UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Facial Emotion Recognition Using Deep-learning Models

Andrés Sebastián Espinel Reyna

Ingeniería en Ciencias De la Computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero en ciencias de la computación

Quito, 11 de diciembre de 2020

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Facial Emotion Recognition Using Deep-learning Models

Andrés Sebastián Espinel Reyna

Nombre del profesor, Título académico **Noel Pérez, Phd**

Quito, 11 de Diciembre de 2020

© DERECHOS DE AUTOR

Nombres y apellidos: Andrés Sebastián Espinel Reyna

Código: 00127763

Cédula de identidad: 1723790125

Lugar y fecha: Quito, 11 de diciembre de 2020

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

Este trabajo propone un método de reconocimiento de gestos faciales basado en arquitecturas de aprendizaje profundo denominadas DCNN1, DCNN2, DCNN3, DCNN4 y DCNN + Autoencoder, que maximizan el rendimiento de clasificación en conjuntos de datos únicos y mixtos. Validamos las arquitecturas propuestas en tres bases de datos diferentes: Jaffe, CK + y la combinación de ambas bases de datos Jaffe y CK + en una estrategia de validación cruzada de cinco veces. Las puntuaciones medias de precisión obtenidas del 95%, 94% y 96% por los modelos DCNN4, DCNN2 y DCNN + Autoencoder se mostraron como el mejor rendimiento para las bases de datos Jaffe, CK + y Jaffe & CK +, respectivamente. Además, de acuerdo con la función de pérdida de entropía cruzada, los modelos seleccionados no incurrieron en sobreajuste.

Palabras clave: detección de rostros, clasificación de gestos faciales, modelos de aprendizaje profundo, inteligencia artificial, imágenes faciales.

ABSTRACT

This work proposes a face gesture recognition method based on deep learning architectures named DCNN1, DCNN2, DCNN3, DCNN4, and DCNN+Autoencoder, that maximize the classification performance on single and mixing datasets. We validated the proposed architectures on three different databases: Jaffe, CK+, and the combination of both databases Jaffe & CK+ over a five-fold cross-validation strategy. The obtained mean accuracy scores of 95%, 94%, and 96% by the DCNN4, DCNN2, and DCNN+Autoencoder models were raised as the best performance for the Jaffe, CK+, and Jaffe & CK+ databases, respectively. Moreover, according to the cross-entropy loss function, the selected models did not incur overfitting.

Keywords: face detection, face gesture classification, deep-learning models, artificial intelligence, face images.

Tabla de contenido

INTRODUCTION **10**

MATERIALS AND METHODS **13**

Face Gesture Databases 13
Jaffe 13
CK+ 13
Deep-Learning Models 14
Proposed Model 15
Experimental Setup 15
Image Preprocessing 17
Test and Training Partitions 18
Models Configuration 18
Assessment Metrics 18
Selection Criteria 19

RESULTS AND DISCUSSION **20**

CONCLUSIONS AND FUTURE WORK **23**

ACKNOWLEDGEMENT **24**

REFERENCES **25**

ÍNDICE DE TABLAS

Table #1: Core architecture of proposed deep models……17

Table #2: Performance results of deep learning Models on the three databases.……20

ÍNDICE DE FIGURAS

Figure #1 Examples of Jaffe (top row) and CK+ (bottom row datasets……...14

Figure # 2 WorkFlow of DCNN3 (top row) and DCNN+Autoencoder (bottom row).16

Figure # 3 Performance of the best models for the Jaffe (left), CK+ (center) and Jaffe & CK+(right) datasets……...18

INTRODUCTION

The recognition of human gestures is a sub-branch of computer vision that uses biometric devices such as cameras to capture human gestures to be interpreted by algorithms and thus recognize emotions or movement patterns. Gesture recognition has helped, for example, with monitoring of medical patients, control in virtual games, navigation of virtual environments, forensics research, body language interpretation, among others (Gordillo, 2015).

In the area of neurology, studies on facial expressions are very important to analyze people behaviour. For example, the study by Gordillo F, Pérez MA, Arana JM, Mestas L, López RM revealed through mathematical algorithms that facial expressions help interaction between people (Gordillo,2017). Situations such as abuse, war or stress generate expressions of anger or sadness. Likewise, positive experiences provide joy expressions. La Universidad de la plata's researched used depth and RGB cameras to capture gestures and signs of faces, creating a database with 3200 videos and 64 different face gestures. This approach used Markovo and feed-forward back-propagation artificial neural networks models to classify facial signs based on the position and gestures of the faces (Ronchetti,2016). Similarly, the research of Léon J. M. Rothkrantz classify facial gestures whenever the subject was talking or not. They also carried out an experimental study on facial muscle actions typical for speech articulation. The later was based by applying a HSV (Hue, Saturation, Value) color-based segmentation of the face. Then, identifying the zones of interest, which were analyzed by a trained Jordan Recurrent Neural Network (JRNN).

Convolutional neural networks (CNN) based models have gained popularity in computer vision to face image labeling problems. In this regard, several models have adapted CNNs for facial emotion recognition (FER) alongside data augmentation and data preprocessing techniques. For example, OpenCV, a highly optimized computer vision and machine learning software library (OpenCV team, 2020), has used a deep CNN for face recognition and emotion classification. Researchers have used OpenCV to extract features and for image classification achieving an accuracy of 98 % in the Jaffe dataset (Veena & others, 2016).Likewise, OpenCV has been used to get bounding boxes around each face of the databases and then have humans doing the cropping and all the corrections necessary to feed the CNNs models for feature extraction and classification obtaining an accuracy of 65% on the FER2013 dataset.(Goodfellow & others,2015) Other approach was using OpenCV in conjunction with a 2-channel deep CNN for processing raw images and LBP (Local Binary Pattern) maps, this model achieved an accuracy of 96% on an asian dataset (Jing & others ,2020).

There are other CNNs based extensions such as WDCNN (Wide First-layer Kernels) and WMCNN-LSTM (Long short-term memory), which are implemented in two different networks for training in combination with two partial VGG16 networks for classification. These models obtained an accuracy of 88% on a Caucasian dataset (Hepeng, Bin, & Guohui, 2019). Another interesting approach was decontaminating the images using the IR (infrared) spectrum. These images were feed into a partial VGG DCNN and to a shallow CNN for classification. This model used a confusion matrix with an overall ACC of 92% on a Caucasian dataset. In addition, a Discriminative Deep multi-task learning CNN (DDMTL) has been used in conjunction with a k-nearest neighbor (kNN) model and an optimization module based on softmax and contrastive

loss functions. This architecture reached an overall accuracy of 67% and 55% on combinations of Caucasian and Asian datasets (Hao, 2020). On the other hand, a deep CNN was used as a robust image selection with a CNN model for emotion classification; this architecture obtained an accuracy of 59 when combining information of two Caucasian datasets (Huadong and Hua, 2020). Moreover, a DCNN with residual blocks were implemented to reach an accuracy score of 93% on a dataset of Asian faces (Kumar, Pourya and Paramjit, 2019). In (Nacer and Mahdi, 2018), it was created a category-based support vector machine (SVM) model using two or more samples of different classes or expressions. From each sample, the HOG (histogram of oriented gradients) and LBP features were computed to encode the faces. In this feature space the SVM classifier reached a maximum accuracy score of 97% on a Caucasian dataset. Finally, the combination of different micro-action-pattern modules with a deep CNN was used to generate more abstract mid-level semantics. The architecture combined two different datasets and achieved accuracy scores of 72.2%, 29.43%, 93.46%, and 25% on different combinations of Caucasians and Asian datasets respectively (Mengyi, Shaoxin, Shiguang and Xilin,2015).

Despite the evolution and development in FER, most of the previous approaches focused on facial recognition on single datasets. Only a few of these works were used on mixing datasets, and the obtained results were not successful at all. Therefore, we proposed a face gesture recognition method based on deep learning architectures that maximize the classification performance on single and mixing datasets in this work.

MATERIALS AND METHODS

Face gesture databases

We considered three publicly databases with samples of different ethnicities, poses, sizes and lighting conditions. A brief description of each database is next:

Jaffe

Jaffe stands for Japanese Female Facial Expression, is a face database consisting of 213 images of 7 facial expressions posed by 10 Japanese women. These expressions are labeled by the word: happy, sad, contempt, surprise, fear, anger, and neutral.

All images are 256 x 256 on gray level, in TIFF format with no compression applied. Jaffe is an open access database, published in 1998 by Michael, J. Lyons et.al. Some samples of this database are shown in Fig 1 first row.

CK+

CK+ is the second version on the Cohn-Kanade database (CK). However, this database includes posed and spontaneous expressions. It was applied to 210 subjects or posers between 18 to 50 years of age from different races (Kanade, Chon & Tian, 2000). For the posed facial expressions, 123 subjects performed 593 sequences. A sequence is defined as the transition of a neutral expression to a peak expression, the peak expression is coded with one emotion label (same as in the Jaffe database). On the other hand, the spontaneous expressions are counted from 66 subjects who smiled to the camera between takes of posed expressions.

Images are either 640 x 490, or 640 x 480 pixels arrays of 8-bit gray level or 24-bit color values (Lucey, et.al, 2010). Some samples of this database are shown in Fig. 1 second row.

Fig. 1. An example of facial gestures, from left to right: anger, disgust, fear, happiness, neutral, sadness, surprise of *Jaffe* (top row) and *CK+* (bottom row) databases.

Deep-Learning Models

Deep learning is composed by a group of algorithms that learn from data. These algorithms are based on the structure and function of the brain's neural networks (Deep Lizard, 2020). These models are very popular when dealing with the classifcation problem.

The simplest deep learning model is called a feedforward neural network. A fully-connected architecture, also called deep feed-forward networks, aims to feed multilayer peceptrons to approximate the output of a given function like the sigmoid function (Gupta,2017). However, fully-connected networks are prone to commit data over-fitting.

Deep CNNs are one of the many deep-learning models that are used for image visualization and classification. These networks are regularized versions of a multilayered fully-connected network. Deep CNNs apply hierarchical patterns of data to assemble a more complex pattern using smaller samples, rather than the traditional approach used by a fully-connected network, that is based in the magnitude measurement of weights to the loss function (Deep Lizard,2020). It consists of multiple convolution layers defined by different kernel sizes with an activation function that connects the next layer. Pooling layers and dropout layers (to avoid overfitting) can also be found. At the end of every

CNN or DCNN there is at least one fully-connected layer with an output that matches with the number of classes to be classified. The main goal of a CNN is to reduce the special variance and extract the important features in an image for classification.

An autoencoder can be implemented alongside a DCNN architecture. The main goal of an autoencoder is to compress data from an input layer (coder), and then uncompress the output into a close approximation of the original input (DeepAI, 2014). This process reduces the spatial variance and reduces the noise in an image.

Proposed Model

Five different deep CNN models were implemented with one of them using an autoencoder scheme. The DCNN1, DCNN2, DCNN3, and DCNN4 are based on the deep CNN architecture. The DCNN+Autoencoder is based on the combination of a deep CNN architecture and a deep feed-forward network on an autoencoder topology. These models were empirically developed to explore their strengths in the facial emotion classification.

The proposed models are inspired in the VGG16 architecture (Simonyan, 2015), which won the Imagenet competition in 2014. The VGG16 model uses kernels of 3x3 sizes and max-pooling layers with sizes 2x2 and a stride of 2 units. Then, there are two fully-connected layers with a softmax activation function. The proposed models are an extension of the VGG16 architecture, including variation of the kernel sizes in different layers to 5x5 and 3x3. Also, the number of convolution layers was reduced from 16 to 3 and 4 layers.

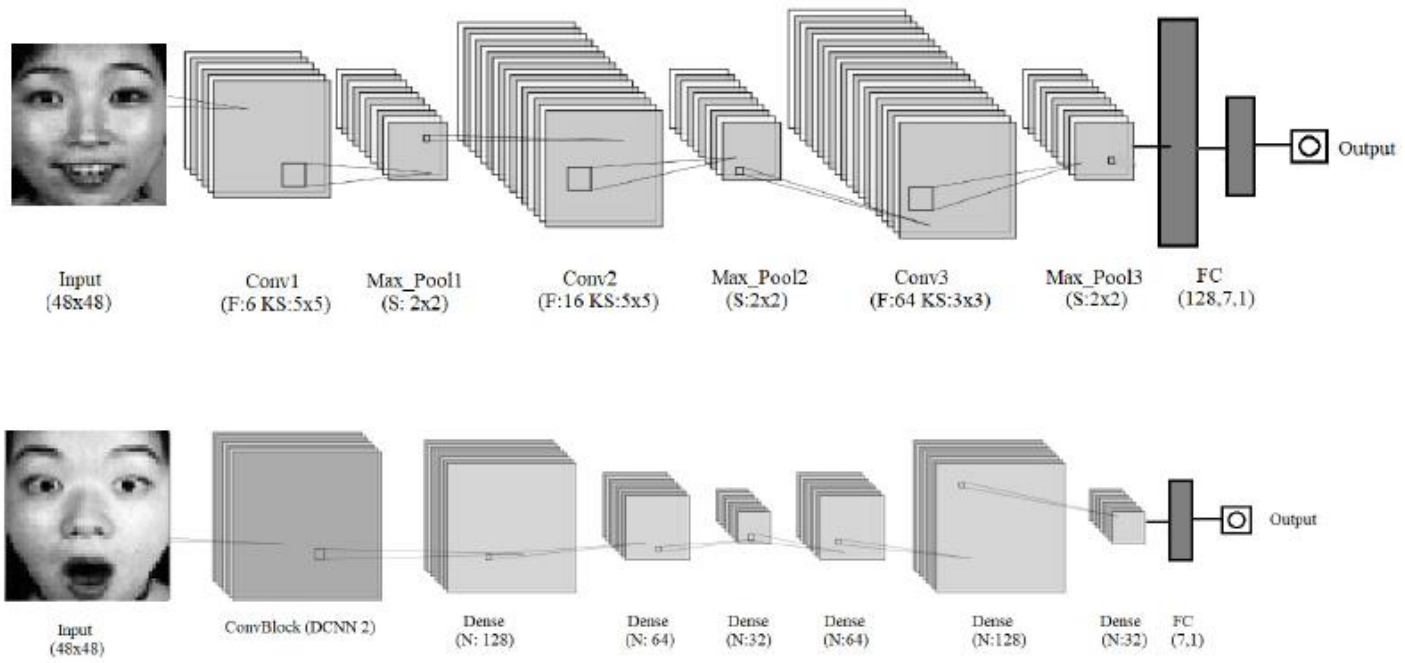

Fig. 2. Workflow of the *DCNN3* (top row) and *DCNN+Autoencoder* (bottom row) models; F - number of filters; KS - kernel size; S - max-pooling size; ConvBlock (DCNN2) - KS - convolutional kernel size; S - max pool kernel size; N- number of neurons

For a better understanding of the proposed method, we focused our description based on the DCNN3 and DCNN+Autoencoder model, as shown in Fig.2. The DCNN3 model receives an input image with size (48x48) that transits through a convolution layer with six filters and a kernel size of 5x5 where the main features of the input image are extracted (see Fig.2 top row). Then, a max-pooling layer of size 2x2 is applied to reduce the spatial variance in the features extracted by the first layer. The same process is applied two more times but, in this case, we used a convolution layer of 16 filters with a kernel size of 5x5 followed by another convolution layer of 64 filters with a kernel size of 3x3. Also, there is a max-pooling layer at the end of each convolution layer of the same size (2x2). Then, it is flatten the output of the last convolutional layer to use it as an input of the fully-connected layer (128,7,1 neurons), which provides the final classification (output).

On the other hand, the DCNN+Autoencoder model uses first a convolution block, as described in the DCNN3 model followed by an autoencoder topology with a deep feed-forward (fully-connected) model (see Fig.2 bottom row). In this network, after the convolution block, the flattened input image passes through an encoder process on the first three dense layers with 128, 64, and 32 neurons respectively. After that, the decoding process starts with the information in the latent space of the network (code) and feeds three more dense layers with 64, 128, and 32 neurons respectively. This generates a compressed version of the original data which is passed to a final fully-connected layer of (7,1) neurons for the final classification.

The remaining deep models, DCNN1, DCNN2, and DCNN4 follow the same logic of the DCNN3 architecture but with some variations in the convolution layers. The core architecture of all the proposed models is summarized in Table I.

TABLE I
CORE ARCHITECTURE OF PROPOSED DEEP MODELS

DCNN1	*DCNN2*	*DCNN3*	*DCNN4*	*DCNN+Autoencoder*
Conv.(64)+Kernel (3x3)	Conv.(6)+Kernel (5x5)	Conv.(6)+Kernel (5x5)	Conv.(6)+Kernel (5x5)	Conv.(6)+Kernel (5x5)
Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)
Conv.(128)+Kernel (5x5)	Conv.(16)+Kernel (5x5)	Conv.(16)+Kernel (5x5)	Conv.(16)+Kernel (5x5)	Conv.(16)+Kernel (5x5)
Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)
Conv.(512)+Kernel (3x3)	Conv.(64)+Kernel (5x5)	Conv.(64)+Kernel (3x3)	Conv.(120)+Kernel (3x3)	Conv.(64)+Kernel (3x3)
Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)	Max-pooling (2x2)
Conv.(512)+Kernel (3x3)	Dense (128)	Dense(128)	Dense (128)	Dense (128)
Max-pooling (2x2)	Fully connected (7,1)	Dense(64)	Fully connected (7,1)	Dense (64)
Dense(256)		Dense (32)		Dense (32)
Dense (512)		Dense(32)		Dense(64)
Fully connected (7,1)		Fully connected (7,1)		Dense(128)
				Dense(32)
				Fully connected (7,1)

Experimental setup

Image Preprocessing

All images were processed to keep only the area of interest inside the image (the facial area without ear and hair). Thus, we used the frontal face cascade classifier method from the OpenCV library to crop the facial area of each image. Keeping only the facial

information allows us to reduce noise and fed the classification models with relevant information. Moreover, all images were resized to meet the required input size (48x48) of the proposed models and normalized with the min-max technique to avoid data dispersions during the model's learning.

Additionally, data augmentation techniques were implemented to increase the number of samples per class on all databases. Thus, operations such as rescale, shear range, zoom range, rotation range, width shift range, and height shift range were applied to achieve approximately 6867 images per database. This augmentation helps with the learning process of the models and avoids overfitting**.**

Test and Training partitions

For all databases, the stratified 5-fold cross-validation method (Gupta, 2017) was applied. In this way, samples are divided into disjoint training and test partitions per fold with samples representation of each output class. Training and testing the models on different partitions guarantee successful learning and further generalization.

Models configuration

For all models, we optimized two main hyperparameters, the training iterations (epochs) in the range from 1 to 100 epochs, and the batch size was set to 32 and 64. Other parameters were used with a standard (fixed) configuration such as the same Adam optimizer, which uses the stochastic gradient descent to update the weights of the models during the training process (Brownlee, 2017). The learning rate was set to 0.01 and the dropouts value to 25% at the end of each convolution layer.

Assessment metrics

We used the mean of accuracy (ACC) to validate the performance of proposed models over five folds. The cross-entropy loss function is also used to assess the probability of a given input sample being classified in the correct class. The more the loss's score is close to zero, the better classification of the input sample (Koech,2020).

SELECTION CRITERIA

The best model will be selected according to the following criteria: (1) The higher mean of ACC score among all models and (2) if there is a tied performance score, the model with the least algorithm complexity is preferred.

The implementation of the proposed method was done with Python programming language version 3.8.3 (Python Core Team, 2019) using scikit-learn (SKlearn) (Pedregosa et al, 2011) and Keras (Chollet et al, 2015) with ImageDataGenerator and TensorFlow backend.

RESULTS AND DISCUSSION

Each model will be trained and tested on the datasets indivually and then the datasets will be combined into the Jaffe & CK+ dataset for further experimentations. The overall values of ACC, datasets and hyperparameters after evaluating each model will be summarized on table II.

TABLE II
PERFORMANCE RESULTS OF DEEP LEARNING MODELS ON THE THREE DATABASES.

Architecture	Input shape	Optimizer	Learning rate	Batch size	Epochs	*Jaffe* ACC	*CK+* ACC	*Jaffe & CK+* ACC
DCNN1	(48, 48, 1)	adam	$1 \cdot 10^{-2}$	32	25	10	15	12
				32	50	14	22	13
				32	75	17	27	13
				32	100	19	29	14
				64	25	11	45	10
				64	50	13	48	11
				64	75	14	53	13
				64	100	16	55	15
DCNN2	(48, 48, 3)	adam	$1 \cdot 10^{-2}$	32	25	76	73	64
				32	50	83	80	70
				32	75	88	83	79
				32	100	90	85	87
				64	25	78	87	72
				64	50	86	91	78
				64	75	91	93	85
				64	100	93	**94**	90
DCNN3	(48, 48, 3)	adam	$1 \cdot 10^{-2}$	32	25	43	74	75
				32	50	54	77	79
				32	75	57	79	83
				32	100	59	80	87
				64	25	50	70	72
				64	50	66	76	80
				64	75	73	80	83
				64	100	75	81	88
DCNN4	(48, 48, 3)	adam	$1 \cdot 10^{-2}$	32	25	64	69	70
				32	50	72	74	74
				32	75	77	82	79
				32	100	82	85	82
				64	25	77	70	76
				64	50	89	76	81
				64	75	93	84	84
				64	100	**95**	90	86
DCNN+Autoencoder	(48, 48, 3)	adam	$1 \cdot 10^{-2}$	32	55	64	70	76
				32	50	68	75	79
				32	75	72	77	82
				32	100	80	83	87
				64	25	50	67	73
				64	50	68	80	89
				64	75	84	83	90
				64	100	92	90	**96**

ACC - mean of the ACC metric over five folds.

In general, the proposed deep-learning models gave accuracies above 85% which is remarkable to deal with the classification problem. The main exception was the DCNN1 model that obtained very poor results: the maximum ACC was 55% on the CK+ dataset followed by 16% on the Jaffe dataset and 15% on the Jaffe & CK+ dataset. This network is the most complex which suggests that it extracts too many irrelevant features and thus decreasing the performance of the network.

For the Jaffe dataset, the more optimal results arrived on the DCNN4 model. In fact, this model provided 95% ACC on the 100th epoch and a batch size of 64. The DCNN4 architecture creates 120 kernels on the last convolution layer. This is the biggest convolution layer out of all the architectures allowing the network to extract more important features. Due to the low number of 10 subjects in the Jaffe dataset, the network is able to recognize and extract more features easily. The DCNN2 and DCNN+Autoencoder had similar results (92% and 93%), but the DCNN3 network scored a lower ACC of 75% due to a worst feature abstraction from the complexity of this architecture.

For the CK+ dataset, the better results appeared on the DCNN2 model. In reality, this architecture achieved 94% ACC on the 100th epoch and a batch size of 64. The DCNN2 architecture is the least complex network; it only implements three convolution layers and one fully connected layer before the output layer. The CK+ dataset has a wide variety of subjects that grant the model the ability to generalize features without needing too many convolutional layers. The DCNN4 model and the DCNN+Autoencoder had the

same ACC. In the other hand, the DCNN3 architecture obtained a worst ACC of 81% from the difficulty of extracting relevant features of this model.

Finally, for the combination of the Jaffe & CK+ datasets, the DCNN+Autoencoder reached better results. As a matter of fact, this model scored 96% ACC on the 100th epoch and a batch size of 64. When combining both datasets, it is more difficult to extract relevant features due to the noise generated by this combination. The autoencoder architecture helps reducing the noise and the spatial variance of each image. In this context, an autoencoder topology will outperform the rest of the models. The rest of architectures (DCNN2, DCNN3 and DCNN4) achieved around 85% ACC on this dataset.

According to our selection criteria, the DCNN+Autoencoder architecture outperformed the rest of the models due to the highest ACC obtained. Moreover, this network also achieved high accuracies in the datasets individually: 92% on the Jaffe dataset and 90% on the CK+ dataset. In addition, the best architectures for each dataset experimentation dealt with the overfitting problem effectively. Following the plots presented in Fig.3 the curves of the mean validation loss are very similar to the curves of the mean training loss for the Jaffe, CK+ and Jaffe & CK+. The plot for the Jaffe dataset indicates the training loss and the validation loss start converging at the 10th epoch on the DCNN4 model; the same scenario reproduces for the Jaffe & CK+ dataset on the DCNN+Autoencoder architecture. Lastly, the curves start converging on the 75th epoch for the CK+ dataset on the DCNN2 network. This shows that our architectures are good for generalization on the testing sets distinct from the training sets.

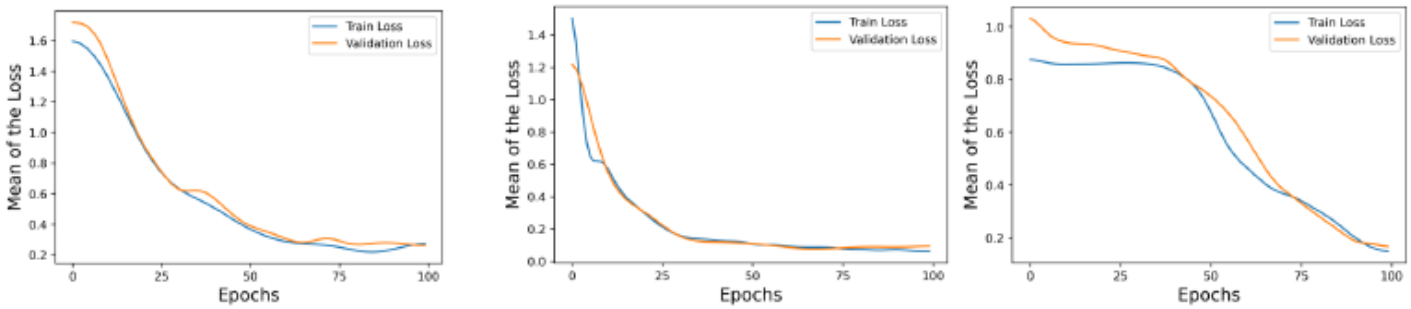

Fig. 3. Performance of the mean cross-entropy loss function over five folds for the best model on the *Jaffe* (left), *CK+* (center), and *Jaffe & CK+* (right) databases.

CONCLUSIONS AND FUTURE WORK

In this work, five different deep CNN architectures were proposed: four DCNNs worked better with 1 dataset (DCNN1, DCNN2, DCNN3 and DCNN4) and one performed better with two distinct datasets (DCNN+Autoencoder), for the classification an ACC test was provided as well as a cross-entropy loss function to measure the performance regarding overfitting of our models. The DCNN1 model had the worst results overall due to high loss values and thus big overfitting problems, as well as very low accuracies in all datasets (55%, 20% and 19%). The best architecture for the Jaffe dataset was the DCNN4 network. On the other hand, DCNN scored the best results on the CK+ dataset. Lastly, the Jaffe & CK+ dataset achieved better results on the DCNN+Autoencoder model.

As future work, we plan on further exploring our autoencoder architecture with more datasets, as well as different hyperparameters such as the number of epochs, optimizers and batch sizes.

ACKNOWLEDGEMENT

Authors thank to the Applied Signal Processing and Machine Learning Research Group of USFQ for providing the computing infrastructure (NVidia DGX workstation) to implement and execute the developed source code. Publication of this article was funded by the Academic Articles Publication Fund of Universidad San Francisco de Quito USFQ.

REFERENCIAS

Brownlee. (s.f.). *machineleaningmastery*. Obtenido de https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

DeepAI. (2020). *DeepAI*. Obtenido de https://deepai.org/machine-learning-glossary-and-terms/autoencoder

Deepak Kumar, J., Pourya, S., & Paramjit, S. (2019). Extended deep neural network for facial emotion recognition. *ScienceDirect*, doi: 10.1016/j.patrec.2019.01.008.

DeepLizard. (2020). *DeepLizard*. Obtenido de https://deeplizard.com/learn/video/OT1jslLoCyA

Goodfellow I., E. D.-H. (2014). Challenges in Representation Learning: A report. *IEEE TRANSACTIONS ON CYBERNETICS*, doi: 10.1016/j.neunet.2014.09.005.

Gordillo, F. a. (2017). La coherencia entre lo que saben de ti y lo que ven en tu rostro afecta a la valoración de tu personalidad. *ResearchGate*, 11.

Gordillo, F. (s.f.). *Neurologia*. Obtenido de https://www.neurologia.com/noticia/5101/importancia-de-la-expresion-facial-de-las-emociones

Gupta. (2017). Deep Learning: Feedforward Neural Network,*TowardsDataScience*. Obtenido de https://towardsdatascience.com/deep-learning-feedforward-neural-network-26a6705dbdc7

Gupta. (2017). Cross-Validation in Machine Learning ,*TowardsDataScience*. Obtenido de https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f

Hao Z., R. W. (2020). Deep reinforcement learning for robust emotional classification in facial expression recognition. *ScienceDirect*, doi: 10.1016/j.knosys.2020.106172

Hepeng, Z., Bin, H., & Guohui, T. (2019). Facial expression recognition based on deep convolution longshort-term memory networks of double-channel weighted mixture. *ScienceDirect*, doi: 10.1016/j.patrec.2019.12.013.

Huiting, W., Yanshen, L., Yi, L., & Liu, S. (2019). Efficient facial expression recognition via convolution neural network and infrared imaging technology *ScienceDirect*, doi: 10.1016/j.infrared.2019.103031.

Jing, L., Kan, J., Dalin, Z., Naoyuki, K., & Zhaojie, J. (2020). Attention mechanism-based CNN for facial expression recognition. *ScienceDirect* ,doi: 10.1016/j.neucom.2020.06.014.

Koech. (2020). *Cross-Entropy Loss Function*. Obtenido de https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e

Kumar J., P. S. (2019). Extended deep neural network for facial emotion recognition. *ScienceDirect*.

Mengyi, L., Shaoxin, L., Shiguang, S., & Xilin, C. (2015). AU-inspired Deep Networks for Facial Expression Feature Learning. *ScienceDirect*, doi: 10.1016/j.neucom.2015.02.011.

ML GLossary. (2017). *ML-CheatSheet.* Obtenido de https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html

Nacer, F., & Mahdi, H. (2018). Exemplar-based facial expression recognition. *ScienceDirect*, doi: 10.1016/j.ins.2018.05.057.

OpenCV. (2020). *Opencv*. Obtenido de https://opencv.org/about/

Pantic M., L. J. (2002). Facial gesture recognition in face image sequences: a study on facial gestures typical for speech articulation. *Researchgate*. Obtenido de https://www.researchgate.net/publication/3996132_Facial_gesture_recognition_in_face_image_sequences_a_study_on_facial_gestures_typical_for_speech_articulation

Ronchetti, F. (s.f.). *Core*. Obtenido de https://core.ac.uk/download/pdf/153562887.pdf

Simonyan K., Z. A. (2015). Very deep convolutional networks for large-scale image recognition. *ICLR* doi: arXiv:1409.1556.

Stone. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological*.

Veena, M., Radhika, M. P., & Manohara, P. M. (2016). Automatic Facial Expression Recognition Using DCNN. *ScienceDirect*, doi: 10.1016/j.procs.2016.07.233.

Wei Z., Y. Z., M., L., & Jingwei G., S. G. (2015). Multimodal learning for facial expression recognition. *ScienceDirec*, 12.

Wei, Z., Youmei, Z., Lin, M., Jingwei, G., & Shijie, G. (2015). Multimodal learning for facial expression recognition. *ScienceDirect*, doi: 10.1016/j.patcog.2015.04.012.

Huadong, L. (2020). Discriminative deep multi-task learning for facial expression recognition. *ScienceDirect*, doi: 10.1016/j.ins.2020.04.041.

Python Core Team, Python 3.8.3: A dynamic, open-source programming language., Python Software Foundation, 2019. [Online]. Available: https://www.python.org/.

Michael J. Lyons, Miyuki Kamachi, Jiro Gyoba.
Coding Facial Expressions with Gabor Wavelets (IVC Special Issue)
arXiv:2009.05938 (2020) https://arxiv.org/pdf/2009.05938.pdf

The Extended Cohn-Kanade Dataset
(CK+): A complete expression dataset for action unit and
emotion-specified expression. Proceedings of the Third International
Workshop on CVPR for Human Communicative
Behavior Analysis (CVPR4HB 2010) San Francisco, USA,
94-101.