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.
- **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.
- **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.
- **Pandas i Matplotlib**: Per a la manipulació de dades i visualització de resultats durant el procés de desenvolupament i avaluació del model.
- **TQDM**: Per monitoritzar el progrés durant l'entrenament del model, la qual cosa és especialment útil per a bucles llargs.

3 Estructura General del Projecte

El projecte està estructurat en diferents arxius i carpetes per organitzar el flux de dades i el codi de manera eficient:

1. `normalize.py`

Aquest arxiu conté la classe `EmotionDataset`, que s'encarrega de preparar les imatges perquè el model les pugui utilitzar de manera òptima. Els passos que realitza són els següents:

1. **Càrrega d'imatges**: La classe `EmotionDataset` carrega les imatges des d'una carpeta especificada. S'utilitza la biblioteca `PIL` (Python Imaging Library) per obrir i manipular aquestes imatges.
2. **Transformacions i Preprocesament**:
    - **Escalat**: Les imatges es redimensionen a 224x224 píxels, la mida d'entrada estàndard per al model ResNet-18.
    - **Conversió a tensors**: Les imatges es converteixen a tensors (estructura de dades numèrica que la xarxa pot processar) perquè PyTorch les pugui gestionar.
    - **Normalització**: S'aplica una normalització de colors amb `torchvision.transforms` per ajustar els valors de color als paràmetres utilitzats en el model ResNet-18. Aquesta normalització centra els valors en un rang que millora la convergència durant l'entrenament.
3. **Visualització de lots d'imatges**: S'inclou una funció que permet visualitzar un lot d'imatges processades, cosa útil per a comprovar que les imatges es carreguen i es processen correctament. Això ajuda a detectar errors durant les proves i a assegurar que les transformacions s'apliquen de manera adequada abans d'entrenar el model.

2. `model.py`

Aquest arxiu defineix l'arquitectura de la xarxa neuronal convolucional (CNN) i gestiona l'entrenament del model. Els passos són els següents:

1. **Definició de l'arquitectura de la CNN**:
    - Es carrega un model `ResNet-18` preentrenat de `torchvision.models`, que proporciona una arquitectura robusta amb capacitat per reconèixer patrons visuals.
    - **Modificació de la capa final**: La capa de sortida del model es modifica per ajustar-se a la classificació en 8 classes (les emocions que es volen reconèixer), canviant el nombre de neurones de la capa final.
2. **Configuració dels hiperparàmetres**:

- **Funció de pèrdua**: Es defineix `CrossEntropyLoss` com a funció de pèrdua, una funció comuna per a tasques de classificació que compara les prediccions del model amb les etiquetes reals.
- **Optimitzador**: S'utilitza l'optimitzador `Adam` amb un `learning rate` inicial de 0.001. Aquest optimitzador ajusta els pesos del model per minimitzar la pèrdua de manera eficient.

3. **Compatibilitat amb GPU**: Es configura el codi per utilitzar la GPU, si està disponible, cosa que accelera significativament l'entrenament. El model i les dades es traslladen a la GPU automàticament quan s'inicialitza l'entrenament.
4. **Entrenament i Validació**: El codi inclou un bucle d'entrenament i de validació. Durant l'entrenament, el model ajusta els seus pesos en cada època per minimitzar la pèrdua d'entrenament. Durant la validació, es mesura la pèrdua de validació i la precisió en un conjunt de dades no vist per comprovar la capacitat de generalització del model.

3. Carpeta `data`

La carpeta `data` emmagatzema el conjunt de dades en tres subconjunts:

- **Entrenament**: Conté les imatges que el model fa servir per aprendre els patrons emocionals.
- **Validació**: S'utilitza per avaluar el model durant l'entrenament sense que aquestes dades influeixin en els pesos del model.
- **Prova**: Conjunt de dades final per avaluar la capacitat de generalització del model un cop completat l'entrenament.

Conjunt de dades: AffectNet

El dataset escollit per a aquest projecte és **AffectNet**, un conjunt de dades extensiu per al reconeixement d'emocions facials. AffectNet inclou més d'1 milió d'imatges obtingudes de la web, amb més de 400.000 imatges anotades manualment amb diverses emocions (alegria, tristesa, sorpresa, enuig, por, disgust, emoció neutra, entre d'altres). Els avantatges d'AffectNet són:

- **Gran volum de dades**: Amb un gran nombre d'imatges anotades, AffectNet proporciona al model una gran varietat d'exemples, cosa que ajuda a capturar amb precisió les emocions facials.
- **Diversitat cultural i demogràfica**: El conjunt inclou persones de diferents edats, contextos culturals i nivells d'intensitat emocional, fet que permet al model aprendre a reconèixer emocions en una varietat de rostres.
- **Complexitat emocional**: La varietat d'intensitats en les expressions i el context temporal ajuda a capturar la complexitat de les emocions humanes i millora la capacitat del model per a aplicacions en el món real.

4 Progressió Setmanal del Desenvolupament

Lliurament	10/11/2024		*Informe Progrés 1*
4	07/10-10/10	4 dies	Selecció del dataset
4	11/10-13/10	3 dies	Normalització d'imatges
5	14/10-20/10	7 dies	Preparació dels conjunts de dades (validació, entrenament, prova)
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
Tot i haver seguit el cronograma, aquesta tasca encara no s'ha completat del tot, ja que es volen implementar ajustos per crear un model més potent i robust, capaç de generalitzar millor i aconseguir una precisió més alta.			

5 Rendiment

Durant el desenvolupament, es van realitzar proves de rendiment del model en el conjunt de validació per mesurar la precisió i la pèrdua en cada època d'entrenament. A continuació, es presenten les mètriques recollides en 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.

Max Images: Nombre màxim d'imatges usades de cada subcarpeta per entrenar el model, útil per reduir el temps d'entrenament.

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

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

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

Èpoques: 10 **Batch_Size**: 32 **Max_images**: ALL DATASET **Temps:** 5h aprox.			
Època	**Pèrdua d'Entrenament**	**Pèrdua de Validació**	**Precisió (%)**
1	1.4791	1.3618	49.00%
2	1.2738	1.3398	51.25%
3	1.1753	1.2713	54.75%
4	1.0717	1.2966	54.50%
5	0.9456	1.3368	54.38%
6	0.7831	1.4797	55.25%
7	0.5759	1.7577	51.62%
8	0.4012	1.8710	51.75%
9	0.2909	2.2184	51.75%
10	0.2259	2.4764	51.75%

Observem que la **pèrdua d'entrenament** disminueix de manera consistent, arribant a un valor molt baix (0.2259) a l'última època. Això indica que el model està aprenent molt bé els patrons en les dades d'entrenament.

La **pèrdua de validació**, en canvi, comença a augmentar significativament després de la quarta època, arribant a un valor molt alt (2.4764) a l'última època, mentre que la **precisió de validació** es manté pràcticament estable al voltant del 51.75% després de la tercera època.

Aquest patró és característic de l'overfitting, on el model s'ajusta excessivament a les dades d'entrenament, perdent la seva capacitat de generalitzar en el conjunt de validació. És especialment comú en conjunts de dades grans i diversos com el que s'utilitza aquí amb més de 400000 images.

Per intentar combatre aquest overfiting es va implementar un Max_images que permet capar el nombre màxim d'imatges de cada subcarpeta, és a dir, de cada emoció. També es va ajustar el nombre de èpoques. Veient que a partir de la 7 època la precisió baixava es va limitar a 6 per reduir el temps. El batch_size es va augmentar 64 per processar més imatges per iteració i intentar reduir encara més el temps.

Èpoques: 6 **Batch_Size**: 64 **Max_images**: 2000

Temps: 1h 12min aprox.

Època	Pèrdua d'Entrenament	Pèrdua de Validació	Precisió (%)
1	1.5438	1.5574	41.62%
2	1.2846	1.3822	49.50%
3	1.1479	1.4410	50.12%
4	0.9992	1.5981	47.62%
5	0.8491	1.5202	50.38%
6	0.6478	1.7433	48.00%

La **pèrdua d'entrenament** disminueix gradualment, però la **pèrdua de validació** no segueix una millora clara i, en lloc d'això, augmenta al voltant de l'època 4. La **precisió** oscil·la al voltant del 50%, però sense un augment substancial.

Aquest resultat és consistent amb un entrenament menys intensiu (en reduir el nombre d'imatges utilitzades), però, tot i així, es mostra una tendència a l'overfitting, probablement perquè les dades no són suficients per permetre al model generalitzar-se en altres exemples.

El temps es va reduir considerablement, però la precisió va baixar com a conseqüència. El canvi del Batch_Size no va afectar, ja que, es processaven cada iteració el doble d'imatges però cada iteració tardava el doble en ser processada.

Èpoques: 6 **Batch_Size**: 64 **Max_images**: 2500

Temps: 1h 40min aprox.

Època	Pèrdua d'Entrenament	Pèrdua de Validació	Precisió (%)
1	1.5025	1.5484,	42.62%
2	1.2631	1.3867,	49.38%
3	1.1450	1.3701,	50.88%

4	1.0146	1.4153,	50.88%
5	0.8596	1.4322,	53.00%
6	0.6933	1.6741,	51.88%

Aquesta configuració mostra un patró similar a la segona configuració, amb una disminució inicial de la pèrdua d'entrenament, però la **pèrdua de validació** comença a augmentar després de la tercera època. La **precisió** es manté en el rang del 50-53%, amb poc increment significatiu.

Tot i que utilitza una mica més de dades que la segona configuració, la capacitat de generalització del model segueix sent limitada.

Resultats:

En resum, els resultats mostren que el model actual té dificultats per generalitzar en el conjunt de validació, degut a un overfitting clar. Les millores recomanades inclouen ajustar els hiperparàmetres, provar amb ResNet-50 amb tècniques de regularització i explorar la possibilitat d'incorporar dades de vídeo amb el dataset SAVEE per capturar informació temporal. Aquests canvis podrien ajudar a millorar la precisió i robustesa del model.

6 Estat del Projecte

Segons el cronograma establert, el projecte ha avançat en diverses fases clau. Ja s'han completat les tasques de **Disseny i implementació de l'arquitectura de la CNN**, incloent-hi la **definició de les capes** i la **configuració dels hiperparàmetres**.

En aquest punt, estem considerant afegir un nou conjunt de dades, **SAVEE** (Surrey Audio-Visual Expressed Emotion), que inclou seqüències de vídeo. La integració de dades de vídeo, a més de les imatges actuals, permetrà que el model capti millor les transicions i expressions emocionals en context temporal, fet que pot millorar la seva precisió i robustesa en aplicacions en temps real. Així mateix, estem valorant l'ús de l'arquitectura **ResNet-50**, amb més capacitat que ResNet-18, per tal de capturar millor els detalls emocionals.

Aquests ajustos podrien suposar una lleugera demora en el cronograma previst, ja que l'entrenament amb ResNet-50 i un conjunt de dades més extens requerirà més temps i recursos. Tot i així, creiem que aquests canvis contribuiran a obtenir un model més potent i capaç de generalitzar millor.

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: https://www.kaggle.com/datasets/thienkhonghoc/affectnet?resource=download

SAVEE Dataset
P. Jackson, S. Haq, J. C. Moore, G. Lan, and A. Battersby, "Surrey Audio-Visual Expressed Emotion (SAVEE) Database," *University of Surrey*, Guildford, UK, 2014.
Available: http://kahlan.eps.surrey.ac.uk/savee/.