ISSN 1870-4069

Aprendizaje profundo de representaciones robustas para clasificación multi-instancia y multi-etiqueta de imágenes

Javier Roberto Veloz Centeno[1], Alfonso Rojas-Domínguez[3], Ivvan Valdez[2], Manuel Ornelas[1], Héctor Puga[1], Martín Carpio[1]

[1] Instituto Tecnológico de León, México

[2] Universidad de Guanajuato, México

[3] CONACYT Research Fellow, México

{veloz_c_22, jmcarpio61}@hotmail.com, alfonso.rojas@gmail.com, si.valdez@ugto.mx, manuel.ornelas@itleon.edu.mx, pugahector@yahoo.com

Resumen. Abordamos el 'Reto de Clasificación de Restaurantes de Yelp', que consiste en predecir los atributos que poseen restaurantes a partir de sus conjuntos de imágenes, etiquetados por la comunidad de Yelp para 9 posibles atributos. Este problema *multi-instancia* y *multi-etiqueta* nos permite explorar una variedad de ideas en el campo de aprendizaje de representaciones. Abordamos el aspecto multi-instancia del problema mediante la agregación de características de alto nivel del conjunto de imágenes de cada restaurante, creando un vector de características prototipo por establecimiento. Las características se extraen mediante Redes Convolucionales Profundas. Posteriormente, utilizamos los vectores de características para entrenar un sistema de clasificadores binarios, uno por atributo. Para mejorar el desempeño de nuestro modelo, inducimos una representación robusta mediante el cálculo de los vectores de características prototipo utilizando redes pre-entrenadas con 3 bases de datos distintas: ImageNet, Food-101 y MIT Places 2. Finalmente, dado que no toda representación individual es igualmente útil para realizar la predicción de un atributo, añadimos un clasificador final que aprende los pesos de predicción para cada representación de un atributo. Nuestra propuesta es un sistema automatizado de principio a fin, que logra un desempeño F1 en el conjunto de evaluación de 0.8177, haciendo a nuestro modelo competitivo con el mejor 10% de los competidores. Recomendaciones y directrices para trabajo futuro también se discuten.

Palabras clave: Redes neuronales convolucionales, aprendizaje de representaciones, clasificación multi-etiqueta, aprendizaje multi-instancia.

Robust Deep Representation Learning for Multi-Instance and Multi-Label Image Classification

Abstract. In this work we address the Yelp Restaurant Photo Classification Challenge which consists in predicting restaurant attributes given its corresponding, variable-size set of images; the restaurant images were provided by Yelp and the labels were annotated by the Yelp Community for the 9 different attributes. The multi-instance and multi-label nature of the problem permits to explore a variety of ideas in the field of representation learning. First, we tackle the multi-instance aspect of the problem by means of aggregating pre-trained CNN feature extractors of a restaurant image-set to create a restaurant prototype feature vector. We then use the aggregated restaurant features to train a system of binary classifiers, one for each attribute. In order to improve our model performance, we induce a robust representation by means of calculating the restaurant prototype features through the use of complementary VGG-16 feature extractors pre-trained on 3 different datasets, namely: Imagenet, Food-101, and MIT Places 2. Due to the fact that not every representation has equal importance for predicting a particular attribute, we add a final classifier which learns a prediction weight for each representation of a given attribute. Our proposal is an end-to-end system, that achieves a test-dataset performance F1-score of 0.8177, which makes our model competitive within the top 10% entries for the challenge. Finally, some recommendations for improvement and future work are discussed.

Keywords: Convolutional neural networks, multi-instance learning, multi-label classification, representation learning.

1 Introducción

En años recientes, apoyándose en un incremento de las capacidades computacionales tales como GPUs, las tareas de visión artificial han sido dominadas por modelos de aprendizaje profundo (DL, por sus siglas en inglés) [1] y en particular por redes neuronales convolucionales (CNNs, por sus siglas en inglés). Los retos de reconocimiento visual a gran escala como el ILSVRC [2] han estimulado la competencia para proponer modelos convolucionales como la red VGG [3], la GoogleNet [4] y las Redes Residuales [5]. Éstos están constituidos por dos secciones: una sección convolucional, para la extracción de características en niveles jerárquicos de complejidad, y la sección de clasificación, donde se aprende la interrelación entre características.

Una ventaja de DL es que los modelos se entrenan automáticamente de principio a fin, evitando el uso de extractores de atributos diseñados por humanos. Esto es posible porque los extractores de atributos de bajo nivel son usados en capas superiores para construir atributos de alto nivel que son específicos para la base de datos en cuestión. Adicionalmente, los pesos aprendidos por estos modelos pueden ser usados en otros

problemas como detección de objetos y segmentación [6] u otros problemas de clasificación. Una técnica que permite el uso de pesos previamente entrenados sobre una base de datos distinta a la tarea en cuestión, es *transfer-learning* mejorada con *fine-tuning* [7]. Aún existen preguntas por contestar sobre el aprendizaje de representaciones, como qué tan buena es la representación optimizada para un problema, dadas distintas bases de datos de entrenamiento.

La mayor parte de los modelos CNN a gran escala están enfocados a la tarea de clasificación de imágenes. El reto de clasificación de restaurantes de Yelp (RCRY)[1], definido sobre una base de datos de restaurantes etiquetados, es un problema de clasificación de imágenes *multi-instancia* (el número de imágenes por restaurante no es fijo) y *multi-etiqueta* (existen 9 posibles atributos por seleccionar). Esto nos permite explorar ideas en aprendizaje de representaciones; en particular, estudiamos cómo inducir características robustas para los restaurantes a través de CNNs pre-entrenadas sobre distintos conjuntos de entrenamiento.

Siguiendo el trabajo en [8], abordamos dicha pregunta por medio de la agregación de características de alto-nivel obtenidas de una red VGG para generar un vector de características por restaurante. Esto se realiza con los extractores de características pre-entrenados en 3 bases de datos distintas. Una vez que los vectores de características por restaurante son calculados, se aplica el procedimiento de fine-tuning en cada uno de los módulos de clasificación de las redes VGG para predecir, por medio de un sistema de clasificadores binarios, si un atributo aplica o no para un restaurante.

Para mejorar el desempeño del sistema, aplicamos aprendizaje por enjambre usando una estructura con 4 pliegues, resultando en un total de 36 módulos de clasificación entrenados. Nuestra metodología posee la ventaja de que maximiza el aprendizaje que se transfiere utilizando 3 bases de datos complementarias, induciendo una representación robusta por restaurante; otra ventaja es la incorporación de un clasificador final que integra las capacidades predictivas de cada representación por etiqueta.

El resto del artículo está organizado de la siguiente manera: La sección 2 presenta antecedentes teóricos. La sección 3 proporciona una descripción de nuestra propuesta. El diseño de experimentos se describe en la sección 4. Nuestros resultados en el PCRY se reportan en la Sección 5. Conclusiones y direcciones para trabajo futuro se presentan en la Sección 6.

2 Antecedentes

En los problemas *multi-instancia* (MIL, por sus siglas en inglés), un número arbitrario de instancias está asociado con una etiqueta de clase. Por lo tanto, el etiquetado de los datos de entrenamiento se vuelve más sencillo (ya que se realiza en conjunto, en vez de manera individual) con la desventaja de que se produce una base de datos débilmente supervisada [9]. En el PCRY cada restaurante está representado por un conjunto de imágenes que comparten la(s) etiqueta(s) de atributos de dicho

[1] https://www.kaggle.com/c/yelp-restaurant-photo-classification

Fig. 1. Ejemplos de imágenes de entrenamiento para 3 restaurantes aleatorios (en filas).

establecimiento; algunos ejemplos de los conjuntos de entrenamiento se muestran en la Fig. 1. Es necesaria una función que relacione las etiquetas por instancia con las etiquetas por conjunto. Bajo la suposición estándar multi-instancia (SMIA, por sus siglas en inglés) un conjunto pertenece a la clase 'positiva' si al menos una de sus instancias es 'positiva' [10]. Siguiendo la SMIA, un problema MIL puede ser visto como un problema de clasificación de una sola instancia, al asignar las etiquetas del conjunto a las instancias asociadas a éste. Sin embargo, ésta no es la única suposición posible para los problemas MIL [11] y para nuestro problema no resulta ventajosa. Una alternativa fue propuesta en [6], donde se obtiene la representación de atributos de un conjunto agregando las características extraídas de las instancias asociadas a éste, y dicha representación agregada se convierte en un vector prototipo para el conjunto. La ventaja es que la red puede aprender cuáles de estas características son útiles para discriminar entre clases. En este trabajo seguimos este último esquema.

Además de ser un problema multi-instancia, el PCRY es también un problema multi-etiqueta. Un enfoque para abordar los problemas multi-etiqueta es entrenar tantos clasificadores como la cardinalidad del conjunto potencia de las etiquetas [12]. Así, un sistema podría aprender los detalles de cada posible configuración. Sin embargo, este enfoque podría conllevar a una severa falta de representantes de cada configuración, debido a su naturaleza exponencial: para nuestro problema particular implicaría entrenar $2^9 = 512$ clasificadores utilizando solamente 2000 vectores de atributos donde no todas las configuraciones están presentes y por lo tanto no pueden ser aprendidas. Un enfoque más simple es convertir el problema multi-etiqueta en un problema de una sola etiqueta [13]. Esto puede lograrse por medio de un sistema de n clasificadores binarios 'aplica' / 'no aplica', uno por cada etiqueta disponible. Este último enfoque es el que seguimos en este trabajo.

La transferencia de extractores de atributos de alto nivel, aprendidos sobre bases de datos a gran escala como ImageNet, pueden ser útiles cuando se aborda un problema

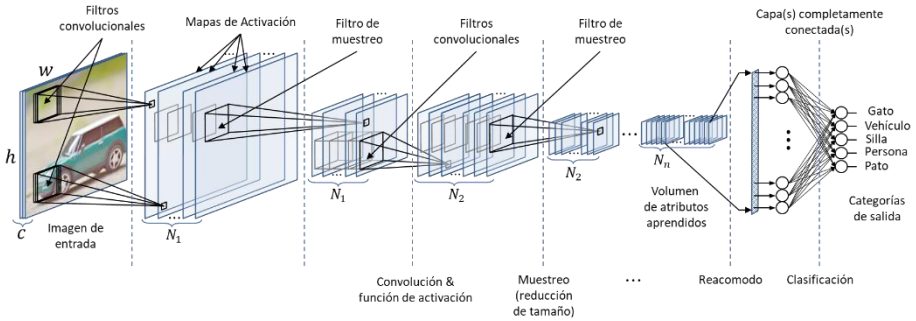

Fig. 2. Estructura de una Red Neuronal Convolucional para clasificación de imágenes.

similar de reconocimiento visual que presenta una cantidad limitada de datos de entrenamiento [6]. Uno de los primeros y más exhaustivos trabajos sobre la transferencia de parámetros aprendidos de una CNN es [14], donde los autores encontraron que la degradación del desempeño entre la tarea original y la nueva tarea esta dictada por la disimilitud entre ellas. Adicionalmente, llegaron a la conclusión de que los extractores de bajo nivel (primeras capas convolucionales) son genéricos en las tareas de visión artificial, mientras que los extractores de alto nivel (últimas capas convolucionales y capas completamente conectadas) son específicos a la base de datos de entrenamiento. En este trabajo, exploramos el uso de extractores de atributos previamente aprendidos, entrenados en 3 bases de datos distintas que consideramos son similares y complementarias a la base de datos del PCRY.

3 Propuesta

En esta sección se describe nuestra propuesta paso por paso; ésta trabaja bajo la suposición de que, usando diferentes bases de datos de entrenamiento, algunas de las características de alto nivel extraídas podrían ser más adecuadas para clasificar algún atributo particular de un restaurante. Por ejemplo, la etiqueta *outdoor seating* podría ser predicha con mayor precisión utilizando atributos aprendidos de una base de datos para clasificación de escenas; lo mismo se puede decir de las etiquetas *good for lunch* y *good for dinner* con respecto a una base de datos de alimentos. Nuestra metodología consiste en los siguientes pasos:

1. Aprovechamos las capacidades de representación de las arquitecturas CNN previamente entrenadas, al utilizar la técnica de transfer-learning sobre sus pesos. Para esta tarea decidimos utilizar la red VGG debido a que es la arquitectura CNN estándar con una precisión competitiva en las tareas de reconocimiento visual a gran escala (2^{do} lugar en la competencia ILSVRC 2014 [3]). La estructura de una CNN y una vista general de sus elementos se ilustran en la Figura 2. La red VGG-16 contiene 3 capas completamente conectadas (FC, por sus siglas en inglés), que comúnmente son referidas como: FC6, FC7 y capa de predicciones.

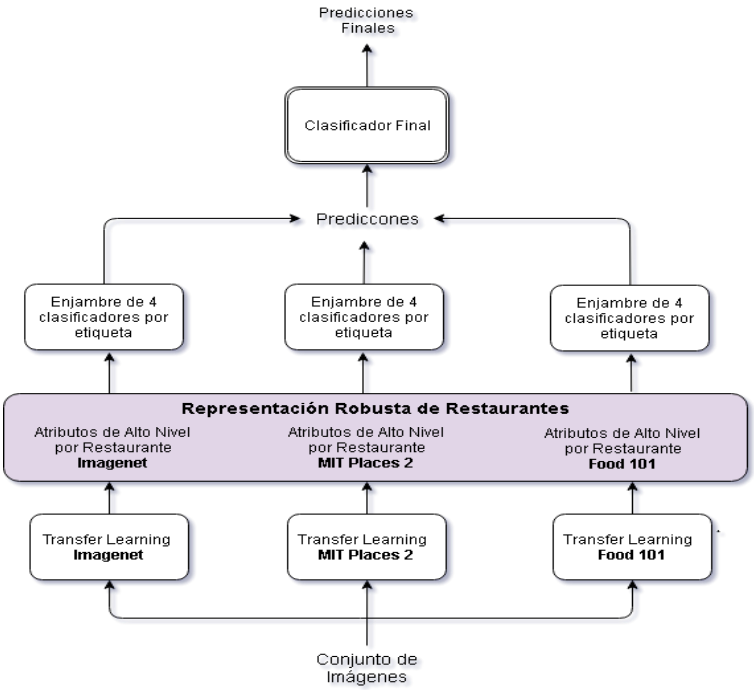

Fig. 3. Modelo propuesto para la clasificación de restaurantes.

2. Una vez cargados los pesos pre-entrenados, extraemos los atributos de alto nivel para cada imagen en la capa FC6. Escogimos la capa FC6 (atributos más específicos) en vez de la última capa convolucional (atributos más genéricos) como una forma de regularización, considerando que una gran proporción de los parámetros de la red VGG-16 conectan la última capa convolucional con la capa FC6 (100 millones de parámetros). Se asume que el aprendizaje de tantos parámetros utilizando una base de datos levemente supervisada conllevaría a un sobre-ajuste.

3. La representación extraída para la j-ésima instancia (imagen) del i-ésimo conjunto (restaurante) se denota como h_{ij}. De acuerdo a esta notación, la representación agregada de un conjunto está dada por: $\hat{h}_i = f(h_{i1}, h_{i2}, \dots, h_{in})$. La función f codifica el mapeo de los atributos a nivel instancia a los atributos a nivel conjunto. En este trabajo la función promedio $\hat{h}_i = avg_j(h_{ij})$ es utilizada para obtener el vector de atributos prototipo para el i-ésimo restaurante. Esta elección sigue la suposición de que activaciones similares se encontrarán presentes en los restaurantes que compartan una etiqueta de clase.

4. Una vez que se tienen los vectores de atributos por restaurante, procedemos a aplicar fine-tuning sobre la capa FC7 y la capa de predicción. Los pesos en la capa FC7 se inicializan con los valores óptimos aprendidos para la tarea original; la capa de

```
# Training of Representation-Dependent Binary Classifiers
for dataset ∈ {ImageNet, Places, Food-101} do:
    DBFC ← aggregated bag features, per restaurant
    shuffle the businesses index BI
    Instances n ← number of restaurants / number of folds
    for fold ∈ {0, 1, 2, 3} do:
        folds[fold] = BI[n×fold to n×(fold + 1)]
    end for
    for attribute j ∈ {0, 1, …, 8} do:
        for validation fold k ∈ {0, 1, 2, 3} do:
            for fold ∈ {0, 1, 2, 3} do:
                if fold is not k do:
                    append folds[fold] to train_idx
                end if
            val_idx ← folds[k]
            end for
            Train classifier_{j,k} on DBFC[train_idx]      // DBFC: Dataset Business FC
            ValProbs_{j,k} ← predict probabilities using val_idx
            ValClasses_{j,k} ← predict classes using val_idx
        end for
        datasetValProbs_j ← concatenate the ValProb_{j,k} on k axis
        datasetValClasses_j ← concatenate the ValClases_{j,k} on k axis
    end for
    datasetValProbs ← concatenate datasetValProbs_j on j axis
    datasetValClasses ← concatenate datasetValClasses_j on j axis
end for
Calculate validation accuracy with datasetValClasses
valProbs ← concatenate datasetValProbs on dataset
```

Fig. 4. Pseudocódigo del clasificador binario para cada representación

```
# Final classifier
valTargets ← get the restaurant target attributes
for validation fold k ∈ {0, 1, 2, 3} do:
    for fold ∈ {0, 1, 2, 3} do:
        if fold is not k do:
            append folds[fold] to train_idx
        end if
    end for
    val_idx ← folds[k]
    Train FinalClassifier_k on ValProbs[train_idx]
    Evaluate FinalClassifier_k on ValProbs[val_idx]
end for
Get the validation accuracy for the final classifier
```

Fig. 5. Pseudocódigo del clasificador final

predicción se modifica para contener una sola unidad sigmoide y así producir un clasificador binario por atributo.

5. Finalmente, ya que no toda representación es igualmente útil en la predicción de una etiqueta, implementamos un clasificador final con 9 neuronas sigmoides de salida y 27 unidades de entrada (9 por cada representación). De esta manera, añadimos el

módulo final a un sistema automático de principio a fin que evita el uso de heurísticas humanas para seleccionar los pesos apropiados de cada representación para una etiqueta dada.

Para el ajuste de hiper-parámetros se utilizó validación cruzada de 4 pliegues, que adicionalmente nos permite utilizar técnicas de aprendizaje por enjambre al entrenar 4 clasificadores por etiqueta. Las salidas de estos clasificadores se promedian para producir un vector de 9 probabilidades, una para cada uno de los 9 atributos.

Con el propósito de obtener una representación robusta de los atributos de un restaurante, los pesos aprendidos de diferentes tareas se utilizan para inicializar la arquitectura VGG-16. Las tareas elegidas son: ImageNet, con 1000 clases y más de un millón de imágenes de entrenamiento [2], MIT Places-2 con 365 clases de escenas naturales y humanas [15], y Food-101 con 101 clases de alimentos [16]. Por lo tanto, los pasos 1-4 deben repetirse por cada representación utilizada. Una representación gráfica de nuestro modelo se presenta en la Figura 3.

4 Diseño de experimentos

Los pesos entrenados de las bases de datos ImageNet y MIT Places 2 se descargaron de Caffe Model Zoo[2]. Para Food-101 no existen esos pesos, así que aplicamos *fine-tuning* sobre los pesos de ImageNet usando Food-101 hasta que la pérdida de validación convergiera.

Los folders de validación se usaron para el ajuste de híper-parámetros durante el entrenamiento de los clasificadores binarios. Se seleccionó una tasa de aprendizaje (lr) con una alta reducción de la perdida de validación durante las épocas iniciales que disminuye considerablemente conforme progresa el entrenamiento. Dado el poder de representación de la red y el potencial de sobre-ajuste, regularizamos los clasificadores con un $dropout\_rate = 0.5$ para los atributos de entrada y las activaciones FC7. Los híper-parámetros usados son: $lr = 10^{-6}$, $épocas = 100$. Estos parámetros se mantuvieron fijos para todos los clasificadores entrenados. El pseudocódigo para el entrenamiento de los clasificadores binarios se muestra en la Fig. 4 y el del clasificador final se muestra en la Figura 5.

El proceso de ajuste sobre los pesos en las capas FC es el siguiente: entrenar los pesos en la capa de predicción mientras se mantienen los pesos de FC7 fijos. Una vez que la pérdida de validación se estabiliza, los pesos de FC7 se descongelan.

El objetivo es obtener una mejor representación mediante los atributos de entrada FC6 en la capa FC7. Elegimos Adam como optimizador, pues se compara favorablemente a otros optimizadores [17]. La métrica oficial para el reto es la puntuación F1, definida para los C posibles atributos de los restaurantes basándose en las métricas de $Precisión$ y $Recall$ que se calculan como se muestra a continuación:

$$Precisión = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FP_k}, \quad (1)$$

[2] https://github.com/BVLC/caffe/wiki/Model-Zoo

$$Recall = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FN_k}, \tag{2}$$

donde TP: Verdaderos Positivos, FP: Falsos Positivos y FN: Falsos Negativos. La $F1\ Score$ es la media harmónica entre $Precisión$ y $Recall$:

$$F1\ Score = \frac{2 \times Precisión \times Recall}{Precisión + Recall}. \tag{3}$$

5 Resultados

Primero presentamos los resultados de precisión en el conjunto de validación obtenidos para cada representación de manera individual: ImageNet, MIT Places 2, Food-101 y para el clasificador final, posteriormente presentamos los resultados sobre el conjunto de evaluación, obtenidos de la página web de la competencia.

Las Figuras 6-8 muestran la precisión de validación, promediada para los 4 pliegues, durante la fase de entrenamiento; la línea solida indica pesos congelados en la capa FC7, la línea punteada indica que todos los parámetros de la red son libres de actualizarse. Además, nótese que la caja de etiquetas esta ordenada de manera descendente por precisión de validación: de arriba-abajo, izquierda-derecha.

Una de las suposiciones al usar características de entrada de más bajo nivel FC6 (en vez de FC7) es que podrían conllevar a una mejor representación en la capa oculta FC7 mediante el uso de la técnica fine-tuning; sin embargo, observamos que el uso de las características extraídas en la capa FC7 (representados por la línea solida) no afecta la precisión de validación. Conjeturamos que esto es porque el conjunto de entrenamiento no es suficientemente grande para entrenar tantos parámetros y el optimizador no induce una mejor representación.

La precisión de validación por representación y para el clasificador final se muestra en la Tabla 1. Analizando la tendencia general de las representaciones individuales observamos que la precisión de validación para Food-101 es más alta en casi todos los atributos, un fenómeno similar había sido observado en el trabajo de [14].

Debido a que aplicamos fine-tuning sobre los pesos de ImageNet para la tarea de reconocimiento de alimentos, los extractores de atributos se volvieron más robustos, incorporando conocimiento de las 1000 clases originales de ImageNet y la tarea objetivo que contenía 101 variedades de alimentos.

Otra tendencia que persiste es el menor desempeño en los clasificadores entrenados con la representación de MIT Places 2. La hipótesis es que esto es debido a la discrepancia entre las tareas: en el PCRY los conjuntos de imágenes consisten principalmente en imágenes de alimentos y raramente en escenas naturales o humanas. A pesar de que asumimos que la etiqueta '*outdoor seating'* podría hacer uso de atributos encontrados en MIT Places 2, notamos que consistentemente esta etiqueta es la más difícil de predecir debido a que contiene poca, e inclusive ambigua, información visual sobre si la etiqueta aplica o no. La predicción del atributo '*has table service*' es considerablemente mejor en todas las representaciones, en particular con las representaciones de ImageNet/Food-101. Esto tiene sentido por dos razones: en primer

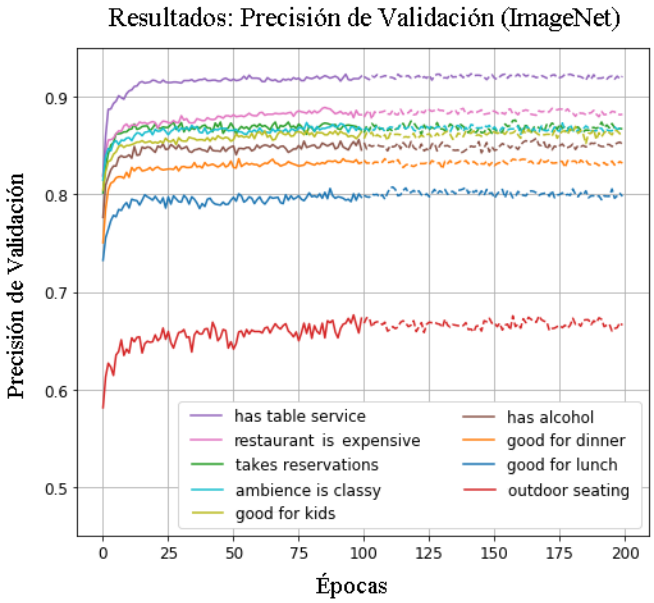

Fig. 6. Resultados (Validación) Precisión para la Representación ImageNet.

Tabla 1. Precisión de Validación para las Representaciones Individuales; el mejor desempeño se muestra en negritas.

	Representación			
Etiqueta	**ImageNet**	**MIT Places**	**Food 101**	**Clasificador Final**
Good for lunch	80.2%	79.8%	81%	**81.6%**
Good for dinner	82.6%	83.6%	**84.2%**	83.7%
Takes reservations	87.1%	86.8%	87.3%	**88%**
Outdoor seating	65.6%	65.7%	68.1%	**69.4%**
Restaurant is expensive	87.5%	87.8%	**89.2%**	89%
Has alcohol	84.6%	84.8%	85.4%	**86.4%**
Has table service	91.9%	89.3%	92%	**92.4%**
Ambience is classy	**86.6%**	85.4%	85.9%	86.5%
Good for kids	85.9%	85.8%	86.7%	**87.2%**
Desempeño	83.6%	83.2%	84.4%	**84.9%**

lugar existe una correlación consistente entre la etiqueta y la presencia de menús lo que la convierte en una clase fácil de predecir, además, la clase menú forma parte de las 1000 clases originales de ImageNet. Pertinente a esta misma idea es que aun con el fine-tuning sobre la representación original de ImageNet, sus capacidades de representación en Food-101 no se perdieron, lo que nos lleva a concluir que la capacidad de aprendizaje de VGG-16 excede a la base de datos Food-101 y por lo tanto la mayoría de los extractores de atributos mantuvieron sus pesos originales.

En los resultados para el clasificador final observamos que el desempeño por atributo, en casi todos los casos, es tan bueno como el mejor desempeño de las

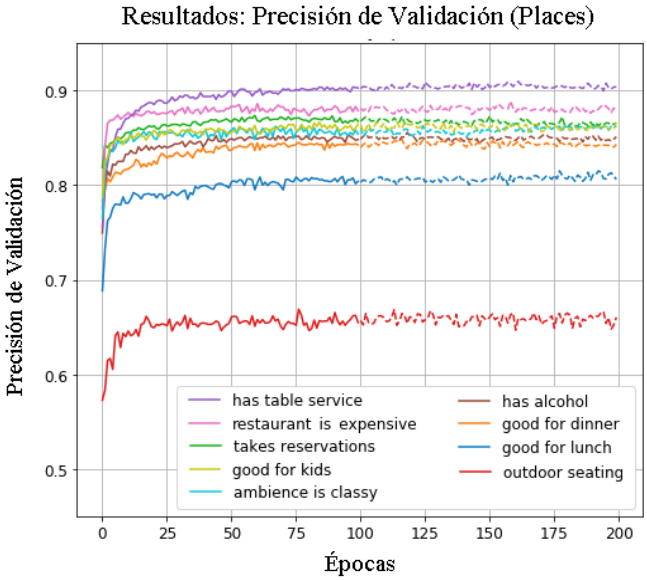

Fig. 7. Resultados (Validación) Precisión para la Representación MIT Places 2.

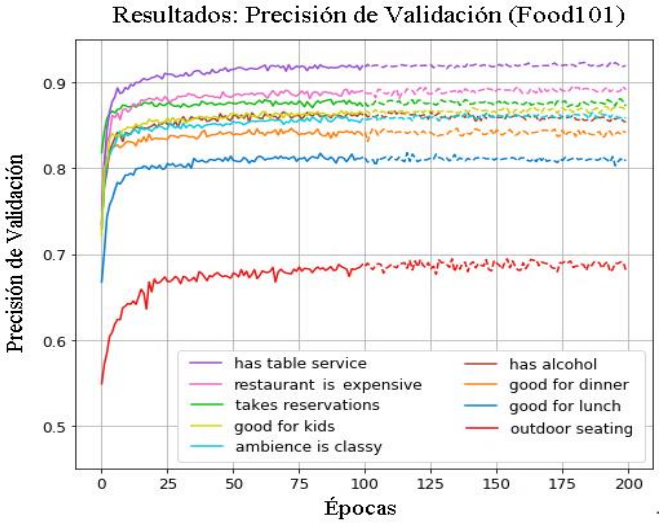

Fig. 8. Resultados (Validación) Precisión para la Representación Food-101.

representaciones individuales. De un análisis detallado de los pesos, observamos que el clasificador final esta asignando más importancia a aquella representación que presenta un mejor desempeño en la precisión de validación. La única etiqueta en la que el desempeño del clasificador final es significativamente peor es 'good for dinner'.

Pensamos que este es el caso por la fuerte dependencia del atributo respecto de las imágenes de comida, y de manera específica, de imágenes de platillos caros, como platos de porcelana (una clase que no está representada en ImageNet/MIT Places 2 pero que aparece en Food-101 como un atributo útil).

Una vez que el enjambre de clasificadores por atributo y el clasificador final fueron entrenados, aplicamos el modelo completo en los restaurantes de prueba, y enviamos las predicciones a la página de la competencia.

Dos resultados fueron obtenidos: una puntuación F1 pública de 0.8091 (en 30% de los datos de prueba), y una puntuación F1 privada de 0.8177 (en 70% de los datos de prueba). Este resultado coloca a nuestro modelo en el 10% superior de los competidores, donde la mejor puntuación fue de 0.83177. La principal ventaja de nuestro sistema es su simplicidad, apoyada por un uso uniforme de redes neuronales que son sistemas automatizados de principio a fin. El resultado final es un sistema modular y de propósito general para abordar tareas de clasificación multi-instancia y multi-objetivo.

6 Conclusiones

Observamos que la representación de características por restaurantes, obtenida mediante la agregación de características de alto nivel de su conjunto de imágenes, es útil cuando se aborda un problema multi-instancia debido a la conformidad con la suposición de que ciertas características con magnitud similar aparecen para restaurantes con una misma etiqueta de clase. La representación de alto nivel influencia de manera directa el desempeño de cada clasificador de atributos, esto se ejemplifica claramente con la etiqueta '*has table service'*, donde los pesos aprendidos mediante la base de datos original extraen atributos de objetos que están fuertemente correlacionados (a saber, los menús) con la etiqueta. La etapa de fine-tuning adicional con la base de datos Food-101, utilizando pesos pre-entrenados con ImageNet, resultó útil para mejorar el desempeño de la representación, debido a su habilidad para la discriminación de las clases en ambas bases de datos.

Mientras más pesos de redes CNN pre-entrenadas se hagan disponibles esperamos lograr un mejor desempeño utilizando modelos de principio a fin que extraen el desempeño óptimo por representación.

Aunque se necesita más evidencia y argumentación, consideramos que la presente metodología se podría extender y aplicarse en otras tareas de clasificación. Como trabajo futuro contemplamos la inclusión de un esquema de aumentación de datos con la finalidad de eludir uno de los factores principales que limitaron el desempeño de nuestro modelo.

Agracecimientos. Este trabajo se llevó a cabo gracias al auspicio del Consejo Nacional de Ciencia y Tecnología (CONACYT) de México, a través de los apoyos 604421 (J. Veloz) y CÁTEDRAS-2598 (A. Rojas). Los autores agradecen a Yelp por hacer la base de datos PCRY disponible públicamente y a Kaggle por hospedar la competencia correspondiente.

Referencias

1. LeCun, Y., Bengio, Y.; Hinton, G.: Deep learning. Nature Research, 521(7553), pp. 436–444 (2015)
2. Russakovsky, O., Deng, J., Su, H., Krause, J., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3), pp. 211–252 (2015)
3. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
4. Szegedy, C., Liu, W., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv: 1512.03385 (2015)
6. Wu, J., Yu, Y., Huang, C., Yu, K.: Deep multiple instance learning for image classification and auto-annotation. In: Proceedings of the IEE Conference on Computer Vision and Pattern Recognition, pp. 3460–3469 (2015)
7. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In: Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, pp. 1717–1724 (2014)
8. Baan, J: A Deep Learning Ensemble approach to the Yelp Restaurant Classification Challenge. (2016)
9. Dietterich, T. G., Lathrop, R. H., Lozano-Pérez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89(1), pp. 31–71 (1997)
10. Foulds, J., Frank, E.: A review of multi-instance learning assumptions. The Knowledge Engineering Review, 25(1), pp. 1–25 (2010)
11. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: An ensemble method for multilabel classification. In: Proceedings of the 18th European Conference on Machine Learning (ECML`07), pp. 406–417 (2007)
12. Tsoumakas, G., Katakis, I., Vlahavas, I.: Data mining and knowledge discovery handbook. (2009)
13. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks?. In: Ghahramani, Z., Welling, M., et al. (eds), Advances in neural information processing systems, pp. 3320–3328 (2014)
14. Zhou, B., Lapedriza, A., Khosla, A, Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)
15. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101: Mining discriminative components with random forests. In: European Conference on Computer Vision (2014)
16. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)