Clasificación de objetos en imágenes usando SIFT

Miriam Mónica Duarte Villaseñor y Leonardo Chang Fernández

Proyecto del Curso Modelos Gráficos Probabilistas y sus aplicaciones
Maestría en Ciencias Computacionales, INAOE.
{mduarte, lchang}@ccc.inaoep.mx

Resumen. En este trabajo se propone el uso de las características SIFT para resolver problemas de reconocimiento de clases de objetos en imágenes. El método propuesto realiza agrupamiento sobre los descriptores de los puntos detectados con el fin de caracterizar la apariencia de las clases. Luego, en la fase de clasificación se utiliza una Red Bayesiana de 3 niveles. Los experimentos realizados sobre la base de datos PASCAL muestran cuantitativamente que las características locales SIFT son apropiadas para representar clases de objetos. Además los mismos también muestran la robustez del método obtenido ante variaciones de iluminación, escala, rotación y oclusión.

Palabras Claves: clasificación de objetos, características locales, SIFT, agrupamiento, redes bayesianas.

1 Introducción

En los últimos años, las características locales (*local features*, en inglés) han demostrado ser muy efectivas en la búsqueda de características o rasgos distintivos entre diferentes vistas de un escenario.

La idea tradicional de estos métodos es primero detectar estructuras o puntos significativos en la imagen y obtener una descripción discriminante de estas estructuras a partir de sus alrededores, la cual será utilizada luego para la comparación usando una medida de similitud entre estos descriptores.

Un detector de puntos de interés es diseñado para encontrar el mismo punto en diferentes imágenes incluso si el punto está en distintas posiciones y escalas. Distintos métodos han sido propuestos en la literatura. Un estudio y comparación de estos se presenta en [13].

Uno de los métodos de características locales más popular y ampliamente utilizado es el SIFT (del inglés: *Scale Invariant Features Transform*) propuesto por Lowe [7]. Las características extraídas por el SIFT son en gran medida invariantes a la escala, rotación, cambios de iluminación, ruido y pequeños cambios de pose o perspectiva. Los descriptores basados en SIFT han mostrado mejores resultados que otros descriptores locales [10].

En un inicio, la utilización del SIFT y de los métodos de características locales en general en problemas de reconocimiento de objeto se resumía a detectar un objeto específico dentro de una escena (Ver Figura 1).

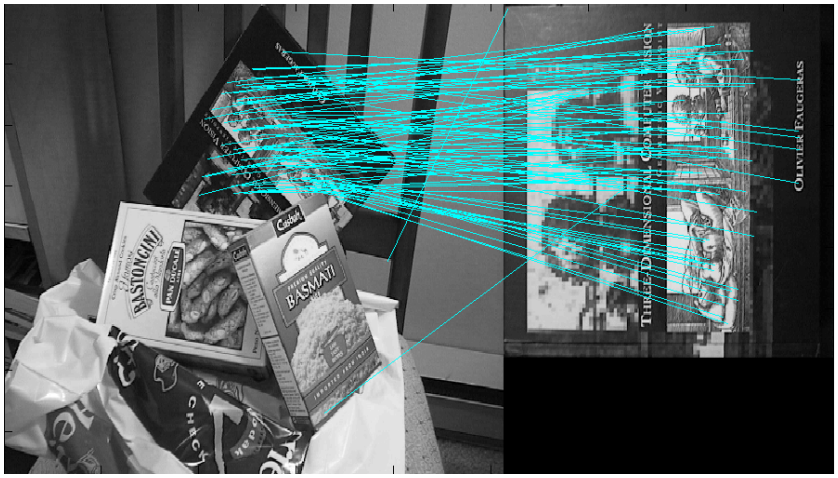

Fig. 1. La utilización de los descriptores SIFT permite de manera directa poder encontrar un objeto específico en una escena. Se puede apreciar su robustez ante rotación, oclusión y escala.

Como objetivo general de este proyecto se propone obtener un método que permita reconocer clases de objetos usando el SIFT.

En otras palabras lo que se propone es lograr detectar en una escena, más que un objeto en específico, todos los objetos de una clase. Por ejemplo detectar cualquier avión, o cualquier cara, motocicleta, gato, hoja, etc.

2 Trabajos Relacionados

La mayoría de los métodos de reconocimiento de clases de objetos basados en apariencia caracterizan los objetos por su apariencia global, usualmente la imagen completa [12] [11]. Estos métodos no son robustos a la oclusión ni a variaciones como rotación o escala. Más aún, estos métodos son aplicables solamente a objetos rígidos. Las características locales se han vuelto muy populares con el fin darle solución a las limitaciones de estos métodos en tareas de detección y reconocimiento de clases de objetos.

Dentro del estado del arte del reconocimiento de clases de objetos, muchos métodos usan agrupamiento como un nivel intermedio de representación [1][6][2][8].

Algunos trabajos han sido reportados en la literatura con el fin de reconocer clases de objetos usando características locales. En [2] para la detección de las regiones usan el detector de Harris-Laplace [9] y el de Kadir y Brady [5]. Estas regiones son descritas usando el descriptor SIFT [7]. En este trabajo Dorkó y Schmid realizan agrupamiento sobre los descriptores con el fin de caracterizar la apariencia de las clases. Luego contruyen clasificadores de partes más

pequeñas de los objetos a partir de los clusters formados, eliminando varios de ellos quedándose solamente con los más discriminativos. En la fase de clasificación cada uno de estos clasificadores es evaluado, permitiendo tomar una decisión sobre la clase del objeto.

En [8] Mikolajczyk y colaboradores evalúan el desempeño de varios métodos de características locales en la tarea de reconocimiento de clases de objetos. Los detectores de regiones invariantes evaluados fueron Harris-Laplace, SIFT, Hessian-Laplace y MSER. Los descriptores de características evaluados fueron los descriptores SIFT, GLOH, PCA-SIFT, Momentos y Correlation Cruzada. En este trabajo los autores evalúan varias combinaciones detector-descriptor. En el mismo también se realiza agrupamiento sobre los descriptores con el fin de caracterizar la apariencia de las clases. Para la clasificación de una nueva muestra, los descriptores extraídos son macheados con los clusters obtenidos y un umbral determina la pertenencia a la clase.

El método propuesto en este trabajo también realiza agrupamiento sobre los descriptores de los puntos extraídos de las imágenes de entrenamiento al igual que los trabajos anteriormente explicados. Se utiliza el detector y el descriptor SIFT. La principal diferencia respecto a estos métodos radica en el uso de una Red Bayesiana en la fase de clasificación. Además de que se presenta una experimentación más profunda para medir su comportamiento ante variaciones de iluminación, escala, pose y oclusión.

3 Descriptores de puntos claves SIFT

Las primeras etapas del algoritmo SIFT encuentran las coordenadas de los puntos claves en determinada escala y a cada punto le asignan una orientación. Los resultados de estas etapas garantizan invariabilidad a localización en la imagen, escala y rotación. Luego se calcula un descriptor para cada punto clave. Este descriptor debe ser altamente distintivo y parcialmente robusto a otro tipo de variaciones como iluminación y perspectiva 3D.

Lowe para crear su descriptor propone obtener un arreglo de 4×4 histogramas de 8 bins. Estos histogramas se calculan a partir de los valores de orientación y magnitud en una región de 16×16 pixeles alrededor del punto de modo tal que cada histograma se forma de una subregión de 4×4. En su parte izquierda la Figura 2 muestra los valores de orientación y magnitud del gradiente alrededor del punto. En la parte derecha se muestran los histogramas para cada subregión donde la longitud de cada flecha corresponde a la suma de magnitudes del gradiente cerca de esa orientación.

El descriptor consiste en un vector resultado de la concatenación de estos histogramas. Dado que son $4 \times 4 = 16$ histogramas de 8 bins cada uno, el vector resultante es de tamaño 128. Este vector es normalizado en aras de lograr invariabilidad a cambios de iluminación.

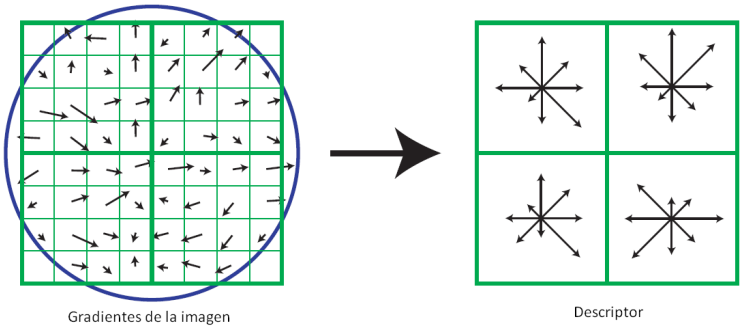

Fig. 2. Esta figura muestra un descriptor de 2×2 calculado a partir de una región de 8×8 pixeles alrededor del punto clave. En este trabajo se utilizan descriptores de 4×4 en regiones de 16×16.

4 Aprendizaje de clases de objetos usando características SIFT

Se desea obtener un modelo capaz de generalizar más allá de cada objeto dentro del conjunto de entrenamiento y que permita el aprendizaje de un modelo general de la clase. Además el aprendizaje debe ser posible a partir de un pequeño número de muestras. Con este fin y en concordancia con varios trabajos reportados en la literatura que fueron mencionados en la Sección 2 se realiza agrupamiento sobre los descriptores locales de las imágenes.

La Figura 3 muestra un esquema a alto nivel del método propuesto, el cual es detallado a continuación:

1. Para cada imagen de entrenamiento se extraen sus características SIFT.
2. Luego se realiza agrupamiento sobre los descriptores extraídos.
3. Cada descriptor en cada cluster es etiquetado con su clase de pertenencia.

Se espera que estos clusters tengan alta precisión, o sea, que cada cluster sea representativo de una clase solamente. En la práctica se demuestra que esto no siempre sucede por lo existirán clusters que son compartidos por varias clases. Métodos adicionales serán necesarios en la etapa de clasificación para resolver estas ambigüedades.

4.1 Agrupamiento de descriptores SIFT

Para construir los clusters de descriptores se usa agrupamiento aglomerativo jerárquico propuesto por [4]. Este algoritmo no depende de la inicialización a diferencia de K-means o EM-clustering. Además esta reportado como superior a K-means en [3].

Dados F descriptores extraídos de cada imagen de entrenamiento el agrupamiento es inicializado con F clusters cada uno conteniendo 1 descriptor solamente. En cada iteración los dos clusters de mayor cohesión son unidos.

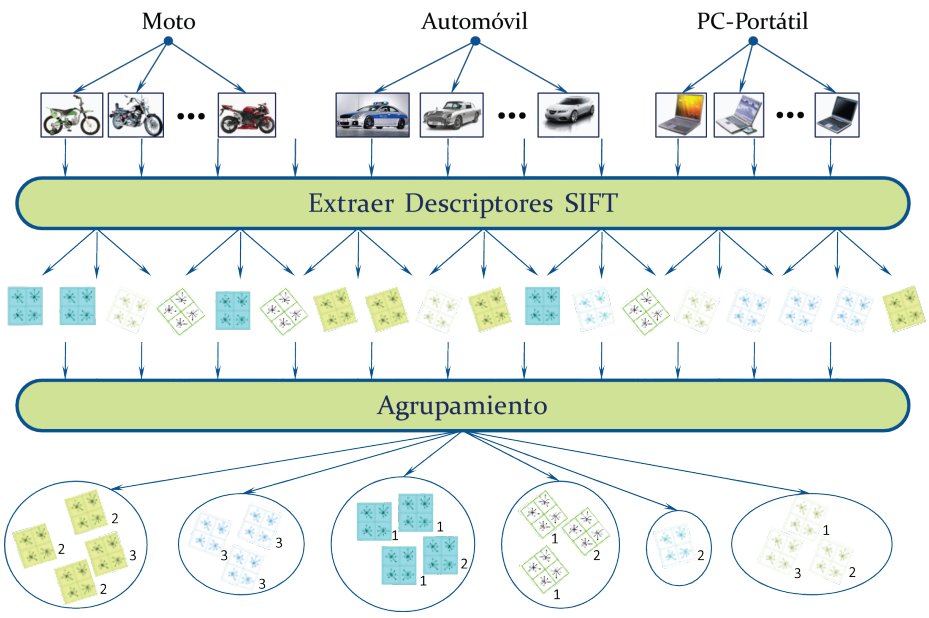

Fig. 3. A partir de varias muestras de cada clase se agrupan los descriptores SIFT extraídos de cada imagen.

La similaridad entre dos clusters cualesquiera puede ser medida de varias maneras, las más comunes son vinculación simple, vinculación completa y vinculación media. En este trabajo se usa vinculación media, la cual se define como la distancia media de todo elemento de un cluster a todo elemento del otro cluster:

$$D(k,l) = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} d(K_{km}, K_{ln}), \tag{1}$$

donde M y N son la cantidad de descriptores en los clusters k y l respectivamente.

El agrupamiento aglomerativo produce una jerarquía de uniones de clusters hasta que el criterio de parada detiene el proceso. Por lo tanto, se obtienen clusters donde la similaridad entre cada par de clusters está por encima de un valor dado. Este valor se usa como umbral para determinar el criterio de parada.

Las principales desventajas del uso de estos métodos radican en que son poco escalables ya que la complejidad en tiempo y memoria es $O(F^2)$ y además de que no existe la posibilidad de relocalizar elementos que hayan sido agrupados de manera incorrecta.

5 Clasificación de objetos

Dada una nueva muestra la clasificación se lleva a cabo primero extrayendo las características SIFT de la imagen de entrada. Luego para cada una de estas características se halla a que cluster del modelo aprendido corresponden y a partir de estos se determina la clase de la imagen de entrada. En la Figura 4 se muestra un diagrama a alto nivel del método propuesto.

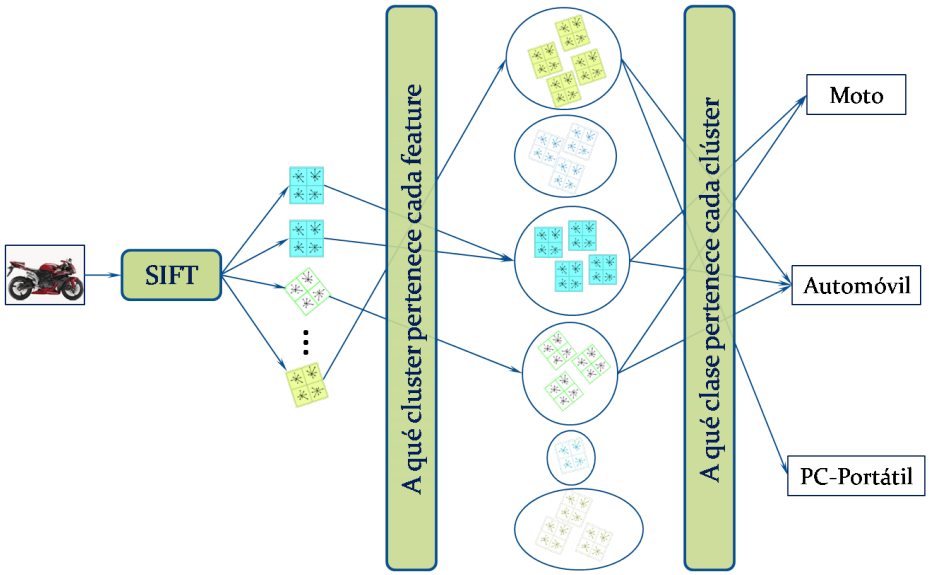

Fig. 4. Esquema de la clasificación para una nueva imagen. Las características SIFT de esta imagen son extraídas y para cada una de ellas identificados los cluster del modelo aprendido al que pertenecen. La clase de la imagen es la clase mayoritaria en estos clusters.

Esta idea puede ser representada como una Red Bayesiana (RB) de 3 niveles. La representación gráfica de esta RB se muestra en la Figura 5. En el primer nivel se tienen los descriptores extraídos para el objeto que se desea clasificar representados por los nodos $f_1, f_2, ..., f_F$ donde F es la cantidad de características extraídas de la imagen. En el segundo nivel se tienen los clusters obtenidos en la fase de entrenamiento representados por $K_1, K_2, ..., K_N$ donde N es el número de clusters obtenidos. Finalmente, en el tercer nivel se tienen las clases de los objetos entrenados representados por $C_1, C_2, ..., C_M$ donde M es la cantidad de clases.

Usando este modelo, la clasificación de una nueva imagen I se realiza de la siguiente manera:

1. Extraer características SIFT de I.

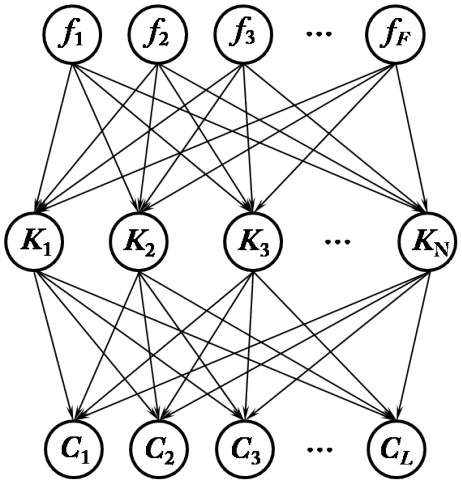

Fig. 5. Representación gráfica de la RB de 3 niveles utilizada para la clasificación de uno nuevo objeto.

2. Para cada descriptor f extraído de I obtener el cluster k_f al que pertenece. Se selecciona el cluster cuya probabilidad de pertenencia sea mayor. Esta probabilidad es función de la distancia del descriptor al cluster la cual es normalizada por la distancia entre los dos clusters más lejanos. Se utiliza la misma distancia D empleada para realizar el agrupamiento la cual fue definida en la Ecuación 1:

$$k_f = \arg\max_i P(f|K_i)P(K_i), \text{ donde}$$

$$P(f|K_i) = 1 - \frac{D(f, K_i)}{\max_{kl} D(K_k, K_l)}$$

3. Para cada cluster $k_{f_1}, k_{f_2}, ..., k_{f_F}$ seleccionado en la etapa anterior (nótese que varias de las características detectadas pueden estar en un mismo cluster) se obtiene la probabilidad de cada clase dada dicha evidencia, esta probabilidad es extraída del modelo entrenado, propagándose además la probabilidad obtenida en la etapa 2.
4. Finalmente se selecciona como la clase del objeto aquella cuya suma de probabilidades de ocurrencia dado cada uno de los cluster seleccionados en la etapa 2 sea mayor:

$$c^* = \arg\max_i \sum_f P(C_i|k_f)P(k_f)$$

6 Experimentos y Resultados

En esta sección se describen los experimentos realizados al método propuesto y se analizan los principales resultados obtenidos.

6.1 Entrenamiento

Para realizar las pruebas se obtuvo un conjunto de imágenes de un repositorio en internet en la siguiente liga:"http://pascallin.ecs.soton.ac.uk/challenges/VOC /databases.html", en ésta base de datos se tiene un total de 102 clases y distintos número de imágenes por cada clase, el formato de las imágenes es JPG y el tamaño promedio de cada imagen es de 300×300 pixeles. Esta base de datos pertenece a la comisión europea PASCAL, que tiene como objetivo la distribución de métodos de análisis de patrones y aprendizaje; como base para tecnologías de interfaces multi-modales, y es utilizada para trabajos que realicen éste tipo de tareas.

La fase de entrenamiento del método propuesto es costosa respecto al uso de memoria RAM, por lo que se realizó el entrenamiento con tan sólo 20 imágenes de cada clase. El método de agrupamiento aglomerativo jerárquico necesita una matriz de disimilaridad, que es una matriz cuadrada. Dado que se tienen 4 clases y 20 imágenes por clase, que suman 80 imágenes para el entrenamiento, por cada imagen se obtiene en promedio 500 descriptores (cada descriptor es un vector de dimensión 128), teniendo así $80 \times 500 = 4000$ descriptores y como la matriz es cuadrada se tiene en total $1,600,000,000$ descriptores. A pesar de contar con 3Gb de memoria RAM, para realizar el entrenamiento, estos cálculos ocasionan que la memoria se llene y si se aumenta el número de imágenes, entonces se desborda la memoria RAM.

Se eligieron 4 clases al azar, al igual que las imágenes para realizar el entrenamiento y las clases fueron:

- Clase 1: Cámaras fotográficas.
- Clase 2: Billetes de dólar.
- Clase 3: Motocicletas.
- Clase 4: Relojes de pulsera.

En la Figura 6 se muestran las imágenes con las que se realizó el entrenamiento.

6.2 Pruebas del sistema

Se realizaron 2 experimentos, en el Experimento 1 se probó el método con 100 imágenes por cada clase, éstas imágenes tienen poca variación en oclusión, escala, iluminación, y rotación. En el Experimento 2 se obtuvo una muestra aleatoria de 10 imágenes para cada clase, y cada imagen sufre agresiones en iluminación, oclusión, rotación y escala, con el fin de probar la robustez del sistema ante estos cambios. A continuación se detallan los resultados.

Experimento 1 El sistema recibe el conjunto de imágenes a clasificar, las imágenes se deben etiquetar con el número de la clase a la que pertenecen, así, las cámaras comienzan con un 1, las de dólar con 2, las motos con 3 y las de reloj con 4. Una vez que el sistema obtiene los descriptores y clasifica la imagen, revisa

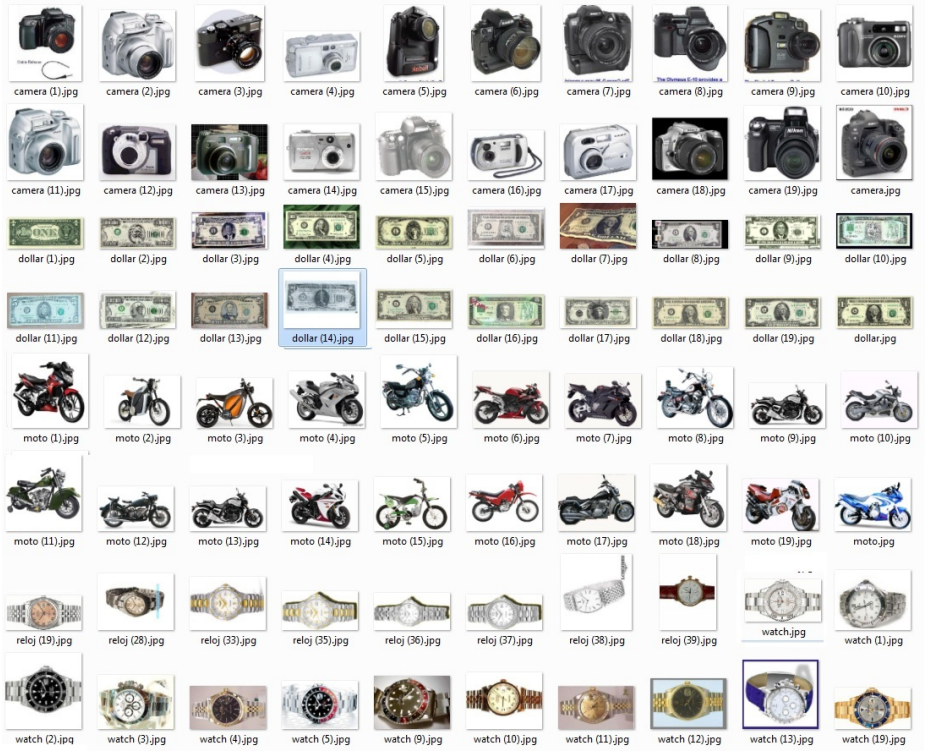

Fig. 6. Imágenes del entrenamiento.

si coincide con el primer carácter que aparece en el nombre de la imagen y si no es el mismo, entonces coloca una copia de la imagen en una carpeta, colocando después de el primer carácter un guión y el número con el que se clasificó, para que podamos identificar claramente las imágenes que se han clasificado mal.

Debido a que las clases no contenían suficientes elementos para realizar las pruebas, se descargaron el resto de las imágenes de internet; con el fin de tener las imágenes completas y las clases equilibradas. En la Tabla 1 se muestran los resultados obtenidos por el sistema de clasificación propuesto, para 100 imágenes por cada clase.

Para interpretar los datos de la Tabla 1 obtuvimos las matrices de confusión para cada clase; una matriz de confusión es formada como se muestra en la Tabla 2. Donde el total de elementos por clases está dado por N, y $N = TP+FN+FP+TN$. La precisión es igual a $Precision = TP/N$ y mide que porcentaje clasificado es correcto. El recuerdo se expresa como $Recuerdo = TP/(TP+FN)$, y calcula cuantos elementos de todos los que debería haber encontrado, se clasificaron bien. TrueNegativeRate o Specificity, está medida obtiene el porcentaje de que cuantos ejemplos negativos son verdaderamente negativos y se obtiene mediante

Tabla 1. Objetos clasificados para cada una de las clases con el método propuesto para el Experimento 1.

	CLASE			
Imagen	Cámara	Dólar	Motos	Reloj
Cámara	87	7	6	0
Dólar	0	99	1	0
Motos	2	2	95	1
Reloj	3	3	3	91

$TNR = TN/(FN+FP)$. Accuracy, calcula el porcentaje de que las predicciones sean correctas y se calcula con la formula $accuracy = (TP + TN)/N$.

Tabla 2. Elementos de una matriz de confusión

Clase	
TP = verdaderos positivos	FN = falsos negativos
FP = falsos positivos	TN = verdaderos negativos

En la Tabla 3 se muestran las matrices de confusión para cada una de las clases, que fueron obtenidos a partir de la Tabla 1.

Tabla 3. Matriz de confusión para cada clase

Cámara		Dólar		Moto		Reloj	
87	5	99	12	95	10	91	1
13	295	1	288	5	290	9	299

Tabla 4. Medidas de evaluación del clasificador propuesto

	Recuerdo %	Precisión %	TrueNegativeRate %	Accuracy %
Cámara	84	94.6	98.3	93.5
Dólar	100	89.2	96.0	95.0
Moto	99	90.5	96.7	95.0
Reloj	89.0	98.9	99.7	94.5
Promedio	90.7	93.3	97.6	94.5

En este experimento se tiene una precisión en promedio del 93% un recuerdo del 90%, es decir que casi todas las imágenes son correctamente clasificadas.

Experimento 2 Se descargaron de Internet 40 imágenes por cada clase para formar un conjunto de prueba en las cuales se tuvieran problemas de iluminación, rotación, oclusión, y escala. Algunas imágenes fueron alteradas con el programa GIMP 2.4. Para aplicar cambios de iluminación para cada imagen se le aplicó la función curves, se le quitó el 50% de brillo a la imagen original. Para la oclusión se insertaron figuras como rectángulos, círculos, estrellas, etc., ocluyendo hasta un 40% la imagen original. Para la escala se agrandó cada imagen al doble de su tamaño original y también se redujo un 50% del tamaño original. Finalmente para la rotación, se cuenta con la función Transform que realiza rotaciones, y aplicamos rotaciones de 45, 90 y 180 grados. En la Figura 7 se muestran algunas imágenes con los cambios mencionados.

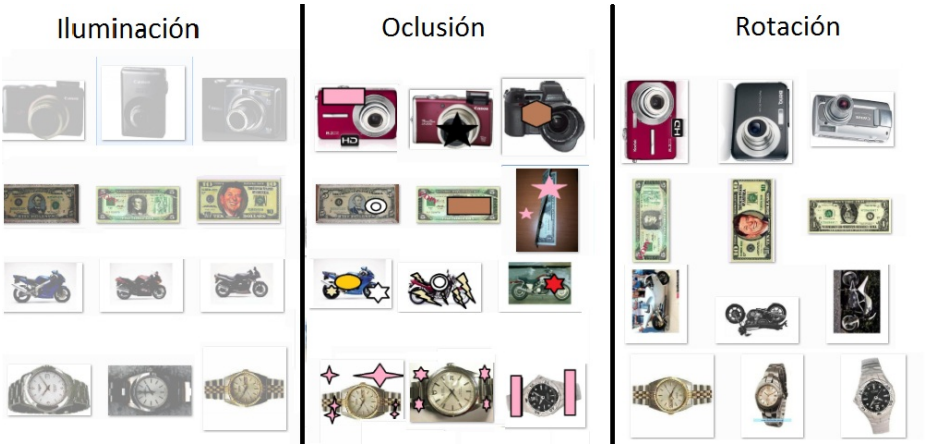

Fig. 7. Ejemplos de imágenes del conjunto de pruebas 2. Se pueden apreciar los problemas de iluminación, oclusión y rotación que presentan.

Los resultados obtenidos después de alterar las imágenes se muestran en la Tabla 5. Los resultados muestran que la clase reloj es la más afectada en oclusión, iluminación, y al cambiar el tamaño de cada imagen de esta clase.

Tabla 5. Resultados del Experimento 2.

	Oclusión		Iluminación		Tamaño-Doble		Tamaño-Mitad		Rotación	
Clase	bien%	mal%	bien%	mal%	bien%	mal%	bien%	mal%	bien%	mal%
Cámara	100	0	90	10	100	0	90	10	100	0
Dólar	100	0	100	0	100	0	100	0	100	0
Moto	100	0	100	0	100	0	100	0	100	0
Reloj	80	20	80	20	80	20	70	30	100	0

Se obtuvo la matriz de confusión para cada clase y se calcularon las medidas de evaluación que se muestran en la Tabla 6.

Tabla 6. Medidas de evaluación del Experimento 2.

	Recuerdo %	Precisión %	TrueNegativeRate %	Accuracy %
Cámara	95.0	92.7	97.5	96.8
Dólar	100	100	100	100
Moto	95.0	90.4	96.6	96.2
Reloj	97.3	90.0	96.7	96.8
Promedio	96.8	93.3	97.7	97.5

En la Tabla 6, se nota que se conserva un alto porcentaje en promedio en todas las medidas de evaluación del clasificador, y esto demuestra que el sistema propuesto es robusto ante la oclusión, iluminación, escala y rotación.

6.3 Evaluación del agrupamiento para la representación de clases de objetos

Con fin de evaluar la mejora que introduce el agrupamiento de los descriptores en la representación de clases de objetos en esta sección comparamos el método propuesto en este trabajo con un método directo de clasificación usando también características SIFT al que llamaremos BASELINE. Este método es descrito a continuación.

1. Extraer características SIFT de cada imagen del conjunto de entrenamiento.
2. Para una nueva imagen I extraer su características.
3. Esta imagen es macheada con cada una de las imágenes del conjunto de entrenamiento. El método de macheo utilizado es el propuesto por Lowe en [7].
4. La clase de la imagen de entrada será aquella que reciba mayor cantidad de correspondencias con la imagen I.

Para realizar este experimento se utilizaron las mismas imágenes de entrenamiento que en los Experimentos 1 y 2. La clasificación fue realizada con el mismo conjunto de datos que el Experimento 1. En la Tabla 7 se muestran los resultados para ese conjunto de imágenes.

Es notable que la clase de las motos está muy bien clasificada ya que tiene de 100 imágenes encuentra 95 sin embargo, las demás clases cuando se equivocan clasifican a los objetos, en la mayoría de los casos en clase motos; esto nos indica que la clase motos contiene los descriptores más similares a los descriptores de las demás clases y que los descriptores por sí solos usados con este tipo de esquemas no tienen gran poder de generalización.

Tabla 7. Resultados de la clasificación usando el método BASELINE.

	CLASE			
Imagen	Cámara	Dólar	Motos	Reloj
Cámara	35	0	63	2
Dólar	0	79	18	3
Motos	4	0	95	1
Reloj	1	0	36	63

Tabla 8. Evaluación del método BASELINE.

	Recuerdo %	Precisión %	TrueNegativeRate %	Accuracy %
Cámara	35.0	87.5	98.3	82.5
Dólar	79.0	100	100	94.7
Moto	95.0	95.0	61.0	69.5
Reloj	63.0	91.3	98.0	89.2
Promedio	68.0	80.9	89.3	84.0

Después de obtener las matrices de confusión para cada clase, se realizaron los cálculos para encontrar la precisión, el recuerdo, el True Negative Rate y el accuracy de este clasificador. Los resultados se expresan en la Tabla 8.

De la Tabla 8 y la Tabla 4 se obtiene la Tabla 9 en donde se comparan los resultados del método BASELINE y el Experimento 1. Ambos experimentos fueron realizados sobre el mismo conjunto de entrenamiento y prueba.

Tabla 9. Comparación entre BASELINE y Experimento 1.

	BASELINE	Método propuesto
Recuerdo %	68.0	90.75
Precisión %	80.9	93.3
TrueNegativeRate %	89.3	97.6
Accuracy %	84.0	94.5

Con el método propuesto se obtiene un recuerdo del 90% que es superior al 68% que se obtiene sin agrupar los descriptores; el recuerdo indica de los 100 objetos que se debían de clasificar cuantos fueron correctamente clasificados. Para la precisión se obtiene un 93% que es superior al 80.9% que se obtuvo sin realizar agrupamiento, la precisión indica el porcentaje de elementos que fueron clasificados como pertenecientes a la clase con respecto al número de elementos que se debieron tomar en cuenta. El true negative rate mide el porcentaje de los elementos que no pertenecen a la clase, con respecto a los elementos que no pertenecen y con los que se equivocó en clasificar, es decir, de todos los que debió de clasificar como no pertenecientes en verdad los pudo distinguir; para

el método propuesto se obtiene un 97.6% que indica que muy pocas veces se ha equivocado,estos elementos son los falsos positivos, y el método que no utiliza agrupamiento tiende a equivocarse más ya que tiene obtiene 89.3%. El accuracy mide el porcentaje de elementos que son correctamente clasificados, tanto para los elementos que pertenecen a cada clase como a los que se ha clasificado como no pertenecientes a esa clase, y el método propuesto supera en un 10% al método que no utiliza agrupamiento.

7 Conclusiones

Como resultado de este trabajo se obtuvo un método para la clasificación de objetos usando características SIFT. El método propuesto realiza agrupamiento sobre los descriptores de los puntos detectados con el fin de caracterizar la apariencia de las clases. Los experimentos mostraron que estas características son apropiadas para representar clases de objetos. Además de mantener invariabilidad antes cambios de iluminación, escala, rotación y oclusión. Las pruebas realizadas mostraron valores de precisión promedios por encima del 90%. En este trabajo también se mostró la aplicación de Redes Bayesianas a problemas de reconocimiento de clases de objetos, mejorando la clasificación.

7.1 Trabajo futuro

Como trabajo futuro se propone usar alguna extensión del método de clustering utilizado que permita reducir la complejidad en espacio en memoria de cuadrático a lineal, para de esta manera poder incluir más clases y más imágenes por clase en el conjunto de entrenamiento.

También se propone utilizar otro detector y/o descriptor que permita obtener un modelo que mejor generalice a la clase.

Referencias

1. Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1475–1490, 2004.
2. Gyuri Dorkó and Cordelia Schmid. Object class recognition using discriminative local features. Technical report, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005.
3. Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.
4. S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 2:241–254, 1967.
5. Timor Kadir and Michael Brady. Scale, saliency and image description. *International Journal of Computer Vision*, 45(2):83–105, 2001.
6. Bastian Leibe, Edgar Seemann, and Bernt Schiele. Pedestrian detection in crowded scenes. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 878–885, Washington, DC, USA, 2005. IEEE Computer Society.

7. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
8. Krystian Mikolajczyk, Bastian Leibe, and Bernt Schiele. Local features for object class recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1792–1799, Washington, DC, USA, 2005. IEEE Computer Society.
9. Krystian Mikolajczyk and Cordelia Schmid. Indexing based on scale invariant interest points, 2001.
10. Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005.
11. Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *Int. J. Comput. Vision*, 38(1):15–33, 2000.
12. Kah-Kay Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:39–51, 1998.
13. Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.