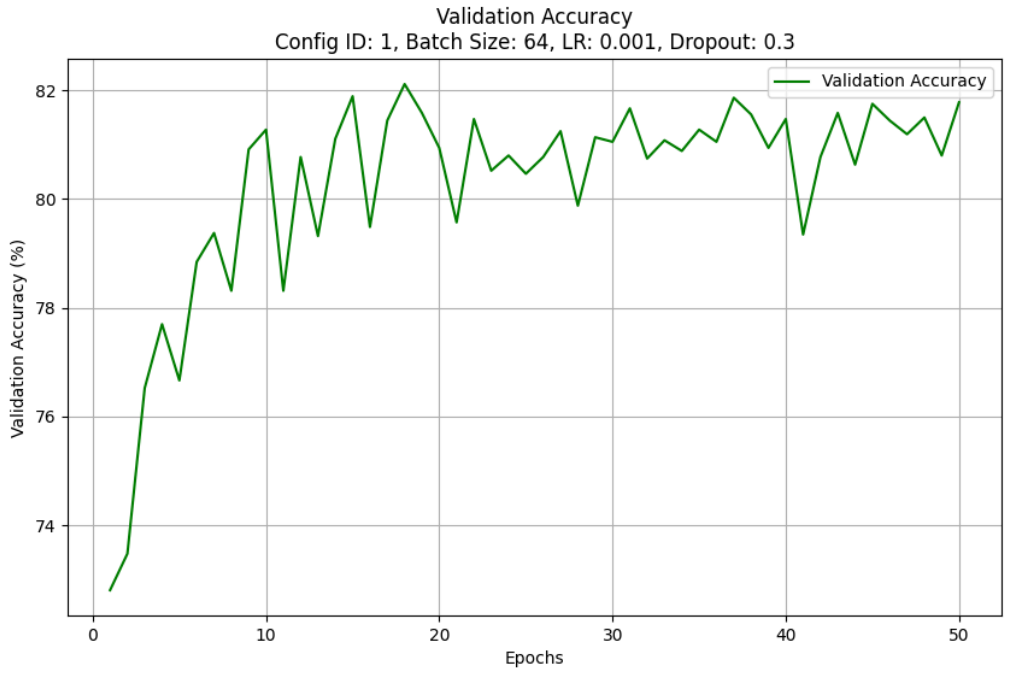
Learning Rate: Ritme d'ajust dels pesos del model per minimitzar l'error.

Les configuracions utilitzades mostren diferents combinacions de **Batch Size**, **Learning Rate (LR)** i **Dropout Rate** per explorar com afecten el rendiment del model d'entrenament en termes de pèrdua (loss) i precisió (accuracy). A continuació, analitzo els resultats i el motiu de les configuracions escollides:

Primerament es mostraran els resultats abans de l'implementació de la regla de l'emoció predominant:

Configuració 1: Batch Size: 64, LR: 0.001, Dropout: 0.3



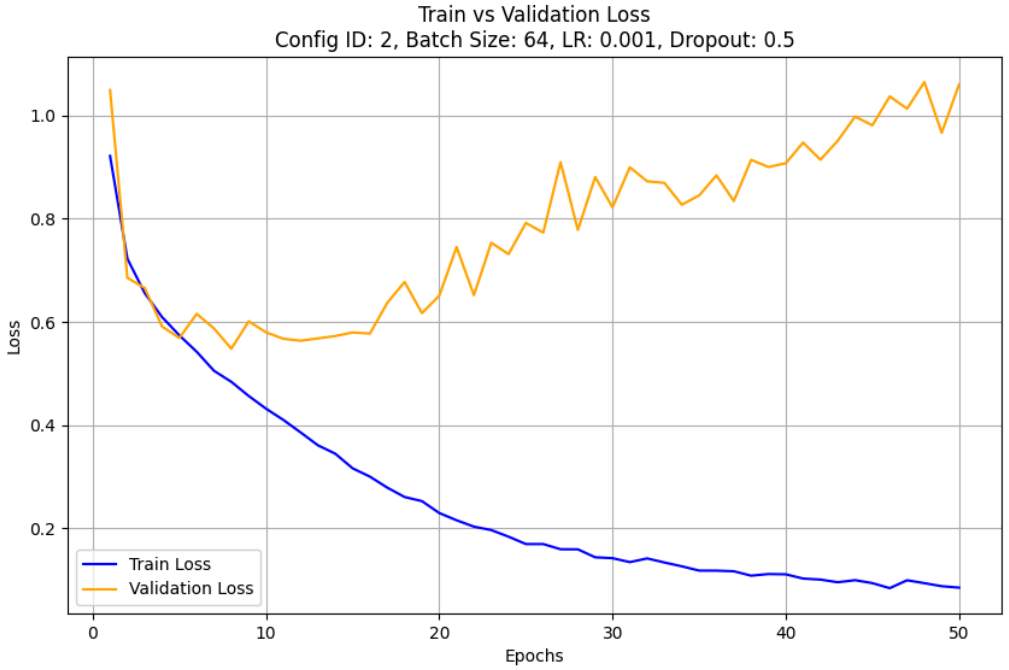


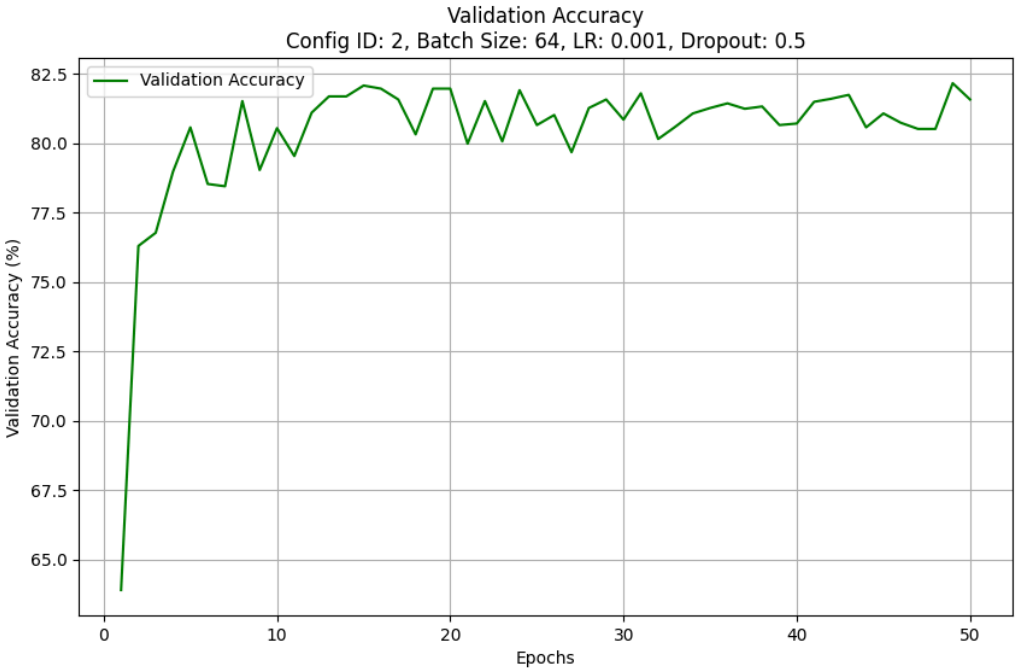
1. **Resultats:**
   - **Train Loss** decreix de manera contínua, indicant que el model està aprenent adequadament sobre les dades d'entrenament.
   - **Validation Loss** comença decreixent però augmenta en les darreres èpoques, suggerint un problema d'overfitting.

- **Validation Accuracy**: Assoleix aproximadament un 82%, mostrant un bon rendiment, però hi ha fluctuacions.

2. **Anàlisi:**
   - L'overfitting és evident en les últimes èpoques. Això pot ser causat per la poca regularització introduïda pel **dropout** (només 0.3).
   - L'augment de la precisió fins al 82% és prometedor, però l'estabilitat hauria de millorar.
3. **Propòsit de la configuració:**
   - Aquesta configuració s'utilitza per establir una línia base amb un **dropout moderat** i un **LR estàndard**.

Configuració 2: Batch Size: 64, LR: 0.001, Dropout: 0.5



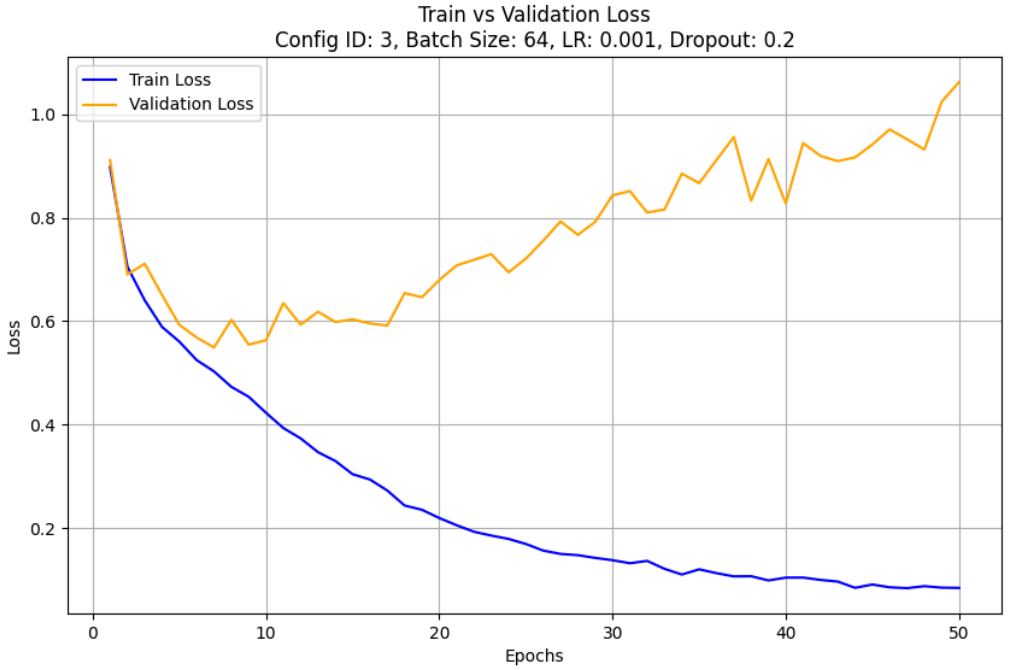


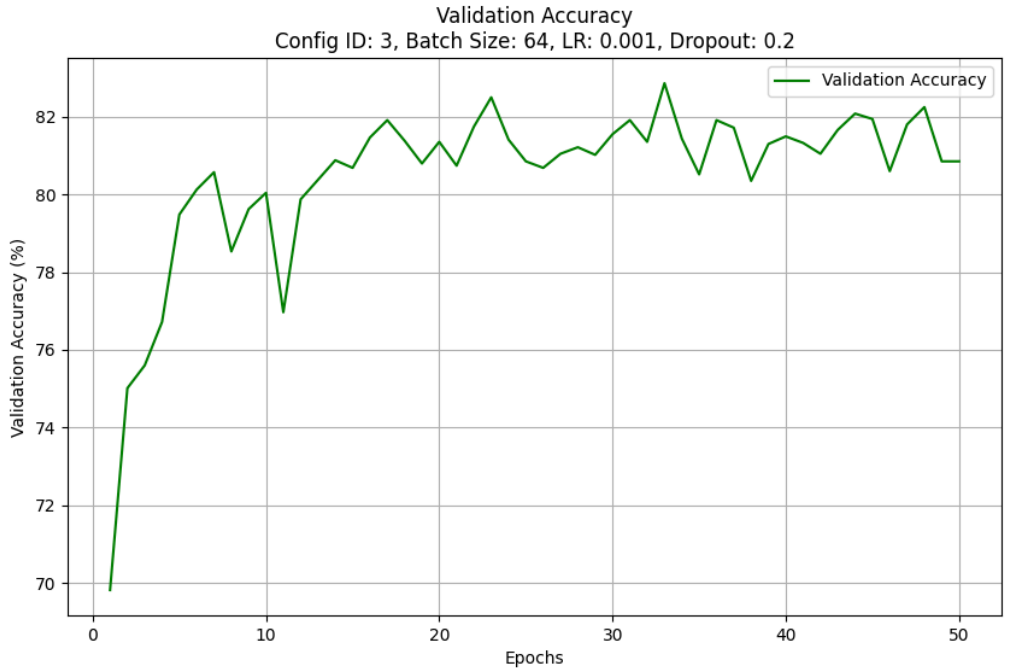
Resultats:

- **Train Loss**: Disminueix de manera més lenta que la configuració anterior, però el model aconsegueix estabilitzar-se millor.
- **Validation Loss** és consistent amb un patró decreixent més uniforme.
- **Validation Accuracy** arriba al voltant del 82.5% i és més estable.

2. **Anàlisi:**
    - Augmentar el **dropout** a 0.5 redueix l'overfitting perquè introdueix més regularització, evitant que el model aprengui patrons massa específics de l'entrenament.
    - La lleugera disminució de l'aprenentatge en les primeres èpoques es compensa amb una millor estabilitat a llarg termini.
3. **Propòsit de la configuració:**
    - Aquesta configuració es va dissenyar per combatre l'overfitting i millorar la generalització.

Configuració 3: **Batch Size: 64, LR: 0.001, Dropout: 0.2**





Resultats:

- **Train Loss**: Decreix molt ràpidament, mostrant que el model aprèn ràpidament.
- **Validation Loss** comença amb una disminució però incrementa cap a les darreres èpoques.
- **Validation Accuracy**: Fluctua entre el 80% i el 82%, mostrant certa inestabilitat.

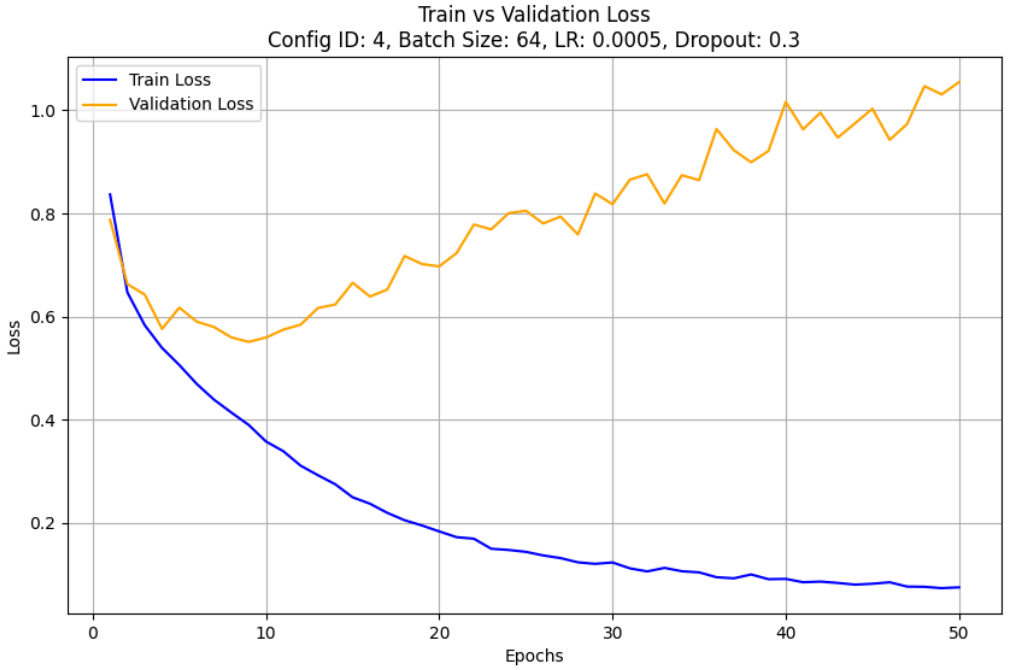
Anàlisi:

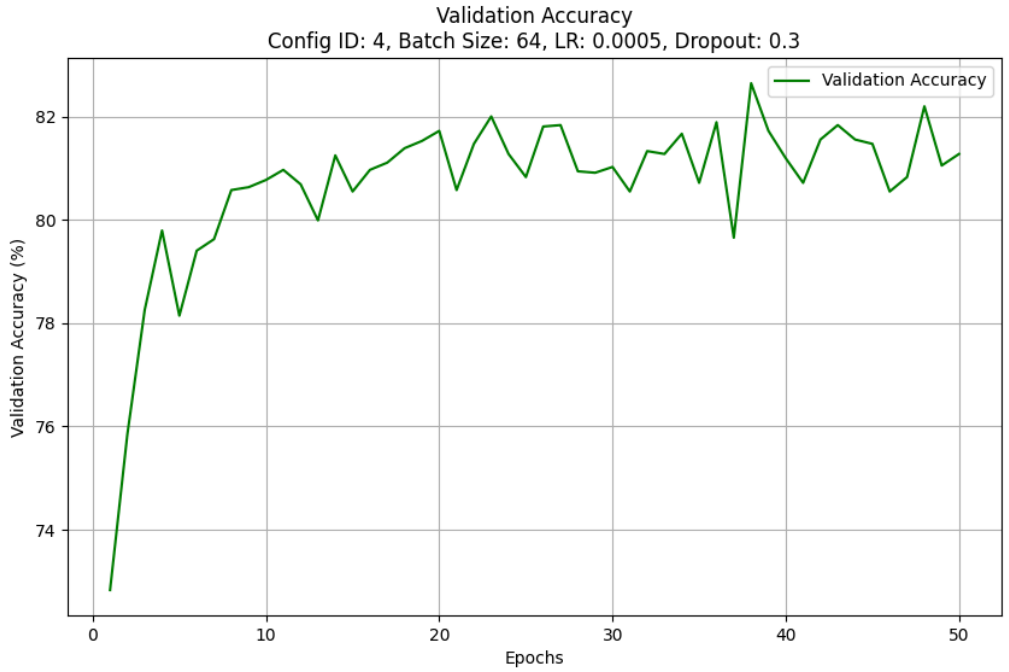
- Un **dropout baix (0.2)** no és suficient per combatre l'overfitting, cosa que es veu reflectida en l'augment de la **Validation Loss**.
- Tot i que el model aprèn ràpidament, no generalitza tan bé en dades de validació.

Propòsit de la configuració:

- Provar si un **dropout més baix** permet un aprenentatge més ràpid amb la penalització d'una possible reducció de generalització.

Configuració 4: **Batch Size: 64, LR: 0.0005, Dropout: 0.3**





Resultats:

- **Train Loss** decreix lentament però de manera molt uniforme.
- **Validation Loss** mostra una tendència ascendent en les últimes èpoques, indicant un lleuger overfitting.
- **Validation Accuracy**: S'estabilitza al voltant del 80-81%.

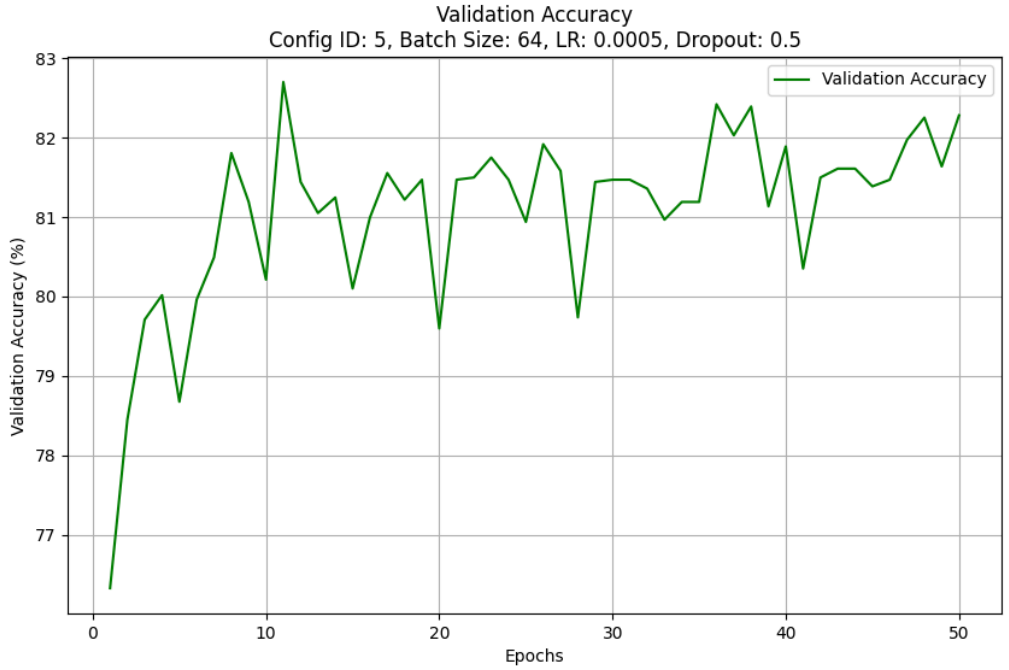
Anàlisi:

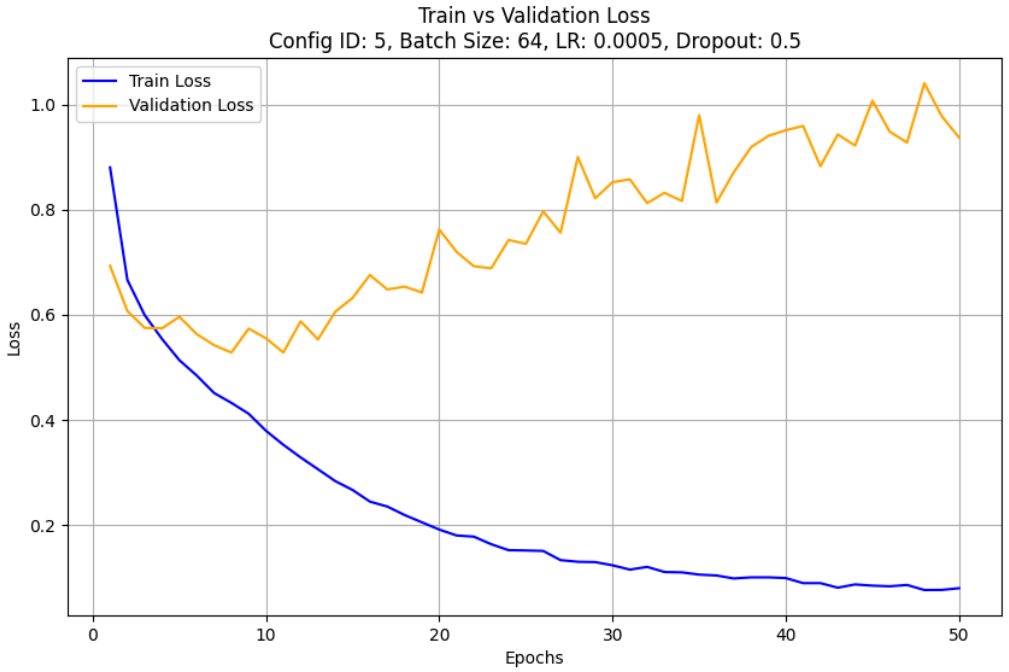
- Un **Learning Rate baix (0.0005)** fa que l'aprenentatge sigui més lent però més consistent, evitant grans fluctuacions.
- El **dropout de 0.3** sembla insuficient per regularitzar completament el model.

Propòsit de la configuració:

- Es va escollir per estudiar l'efecte d'un **Learning Rate baix** en la capacitat del model per generalitzar.

Configuració 5: **Batch Size: 64, LR: 0.0005, Dropout: 0.5**





Resultats:

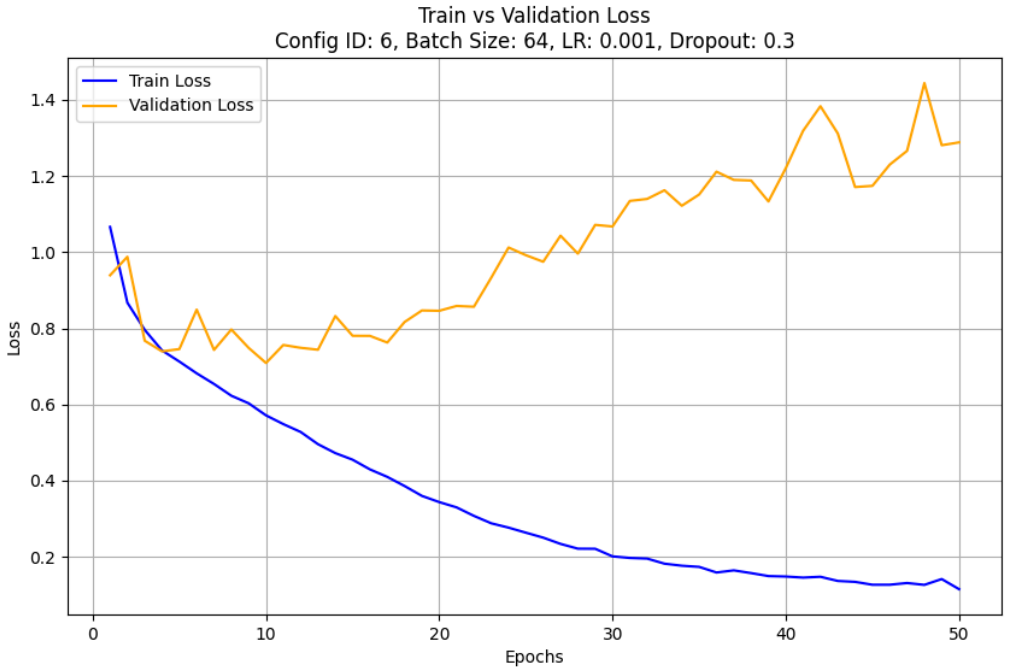
- **Train Loss**: Disminueix de manera molt constant al llarg de totes les èpoques.
- **Validation Loss** es manté estable durant l'entrenament.

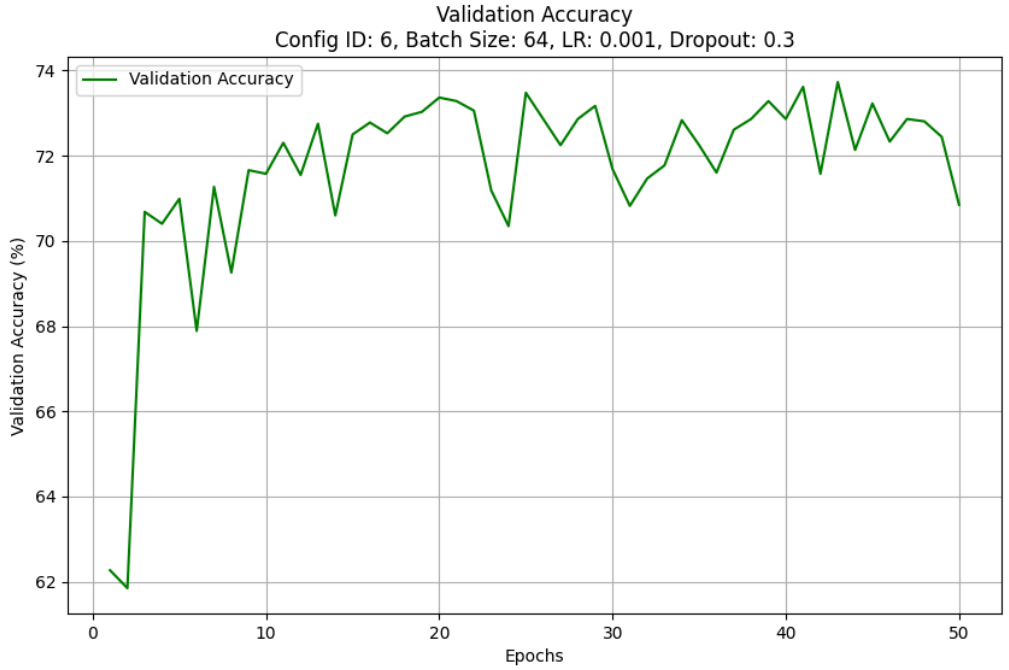
    - **Validation Accuracy** aconsegueix aproximadament un 83%, amb molt poca variabilitat entre èpoques.
2. **Anàlisi:**
    - Aquesta configuració combina un **dropout alt (0.5)** amb un **LR baix (0.0005)**, obtenint els millors resultats de generalització.
    - L'augment de la regularització ajuda a estabilitzar tant la **Validation Loss** com la **Accuracy**, fent que el model sigui més robust.
3. **Propòsit de la configuració:**
    - Optimitzar tant la regularització com la precisió general utilitzant valors conservadors per al **Learning Rate** i el **dropout**.

Conclusió

- Les configuracions es van canviar progressivament per controlar problemes d'**overfitting** i millorar la **generalització**.
- Les configuracions amb **Dropout més alt (0.5)** i **Learning Rate baix (0.0005)** van oferir els millors resultats en estabilitat i precisió.

Configuració 6: Batch Size: 64, LR: 0.001, Dropout: 0.3 (Aplicació Regla de l'Emoció Predominant)





Aquesta configuració (Config ID: 6, Batch Size: 64, Learning Rate: 0.001, Dropout: 0.3) es va realitzar després d'implementar la nova regla de classificació basada en l'emoció predominant. Aquesta regla estableix que l'emoció dominant ha de tenir almenys un 30% més de vots que la segona emoció més votada; en cas contrari, l'etiqueta es considera

"Unknown". Aquesta modificació ha tingut un impacte significatiu en el rendiment del model.

Comparativa amb la Configuració Original Sense la Regla

Validació de la Precisió:

- Amb la regla de l'emoció predominant implementada, l'**accuracy de validació** s'estabilitza al voltant del 72%-74%, amb fluctuacions lleugeres però sense una millora substancial després de les primeres èpoques. Això reflecteix una dificultat afegida per al model a causa de l'augment de la complexitat de la classificació amb més etiquetes "Unknown".
- En canvi, en la primera configuració sense la regla, la precisió arribava a **valors més elevats del 80% o més**, indicant que el model trobava més fàcilment patrons en les etiquetes originals, encara que potser no representaven una classificació tan rigorosa.

Pèrdua de Validació:

- Amb la regla aplicada, la **pèrdua de validació** augmenta significativament després de la primera meitat de les èpoques, arribant a valors superiors a 1.4. Això podria ser un signe d'una major dificultat per al model per ajustar-se a les noves etiquetes.
- Sense la regla, la pèrdua de validació es mantenia més baixa, indicant que el model estava més ajustat a les etiquetes inicials, però possiblement amb menys robustesa en dades no vistes.

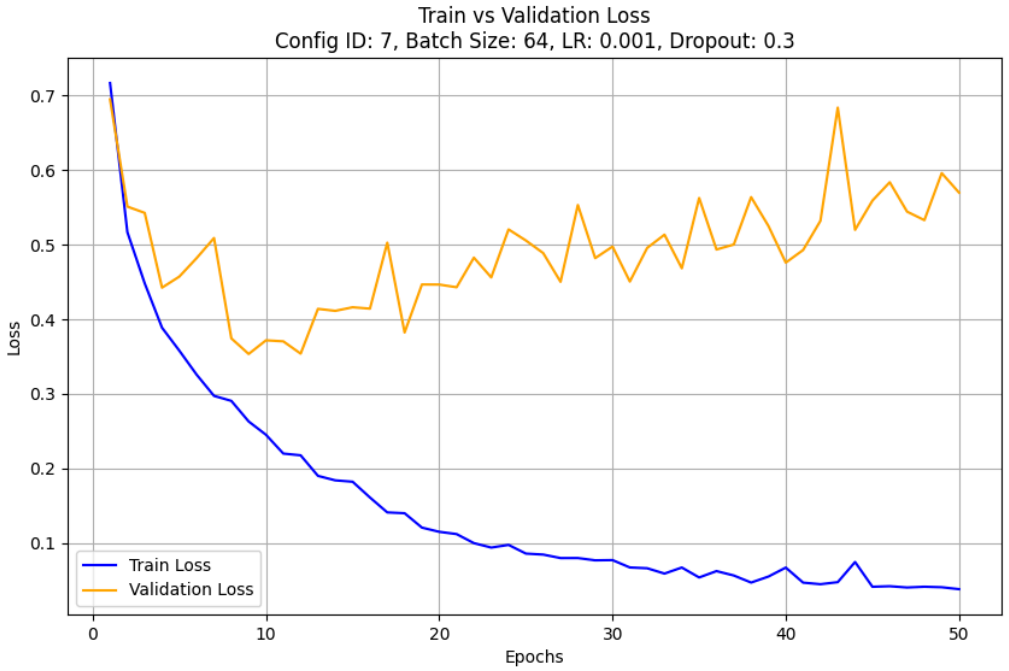
Impacte de la Regla de l'Emoció Predominant:

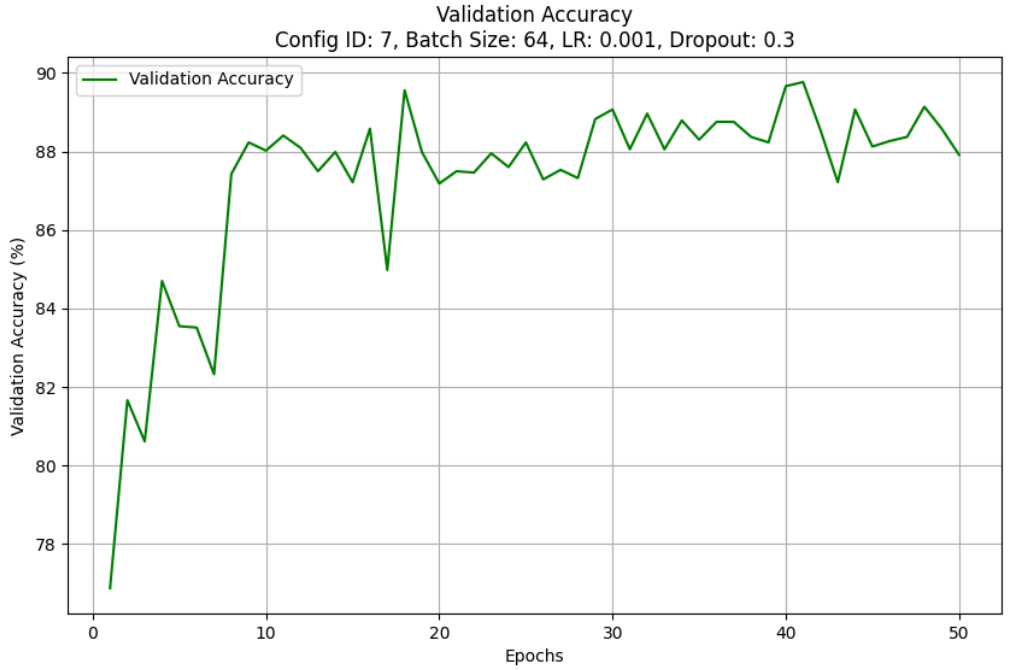
L'aplicació de la regla ha provocat un augment de la quantitat d'etiquetes "Unknown", cosa que ha fet que el model necessiti treballar amb dades més equilibrades però també més complexes. Això redueix la precisió aparent, però podria millorar la generalització en escenaris reals, ja que el model ara és més rigorós en la seva classificació.

Conclusions:

- La nova configuració mostra una precisió més realista i ajustada a l'objectiu de capturar emocions amb més fiabilitat.
- Tot i una lleugera pèrdua en precisió aparent, aquesta configuració representa un pas endavant cap a un model més robust i aplicable en escenaris pràctics.
- Com a següent pas, es podria investigar l'impacte d'augmentar les dades etiquetades amb emocions clares i ajustar hiperparàmetres com el batch size o el learning rate per optimitzar encara més aquest enfocament.

Configuració 7: Batch Size: 64, LR: 0.001, Dropout: 0.3 (Sense images “Unknown”)





Aquesta configuració s'ha realitzat després de descartar les imatges etiquetades com "Unknown", aplicant la regla de l'emoció predominant.
Comparar les configuracions 6 i 7 ens permet analitzar l'impacte de no considerar les imatges etiquetades com "Unknown" a l'hora d'entrenar el model. La configuració 6 incloïa

aquestes imatges, mentre que en la configuració 7 s'han eliminat per centrar l'entrenament en les imatges amb etiquetes clarament assignades a una emoció.

Configuració 6

- **Validation Accuracy**: Es manté entre el 72% i el 74% després de les primeres èpoques, amb una certa variabilitat. Això indica que, tot i que el model és capaç de reconèixer patrons en les dades, l'ús de les imatges "Unknown" pot haver afegit soroll a l'entrenament, limitant-ne la capacitat de generalització.
- **Validation Loss**: Presenta una tendència a augmentar amb el temps, la qual cosa reflecteix un possible problema d'overfitting. Aquest fet es deu, probablement, a la dificultat del model per ajustar-se a les imatges amb etiquetes incertes.

Configuració 7 (Sense imatges "Unknown")

- **Validation Accuracy**: L'absència de les imatges "Unknown" ha permès que la precisió de validació augmenti significativament, arribant a prop del 90% i mantenint-se més estable. Això demostra que les dades amb etiquetes clares milloren la capacitat del model de generalitzar a noves dades.
- **Validation Loss**: Malgrat algunes fluctuacions, la pèrdua de validació és generalment més baixa que en la configuració 6. Aquesta millora indica que el model està aprenent patrons més significatius gràcies a la qualitat més alta de les dades utilitzades en l'entrenament.

Conclusions

L'exclusió de les imatges "Unknown" en la configuració 7 ha tingut un impacte positiu en el rendiment del model:

1. **Augment de la precisió de validació**: El model aconsegueix resultats més consistents i alts, suggerint una millor generalització.
2. **Reducció del soroll**: L'eliminació de les etiquetes incertes ha millorat la qualitat de l'entrenament, permetent al model aprendre patrons més clars i útils.
3. **Millora de la pèrdua de validació**: Les dades més precises han reduït els errors en les prediccions del model.

Aquestes observacions suggereixen que les imatges "Unknown" no aporten valor a l'entrenament del model i que centrar-se en dades amb etiquetes més clares és una estratègia més efectiva per aconseguir resultats robustos.

6 Actualització dels Objectius

L'experiència adquirida durant les fases inicials del projecte, així com l'avanç en les tasques actuals, ha revelat una sèrie d'aspectes que no s'havien considerat inicialment o que han canviat la percepció sobre els objectius establerts. Aquest aprenentatge pràctic ha fet evident la necessitat de revisar i, en alguns casos, redefinir els objectius del projecte per alinear-los millor amb la realitat tècnica i els reptes trobats.

Raons per Replantejar els Objectius

Desviació entre Expectatives i Resultats Reals
Durant les proves inicials amb el dataset FERPlus i la implementació del model ResNet-18, s'han observat limitacions en termes de precisió i temps d'entrenament. Tot i que l'objectiu inicial era assolir una precisió del 75%, la realitat ha mostrat que aquest objectiu pot ser massa ambiciós sense implementar optimitzacions addicionals, com l'ús de ResNet-50 o tècniques més avançades de preprocessament de dades.
Impacte del Canvi de Dataset
Inicialment es va considerar l'ús d'un conjunt de dades basat en AffectNet, però l'adopció final de FERPlus ha canviat els requisits del projecte. FERPlus, tot i ser de gran qualitat, presenta menys dades i una distribució més desbalancejada, fet que ha obligat a desenvolupar tècniques específiques per manejar aquestes diferències. Aquest canvi requereix ajustar objectius relacionats amb la generalització i la robustesa del model.
Reducció del Temps d'Entrenament
L'ús inicial de la CPU i la seva substitució per una GPU ha suposat una millora dràstica en els temps d'entrenament (de 20-30 minuts per època a només 2 minuts). Aquesta millora permet ara explorar configuracions més complexes, augmentar el nombre d'èpoques i fer proves més exhaustives. Això implica reorientar els objectius cap a l'explotació d'aquesta capacitat millorada.
Nous Reptes Tècnics Identificats
La implementació de tècniques com el Dropout, l'augmentació de dades i la prevenció de l'overfitting han demostrat ser essencials per millorar el model. Aquests avenços han fet evident la necessitat de donar prioritat a objectius més específics, com la reducció de l'impacte de l'overfitting i la millora de la precisió en categories minoritàries.
Impacte de l'Experiència Pràctica
L'experiència adquirida fins ara ha proporcionat una comprensió més profunda dels requisits tècnics i les limitacions pràctiques. Això ha portat a considerar nous objectius, com la comparació exhaustiva entre ResNet-18 i ResNet-50, o la necessitat de dissenyar una interfície d'usuari més interpretativa per fer el sistema més accessible.

ID	Nom	Objectiu	Prioritat
01	**Alta precisió del model**	Assolir una precisió mínima del 75% en la classificació de les emocions utilitzant el dataset FERPlus. Això garantirà un model robust i fiable per a aplicacions en temps real.	Essencial
02	**Reducció del temps d'entrenament**	Reduir el temps d'entrenament a menys de 2 minuts per època, utilitzant optimitzacions com l'ús de GPU i ajustos en els hiperparàmetres.	Essencial
03	**Generalització del model**	Aconseguir que el model mostri una alta capacitat de generalització amb dades no vistes, assegurant que l'error de validació sigui inferior al 5%.	Essencial
04	**Robustesa en múltiples emocions**	Classificar amb precisió les 8 emocions bàsiques del dataset FERPlus, incloent-hi emocions més complexes com "Contempt" i categories com "Unknown".	Essencial
05	**Comparació d'arquitectures**	Identificar si ResNet-18 o ResNet-50 proporciona millor rendiment en termes de balanç entre precisió i temps d'entrenament, seleccionant l'arquitectura més òptima.	Essencial
06	**Consistència en dades desbalancejades**	Minimitzar l'impacte del desbalanceig en categories minoritàries com "Fear" o "Disgust", garantint una F1-score superior al 70% per a totes les classes emocionals.	Secundari
07	**Temps de resposta en temps real**	Garantir que el sistema pugui processar i predir emocions en menys de 200ms per fotograma en entorns en temps real.	Secundari

O8	**Visualització interpretativa del model**	Desenvolupar gràfics i interfícies que mostrin la probabilitat associada a cada emoció, facilitant l'interpretació dels resultats obtinguts pel model.	Secundari
O9	**Reducció de l'overfitting**	Implementar tècniques per reduir l'overfitting, com el Dropout i augmentació de dades, amb l'objectiu de mantenir una diferència mínima entre els errors d'entrenament i validació.	Exploratori
O10	**Ampliació de l'ús del model**	Desenvolupar un model que pugui ser utilitzat en aplicacions pràctiques, com ara la detecció d'emocions en temps real per a l'atenció al client o sistemes educatius.	Exploratori

7 Referències

PyTorch
A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024-8035.
Available: https://pytorch.org/

Torchvision
The PyTorch team, *Torchvision: A Library for Computer Vision Research and Applications*,
Available: https://pytorch.org/vision/stable/.

OpenCV
G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
Available: https://opencv.org/

Pandas
W. McKinney, "Data Structures for Statistical Computing in Python," *Proceedings of the 9th Python in Science Conference*, 2010, pp. 56-61.
Available: https://pandas.pydata.org/

Matplotlib
J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, 2007.
Available: https://matplotlib.org/

TQDM
Noam Raphael, "TQDM: A Fast, Extensible Progress Meter for Python and CLI," 2016.
Available: https://tqdm.github.io/.

AffectNet Dataset
A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 18-31, Jan. 2019.
Available: http://mohammadmahoor.com/affectnet/

FERPlus Dataset
T. Simonyan et al., "FER2013 Dataset Enhancement: FERPlus," *IEEE Transactions on Affective Computing*. Disponible a FERPlus
Available: https://github.com/microsoft/FERPlus