Mapas de expresiones para la clasificación de expresiones faciales basadas en features

Alicia Alvarez Mon[1], María Elena Buemi[1,2], and Daniel Acevedo[1,2]

[1] Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Computación. alvarezmonaliciaa@gmail.com
[2] CONICET-UBA. Instituto de Investigación en Cs. de la Computación (ICC). Ciudad Autónoma de Buenos Aires, Argentina. {mebuemi,dacevedo}@dc.uba.ar

Abstract. La clasificación de emociones en imágenes o videos de caras es un tema que es abordado con diferentes enfoques. Algunos de ellos se encuentran basados en técnicas de alto costo computacional como los que utilizan aprendizaje profundo. Otros enfoques más clásicos utilizan herramientas basadas en *features* o características extraídas de la cara, como aquellas que incluyen información de forma, textura o estructura. En este trabajo proponemos la clasificación de expresiones mediante la construcción de un mapa de expresiones basado en mapas auto-organizados. Dicho mapa se entrena a partir de vectores de dimensión reducida conteniendo información de forma, estructura (a partir de landmarks faciales), y de textura (a partir de componentes de alta y baja frecuencia). Sobre la base CK+ alcanzamos una tasa de reconocimiento superior al 80% a través del mapeo de las instancias de test a los agrupamientos representados en el mapa.

Keywords: Expresiones faciales · Reconocimiento · Features.

1 Introducción

La clasificación de expresiones faciales, es un tema de gran relevancia, debido a la gran cantidad de tópicos que tienen como base la identificación de dichas expresiones. Estas clasificaciones se basan en un modelo categórico que etiqueta una secuencia de video con una de las siguientes expresiones: *Happiness*, *Surprise*, *Disgust*, *Fear*, *Anger* y *Sadness*. Hay numerosa bibliografía sobre la tarea de identificar expresiones faciales. Entre los numerosos métodos que se reportan, aquellos basados en extracción de features resultan interesantes por su bajo costo computacional, por ejemplo, LBP [13], Wavelet [14], filtros de Gabor [12], etc. Estos métodos pueden separarse en aquellos que tratan la imagen estática para los que se desarrollaron los métodos basados en apariencia y los que clasifican emoción a partir de secuencias dinámicas [3]. Estos últimos siguen el desplazamiento de landmarks a través de los frames de un video. Un compendio completo de metodologías para reconocer expresiones se puede encontrar en Kopalidis et al. [8]. La clasificación de expresiones faciales con mapas de expresiones están basados en mapas autoorganizados (SOM) y permiten la clasificación no supervisada de expresiones faciales basadas en features. Los SOM fueron presentados

por Teuvo Kohonen y emulan el proceso de memorización y categorización de la corteza óptica del cerebro humano utilizando features en su aprendizaje. El uso de mapa de expresiones para etiquetar expresiones faciales, es presentado por Agrawal et al. [2], que lo utiliza para abordar problemas de síntesis de expresiones faciales. El presente trabajo está orientado a la clasificación de expresiones faciales usando mapas de expresiones y que se diferencia de otras técnicas al utilizar features que contienen información de forma, estructura y textura para aminorar ruido y mantener detalles. Se trata de un método no supervisado que permite obtener una tasa de reconocimiento confiable para tratamientos posteriores.

El trabajo se organiza de la siguiente manera. En la Sec. 2 se describe cómo se obtienen los features y cómo se construye el mapa de expresiones con el SOM. En la Sec. 3 se describe el dataset, detalles del entrenamiento del SOM y los resultados finales. Finalmente, se presentan las conclusiones en la Sec. 4.

2 Metodología y Clasificación

Utilizamos la database CK+ [10], que está agrupada en secuencias de imágenes de una persona. En cada serie la persona realiza una emoción. La primera imagen de cada secuencia es la cara neutral del sujeto, y a medida que avanza la secuencia se intensifica la emoción mostrada, teniendo el pico de intensidad en la última imagen o frame de la secuencia.

Tenemos un modelo que utiliza un Mapa de Expresiones que entrena con los vectores de features calculados sobre los datos de entrenamiento y, luego de entrenado el modelo, recibe como dato de entrada una serie de imágenes que expresa una emoción, y nos permite identificar esa emoción.

2.1 Extracción de Features

Para trabajar con las imágenes de caras primero se realiza un alineamiento a una forma de referencia usando el análisis Procrustes. El dataset CK+ cuenta con 68 landmarks faciales, que delimitan la forma de la cara.

Nuestras imágenes están agrupadas en secuencias de k frames donde cada una de ellas se corresponde a un sujeto realizando una emoción; cada sujeto puede tener asociado más de una secuencia en caso de realizar más de una emoción.

El vector final de features final estará formado por la concatenación de otros tres vectores que tiene componentes de textura, forma y estructura.

Para obtener la componente de forma b^s se calcula la diferencia entre los landmarks de cada frame de una secuencia con el frame inicial tras una posterior reducción de la dimensión. Es decir, siendo f_i el vector de landmarks del i-ésimo frame, $1 \leq i \leq k$, obtenemos el vector $d_i^s = f_i - f_0$ como componente de forma. Realizamos reducción de la dimensión a través de Análisis de Componentes Principales (PCA). Para ello definimos la matriz P^s de transformación obtenida a partir un total de r frames de $2l$ atributos cada una (siendo l la

cantidad de landmarks). Para cada frame obtenemos el vector b_j^s, con $1 \leq j \leq r$, como

$$b_j^s = (P^s)^t(d_j^s - \mu^s) \tag{1}$$

Para calcular el componente de estructura b^e, utilizamos 17 atributos medidos a partir de los landmarks, 4 que utilizan además la textura, y uno que marca el nivel de intensidad de emoción del frame [2]. Para este último atributo se observa una variación significativa durante el curso de la secuencia.

La intensidad de emoción en un frame se estima midiendo la cantidad de desviaciones de forma respecto del frame inicial de la secuencia, por lo que usamos el vector d_j^s del feature anterior, promediado sobre todos los landmarks en ese frame. Llamamos a_i al factor de normalización, que es el máximo valor del elemento de los vectores d_{ji}^s $(0 \leq i \leq 2l)$ de todos los frames. Entonces, la intensidad de emoción del frame k se la representa con la siguiente ecuación:

$$\gamma_k = \frac{\sum_{i=1}^{2l} d_{ji}^s / a_i}{2l} \tag{2}$$

Luego, para los demás atributos, 17 solo usan los landmarks para definirlos [11], mientras que los 4 puntos restantes usan más elementos de la textura [1]. El primero es el average hull de los puntos faciales de los labios. Los otros 3 requieren conseguir el coeficiente de curvatura de la sección de las mejillas.

Sea $(x, y, I(x, y))$ un píxel donde $I(x, y)$ es la intensidad del mismo en la posición (x, y). Dicho píxel tiene las 2 tangentes asociadas: $(1, 0, \partial I/\partial x)$ representando el ritmo de cambio de $I(x, y)$ con respecto a x, y $(0, 1, \partial I/\partial y)$ lo mismo para la dirección y. Consideramos el vector normalizado

$$\alpha(x, y) = \frac{(\partial I/\partial x, \partial I/\partial y, -1)}{\|(\partial I/\partial x, \partial I/\partial y, -1)\|} \tag{3}$$

donde $\|\cdot\|$ representa la magnitud de un vector. Un vector normalizado en dirección hacia el eje z es $\beta = [0, 0, 1]$. Definimos coeficiente de curvatura $c(x, y)$ en la posición (x, y) como

$$c(x, y) = 1 - \alpha(x, y) \cdot \beta \tag{4}$$

Calculamos la curvatura por cada elemento dentro del borde y sacamos su media y sus centros de masa para los atributos restantes.

Luego de calcular todos los 22 atributos en cada frame obteniendo el vector φ_j^e correspondiente, mediante la matriz P^e de PCA obtenemos el vector reducido b^e definido como

$$b_j^e = (P^e)^t(\varphi^e - \mu^e). \tag{5}$$

Finalmente, obtenemos los componentes de textura. Dado una imagen f, queremos descomponerla como $f = u + v$, siendo u el denominado *cartoon* o componente geométrico, y v la *textura* propiamente dicha, que refiere a las oscilaciones y el ruido. Utilizamos una descomposición cuyo framework se basa en los modelos de Meyer et al. [5, 9]:

$$\inf_{(u,v)\in X_1 \times X_2} \{F_1(u) + \lambda F_2(v) : f = u + v\} \tag{6}$$

con λ un parámetro que regula la granularidad de la textura, donde tenemos que elegir X_1 y X_2, así como F_1 y F_2 según el problema, siguiendo siempre la idea de que v es penalizado por F_1 ($F_1(v) >> F_2(v)$) y viceversa. Usamos la Total Variation y la norma L1 (TV-L1), y el algoritmo de Chambolle-Pock [9] para la descomposición en u (cartoon) y v (textura). El componente v representa expresión, y el u todavía depende de datos del entorno como la iluminación y tono.

Para contrarrestar la dependencia de u del entorno, dividimos cada elemento por la media de u en esa imagen. Luego, para una serie de frames, en donde llamamos u_k y v_k a las descomposiciones del k-ésimo frame, se utiliza directamente la diferencia entre el k-ésimo frame de la secuencia y el inicial, que llamamos $d_k^u = u_k - u_0$.

Calculados estos datos para todos los frames de todas las secuencias, usamos PCA para reducir la dimensión y obtener los componentes de textura:

$$b_r^u = (P^u)^t(d_r^u - \mu^u), \quad b_r^v = (P^v)^t(v_r - \mu^v) \tag{7}$$

los vectores b_r^u y b_r^u son los componentes de textura del vector b de cualquier frame r de alguna secuencia.

Por último, concatenamos los componentes b^s, b^e, b^u y b^v para obtener el vector b para describir nuestro dominio de expresiones.

2.2 Mapa de Expresiones

Nuestro modelo de Mapas de Expresiones se encuentra basado en el SOM (Self Organizing Map) [6]. Podemos pensarlo como una grilla de dos dimensiones con una neurona por cada nodo de la grilla.

Los datos de entrenamiento consisten en los vectores de features b según se detalló en la sección previa, uno por cada imagen del dataset utilizado. Las imágenes consideradas son aquellas en las cuales algún sujeto del dataset se muestra sin realizar ninguna expresión (neutro) o mostrando una emoción en su máxima expresión (apex). Si cada vector de features b tiene D dimensiones entonces cada neurona j ($j = 1 \dots m$) del mapa se conecta al espacio de entrada por un vector w_j de pesos D dimensional.

En el mapa entrenado, cada nodo guarda un patrón de expresión de la forma del vector w_j. Al comienzo del entrenamiento, todos los pesos se inicializan aleatoriamente. Después del entrenamiento, el mapa puede describir el espacio de emociones, y las emociones parecidas se espera que se ubiquen topológicamente cerca.

Como se mencionó, el mapa de expresiones usa el método de aprendizaje de Self Organizing Map (SOM) [4]. El entrenamiento se realiza en 3 etapas:

1. Competición: Dado un vector de entrada b, las neuronas compiten entre sí. La neurona $\kappa(b)$ cuyo vector de pesos $w_{\kappa(b)}$ mejor se ajuste a b es la ganadora.
2. Cooperación: La ganadora $\kappa(b)$ coopera con el resto de sus vecinas para adaptarse. Se usa la función de vecindario $f_{j,\kappa(b)}$ monótamente decreciente de

distribución de distancia espacial en dos dimensiones, siendo j una neurona vecina.

3. Adaptación: durante la k-ésima etapa del proceso de entrenamiento se actualizan los valores de los pesos w_j asociados a cada neurona j, $1 \leq j \leq m$, según la siguiente ecuación:

$$w_j(k+1) = w_j(k) + \eta(k)\, f_{j,\kappa(b)}\ \ (b - w_j(k)) \tag{8}$$

donde k es la iteración y η es la tasa de aprendizaje (learning rate) que decrece con cada iteración. La función de vecindario $f_{j,\kappa(b)}$ es monótoma decreciente, cuyos parámetros j y $\kappa(b)$ indexan los pesos $w_j(k)$ y $w_{\kappa(b)}(k)$ de la iteración k-ésima (ver Eq. 9).

El proceso de entrenamiento del mapa de expresiones es no supervisado. Al final de este proceso se realiza una etapa de 'etiquetado' de las neuronas, es decir, se les asigna (si es posible) alguna clase o emoción asociada.

Inicialmente, mapeamos cada vector de features del entrenamiento con la neurona cuyo vector de pesos w sea más similar. Al finalizar, observamos cada neurona, y la mapeamos a la emoción que más vectores asociados a ésta compartan. En caso de no tener ningún vector asociado, no se le asigna ninguna etiqueta a esa neurona. Un ejemplo del resultado de este proceso de etiquetado se muestra en la Figura 1.

Para el proceso de clasificar la emoción de una secuencia, calculamos el vector de features de su último frame (emoción en apex) siguiendo el proceso descripto en la Sec. 2.1. La emoción final que se le asigna es la clase perteneciente a la neurona del SOM más cercana a este vector. En el caso de que esta neurona no esté etiquetada, se observan sus neuronas vecinas, y se asigna la emoción más ocurrente en las neuronas vecinas que fueron etiquetadas en el entrenamiento.

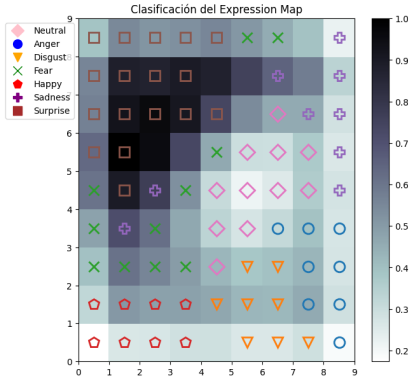


Fig. 1. Mapa de Expresiones caracterizando el agrupamiento de diferentes expresiones para una grilla de 9×9 neuronas. El color de cada celda es la suma normalizada de las distancia euclídea de una neurona y sus vecinos.

3 Experimentación

3.1 Dataset

Utilizamos el dataset CK+ [10], que posee 123 sujetos distintos, que actúan en 593 secuencias de videos de emociones en total. De las 7 emociones disponibles, se descartan las series que no están etiquetadas o son de la clase 'contempt', quedando 320 series. Contando todos los frames que constituyen las secuencias, tenemos un total de 5849 frames. Dividimos un 70% para entrenamiento y 30% para testing, por lo que la parte de training tiene 225 series de 86 personas, con 4122 frames en total y la parte de test tiene 37 sujetos, 95 series y 1727 frames en total.

En el entrenamiento de nuestro modelo, se utiliza únicamente los frames neutrales (sin intensidad de emoción) y los que muestran la emoción en su *apex*, que corresponden al primer y último frame de cada secuencia. Esto significa que vamos a usar todos los vectores de entrenamiento para etiquetar las neuronas, y que descartamos frames intermedias para centrarnos en la información provista por las emociones con sus rasgos bien definidos.

En el conjunto de entrenamiento tenemos $225 \times 2 = 450$ frames ya que se consideran 2 frames por secuencia. De las secuencias que se corresponden a un mismo sujeto, solo se considerará un frame neutral.

Para el testeo utilizamos las 95 secuencias de testing, donde nos interesa clasificar la emoción de frames en apex.

3.2 SOM

Nuestro SOM tiene como parámetros la cantidad de neuronas, la dimensión del vector de features, el learning rate inicial y el σ que es un parámetro de la función espacial de vecindario, siendo ésta definida como:

$$f_{j,\kappa(b)} = \exp\left(\frac{-\|w_j - w_{\kappa(b)}\|^2}{\sigma^2}\right) \tag{9}$$

donde w_j son los pesos asociados a la neurona j-ésima, $\kappa(b)$ es la neurona más cercana al vector de features b, $w_{\kappa(b)}$ son los pesos de esa neurona. El parámetro σ corresponde al valor del radio alrededor de la neurona más cercana y afecta la magnitud de cambio de los pesos. No se modifica en el tiempo como lo hace el learning rate

Dado n que indica la cantidad de instancias a entrenar, se suele estimar una cantidad de neuronas cercana a $5 \cdot \sqrt{n}$ para el entrenamiento [7]. Buscamos una matriz cuadrada de $r \times r$ que se corresponderá al mapa de neuronas. El r utilizado será un valor tal que $r^2 \approx 5 \cdot \sqrt{n}$.

Nuestra cantidad inicial de vectores de entrenamiento es de 297, pero debido a que hay un gran disparidad en la cantidad de secuencias de 'fear' y 'sadness' en comparación al resto, se les agrega los penúltimos frames a las secuencias de ambas. Como se muestra en la Tabla 1, este cambio genera una distribución más balanceada de las emociones en el conjunto de entrenamiento.

Table 1. Distribución de emociones en los frames etiquetados de los datos de entrenamiento.

Emociones	Neutral	Anger	Disgust	Fear	Happy	Sadness	Surprise
Distr. Original	72	60	34	19	50	21	41
Distr. Balanceada	72	60	34	38	50	42	41

Por lo tanto tenemos $297+19+21 = 337$ frames. La estimación de la cantidad de neuronas a utilizar nos da alrededor de 91 neuronas, por lo que usamos una matriz de tamaño 9×9 para el mapa de neuronas (81 en total).

Los vectores de features se calculan para cada frame según lo descripto en la Sección 2.1. Cada vector será de 492 componentes en total,resultado de, cada vez que se aplica la reduccion de la dimensión por PCA, conservar más de 90% de varianza explicada.

3.3 Resultados

Table 2. Porcentajes promedio de acierto del SOM sobre el conjunto de entrenamiento para distintas combinaciones de valores iniciales de σ y learning rate.

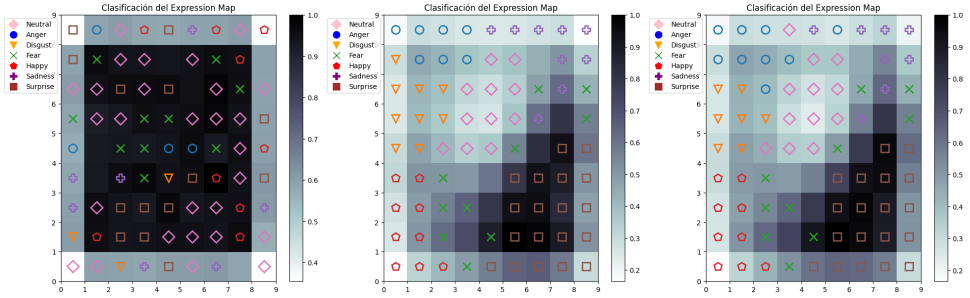
Learning Rate	σ	% Accuracy
0.5	1.75	87.29 ± 1.10
0.5	1	84.42 ± 2.30
0.5	1.5	87.38 ± 2.54
0.5	2	88.33 ± 1.62
0.5	2.25	88.18 ± 2.17
0.25	1.75	**88.90 ± 1.31**
0.25	1	84.80 ±1.72
0.25	1.5	86.94 ± 0.99
0.25	2	88.72 ± 1.67
0.25	2.25	88.39 ± 2.17
0.15	1.75	88.45 ± 1.22
0.15	1	85.34 ± 2.20
0.15	1.5	87.29 ± 1.70
0.15	2	88.60 ± 1.32
0.15	2.25	88.39 ± 2.42

Entrenamos nuestro SOM con los datos de entrenamiento y estimamos la tasa de acierto en el conjunto de entrenamiento. Debido a la naturaleza aleatoria tanto en la inicialización de pesos como en el proceso de entrenamiento del SOM, testeamos varios valores iniciales de hiperparámetros (learning rate y σ). El rango de σ es el mismo usado en la demo presentada en [9], cuyo código usamos para la decomposición de los componentes de textura. En la tabla 2 se muestra el promedio de porcentajes de aciertos en el conjunto de entrenamiento.

El mejor rendimiento para el entrenamiento se logra con valores iniciales de learning rate $= 0.25$ y $\sigma = 1.75$.

En general, nuestro SOM tiene mejores resultados luego de ser entrenado por varias épocas. A partir de las 80 épocas la distribución se termina de estabilizar, como se ve en la figura 2.

Utilizando estos valores iniciales, entrenamos el mapa de expresiones definitivo el cual obtiene un porcentaje de accuracy de 82.45% sobre el conjunto de test.



(a) Inicio (b) 80 Épocas (c) 90 Épocas

Fig. 2. Progresión del mapa de expresiones para diferentes cantidad de épocas.

4 Conclusiones

En este trabajo planteamos un esquema simple de clasificación de imágenes que representan emociones basado en un mapa de expresiones. A partir de la construcción de vectores de características que incorporan información de forma, textura y estructura, utilizamos un método no supervisado para su agrupamiento y etiquetado final.

Nuestro modelo alcanza más del 80% de aciertos con los datos de evaluación con los hiperparámetros elegidos y los features seleccionados. Nuestro modelo además converge rápidamente a una configuración estable dados los datos de entrenamiento.

En el futuro nos proponemos aumentar la cantidad de frames para el entrenamiento del mapa de expresiones además de los que usamos para etiquetar, de esa manera se podría agregar más robustez al entrenamiento.

References

1. Agarwal, S., Chatterjee, M., mukherjee, D.P.: Synthesis of emotional expressions specific to facial structure. In: Proceedings of the Eighth Indian Conference on

Computer Vision, Graphics and Image Processing. ICVGIP '12, Association for Computing Machinery (2012)
2. Agarwal, S., Mukherjee, D.P.: Synthesis of realistic facial expressions using expression map. IEEE Transactions on Multimedia **21**(4), 902–914 (2019)
3. Allaert, B., Ward, I., Bilasco, I., Djeraba, C., Bennamoun, M.: A comparative study on optical flow for facial expression analysis. Neurocomputing **500**, 434–448 (2022)
4. Asan, U., Ercan, S.: An introduction to self-organizing maps. Computational Intelligence Systems in Industrial Engineering: with Recent Theory and Applications pp. 299–319 (11 2012)
5. Buades, A., Le, T., Morel, J.M., Vese, L.: Cartoon+Texture Image Decomposition. Image Processing On Line **1**, 200–207 (2011)
6. Haykin, S.: Neural Networks: A Comprehensive Foundation (3rd Edition). Prentice-Hall, Inc., USA (2007)
7. Jing Tian, M.H.A., Pecht, M.: Anomaly detection using self-organizing maps-based k-nearest neighbor algorithm. PHM Society European Conference **2**(1) (2014)
8. Kopalidis, T., Solachidis, V., Vretos, N., Daras, P.: Advances in facial expression recognition: A survey of methods, benchmarks, models, and datasets. Information **15**(3) (2024). https://doi.org/10.3390/info15030135
9. Le Guen, V.: Cartoon + Texture Image Decomposition by the TV-L1 Model. Image Processing On Line **4**, 204–219 (2014)
10. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops. pp. 94–101 (2010)
11. Mao, H., Jin, L., Du, M.: Automatic classification of chinese female facial beauty using support vector machine. pp. 4842–4846 (10 2009)
12. Pumlumchiak, T., Vittayakorn, S.: Facial expression recognition using local gabor filters and pca plus lda. In: 2017 9th International Conference on Information Technology and Electrical Engineering (ICITEE). pp. 1–6 (2017)
13. Shan, C., Gong, S., Mcowan, P.: Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing **27**, 803–816 (05 2009)
14. Zhang, Z., Lyons, M., Schuster, M., Akamatsu, S.: Comparison between geometry-based and gabor-wavelets-based facial expression recognition using multi-layer perceptron. In: Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition. pp. 454–459 (1998)