# Clasificación mediante Memoria Temporal Jerárquica

*Fabián Fallas-Moya*
*Sede del Atlántico*
*Universidad de Costa Rica Cartago,*
*Costa Rica*
*fabian.fallasmoya@ucr.ac.cr*

*Francisco Torres-Rojas*
*Escuela de Computación*
*Instituto Tecnológico de Costa Rica*
*San José, Costa Rica*
*torresrojas@gmail.com*

**Fecha de recibido: 13 de mayo 2018**

**Fecha de aprobado: 1 de junio de 2018**

**Resumen—** Con los avances recientes en tecnología (hardware y software) la humanidad tiene un interés en tener máquinas que se comporten como los humanos. Un aspecto que los investigadores deben superar es cómo imitar los procesos cognitivos del cerebro; procesos cognitivos como reconocimiento de patrones visuales, reconocimiento de voz, comprensión del espacio, etc. Esta tarea necesita un algoritmo que recibe información sin formato del entorno, por lo tanto, se necesita un método de procesamiento de señal para convertir la entrada bruta en información útil. Computer Vision es un campo de investigación interesante porque el proceso de captura de imágenes es simple y el hardware para procesar estas imágenes está disponible con la tecnología actual. Esta investigación se centra en el campo de la clasificación de imágenes utilizando la memoria temporal jerárquica (por sus siglas en inglés HTM), una técnica de aprendizaje automático que imita la neocorteza y emula procesos cognitivos.

**Palabras clave-** Memoria temporal, aprendizaje automático, proceso cognitivo

**Abstract—** With recent advances in technology (hardware and software) humanity has an interest in having machines that behave like humans. One aspect that researchers must overcome is how to mimic the brain's cognitive processes; cognitive processes such as visual pattern recognition, voice recognition, space understanding, etc. This task requires an algorithm that receives raw information from the environment, therefore a signal processing method is needed to convert

the raw input into useful information. Computer Vision is an interesting field of research because the process of capturing images is simple and the hardware to process these images is available with current technology. This research focuses on the field of image classification using hierarchical temporary memory (HTM), a machine learning technique that mimics the neocortex and emulates cognitive processes.

**Keywords-** Temporal memory, machine learning, cognitive process

## I. INTRODUCCIÓN

El poder de cómputo de una computadora personal es mayor de lo que cualquiera podría imaginar en el pasado. La principal ventaja del uso de computadoras es su capacidad de procesar información más rápido de lo que un humano puede hacerlo; en este contexto, el avance en la visión por computadora es notable. La visión por computadora es un campo que incluye métodos para adquirir, procesar, analizar y comprender imágenes para producir información útil [1]. La clasificación de video podría aplicarse a un seguimiento de cuerpo completo (personas). Este tipo de clasificación es un tema de investigación común, debido a su relevancia en robótica y vigilancia. Esta investigación propone una técnica para clasificar objetos (personas) utilizando un nuevo algoritmo basado en la memoria temporal jerárquica (HTM).

### *A. Clasificación y rastreo en video*

El rastreo en video es el proceso de encontrar la ubicación de uno o más objetos en movimiento en cada cuadro (imagen) de una secuencia de video[8]. Un blanco es el objeto de interés rastreado por un sistema. Un sistema puede rastrear animales, personas, vehículos, microorganismos u otro objeto específico. Sin embargo, rastrear personas es de mucho interés porque puede generar información importante en áreas como el transporte público, la congestión del tráfico, el turismo, la seguridad y la inteligencia empresarial.

Hay un aspecto importante en un sistema de rastreo: detectar si el objetivo está presente en una imagen o no. Es por eso que se puede

implementar un algoritmo de clasificación para detectar si un objetivo específico está presente en una imagen o no. La clasificación es el primer paso para rastrear un objeto y es la principal preocupación de esta investigación.

*B. Retos*

Un video se compone de una secuencia de fotogramas (frames). Una particularidad del video es que su contenido puede verse afectado por aspectos ambientales, por ejemplo, transformaciones de forma o iluminación[5]. Un reto por superar es la confusión (conocido en inglés como clutter), que ocurre cuando la apariencia de un objeto o el fondo son similares a la apariencia del blanco. Los cambios en la pose también dificulta el proceso de clasificación del video. Cuando un objetivo gira, podría mostrar características nuevas que el sistema no reconoce. Otro desafío es la oclusión (en inglés occlusion), es decir, cuando un objetivo no se observa mientras esta ocluido parcial o totalmente por otros objetos. Por ejemplo, cuando una persona se mueve detrás de un automóvil u otro objeto, algunos fotogramas del video pierden el blanco. Este último aspecto (oclusión) es el reto que esta investigación intenta superar.

El rastreo de video se puede definir como un proceso que involucra tres tareas: extracción de características, representación de objetivos y localización. Esta investigación propone una nueva estructura de algoritmo, donde se implementa un proceso de clasificación para verificar la presencia del objetivo en imágenes. El algoritmo utiliza una tecnología emergente para la clasificación como es HTM.

*C. Componentes de HTM*

Una implementación HTM completa utiliza los siguientes componentes:

1. Datos brutos (Raw Data): son los datos en su forma más simple, como enteros, letras, números flotantes, etc.
2. Encoders: Transforma datos sin procesar en representaciones dispersas distribuidas (SDR -

siglas en inglés de Sparse Distributed Representations).

3. Spatial Pooler: recibe el SDR y elige un subconjunto de columnas (de su región).
4. Temporal Pooler: recibe las columnas seleccionadas del spatial pooler y conecta las celdas (sinapsis).
5. CLA (siglas en ingles de Cortical Learning Algorithm): convierte la predicción del temporal pooler en el valor predicho.

Una opción para construir una implementación completa de HTM es usar una herramienta como el Online Prediction Framework (OPF). Este construye una estructura completa con todos los componentes y sus conexiones.

## II. ALGORITMOS PROPUESTOS

La Figura 1 ilustra la implementación de los algoritmos. El caso de las redes neuronales artificiales (ANN - siglas en inglés de Neural Networks, o conocidas simplemente como (NN) Neural Networks) y las máquinas de soporte vectorial (SVM - siglas en inglés de Support Vector Machines) son similares. Ellos reciben los datos directamente al algoritmo de clasificación. NN usa la metodología de feed-forward backpropagation, utilizando la biblioteca PyBrain. SVM usa la biblioteca LIBSVM. Con respecto a HTM Soft, combina un temporal pooler con un clasificador de vecino más cercano (kNN - siglas en ingles de k-nearest neighbor)[7].

Se usa el nombre “Hard” para describir una implementación completa de HTM. Este recibe datos brutos y los procesa con todos los componentes mostrados en la Figura 2. Esto datos brutos se obtienen mediante el paso de extracción de características, para esto se utiliza la transformación de características invariantes de escala (SIFT – siglas en inglés de Scale-Invariant Features Transform).

### *A. SIFT*

La primera tarea para hacer es la extracción de características (figura 1) usando SIFT. Es una de las técnicas de extracción de características más

exitosas [4]. Realiza dos acciones principales: detecta puntos de interés de una imagen (llamado detector de puntos de interés) y obtiene un descriptor local de cada punto (llamado descriptor). Su principal ventaja es evitar diferencias en la rotación y la escala.

En el primer paso todos los puntos de interés de la imagen se detectan usando funciones de diferencia de Gauss [4]. Luego, como Solem [9] describe, se toma el gradiente del punto para indicar la dirección de este. A continuación, se toma una matriz de subregiones alrededor del punto y para cada una se calcula un histograma de orientación del gradiente. Finalmente, los histogramas se concatenan para formar un vector descriptor. Sobre una imagen de alta definición, podemos tener aproximadamente 430720 puntos SIFT. Esta cantidad no es factible para el hardware disponible. Como resultado, se impusieron algunas restricciones: el uso de imágenes de 50 × 50 pixeles y 12 puntos de interés.

*B. SIFT denso*

La Figura 3 muestra un ejemplo de una imagen utilizada para hacer el proceso de clasificación. Tiene una persona del lado izquierdo y levanta las manos. Esta implementación es diferente a SIFT, debido a que los puntos se eligen de forma estática con un radio fijo. A esto se le conoce como SIFT denso. Un beneficio de este enfoque es tener archivos SIFT del mismo tamaño.

En resumen, SIFT denso se puede describir como[3]: (a) la ubicación de cada punto clave no proviene de la característica de gradiente del píxel, sino de una ubicación pre-diseñada; (b) la escala de cada punto clave es la misma que también está pre-diseñada; (c) la orientación de cada punto clave es siempre cero. Con estas suposiciones, SIFT denso puede correr más rápido que la implementación normal. Así como Han et al.[3] lo hicieron, esta investigación se centra en las personas.

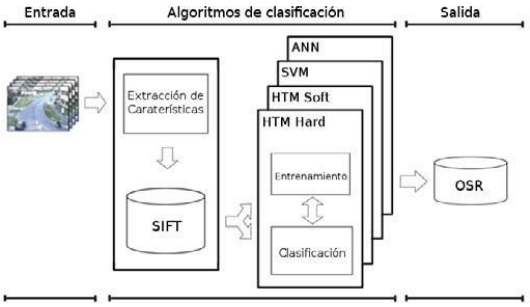


*Fig. 1. La estructura de los algoritmos de rastreo de objetos, desarrollada para esta investigación.*


*Fig. 2. Una estructura completa de una red HTM (una jerarquía completa).*

### *C. Un nuevo algoritmo*

Para crear una estructura completa de una red HTM (como se ve en la Figura 2) se utilizó la herramienta OPF. Esta es una herramienta que crea una estructura HTM con todos loscomponentes de HTM.

Es por eso que se tuvo que hacer una modificación importante (basada en la propuesta de Costa [2]). El primer paso fue crear un modelo en OPF, este modelo se creó para manejar datos pero a una escala baja, por ejemplo, recibiendo de uno a diez valores a la vez. Se usan 12 puntos de interés (de un proceso SIFT denso), cada punto tiene 128 valores (provenientes del descriptor), para un total de 1536 valores. Se modifica un modelo OPF para que aceptara 1536 valores flotantes.

La Figura 2 muestra la secuencia completa de HTM. Primero, los encoders transforman los datos brutos (en este caso puntos SIFT) a datos SDR. Luego, estos SDR son procesados por el spatial pooler y su salida es procesada por el temporal pooler, y finalmente por el CLA. El último paso del algoritmo es calcular la tasa de éxito de la oclusión (OSC-siglas en inglés de Occlusion Success Rate), que sería la precisión del algoritmo.

## III. RESULTADOS Y ANÁLISIS

Este algoritmo se comparó con otras tres implementaciones: redes neuronales, máquinas de soporte vectorial y HTM ´Soft (utilizando el Temporal Pooler conectado con un clasificador k-Nearest Neighbor).

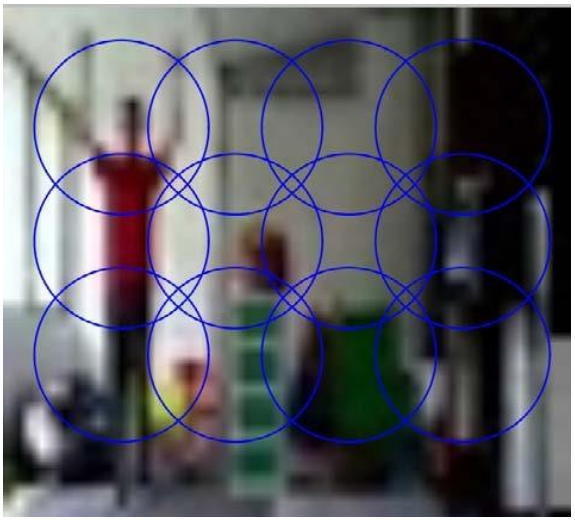

*Fig. 3. Implementación de SIFT. Utiliza una técnica llamada SIFT denso, se establecieron 12 puntos y se aumentó el radio de cada uno. Esta imagen esta tomada de uno de los videos usados para esta investigación.*

La Figura 4 muestra el diagrama de cajas (definido por Massart et al.[6]) como resultado de estas cuatro implementaciones. Este diagrama de cajas ayuda a identificar la media del 50% de los datos, la mediana (la línea horizontal gruesa dentro de los recuadros) y los puntos extremos. Un aspecto es que la técnica de NN tiene el cuartil más bajo y la mediana más baja, lo que demuestra que esta técnica tiende a fallar más que las demás.

HTM Hard y SVM tienen cajas y mediana similares, pero SVM tiene el cuartil superior y la mediana ligeramente más alta, con mejores resultados. Además, el punto máximo de SVM y NN alcanza el valor de precisión de 1.0, a diferencia de los demás, lo que demuestra que estas técnicas fueron precisas en algunas ejecuciones (en condiciones específicas).

Finalmente, el aspecto más interesante es que la caja HTM Hard encierra su 50% de valores en la caja más delgada y el rango más pequeño desde el cuartil superior al punto máximo y desde el cuartil inferior al punto mínimo. Esto significa que la mayoría de los valores de esta técnica están alrededor del valor de precisión de 0,5. Este valor es similar a los otros, pero muestra que este en particular tiende a tener menos fallas durante las pruebas. La figura 4 también muestra muchos valores atípicos, lo cual es normal debido al pequeño rango de cuartiles. En comparación con las otras técnicas. HTM Hard es el algoritmo más estable de todos.

## IV. CONCLUSIONES

Esta investigación ha demostrado el desarrollo de un nuevo algoritmo que utiliza la herramienta OPF para implementar una estructura HTM para clasificar patrones. La evidencia muestra que el algoritmo propuesto tiene una mayor precisión que el uso otras técnicas de clasificación. ANN, SVM y HTM Soft funcionaron rápidamente en un hardware limitado (un ´procesador Intel i7 con 8 GB de RAM) con un aproximado de 5 segundos (por ejecución). Sin embargo, HTM Hard duró 20 minutos por corrida, que es considerablemente un tiempo de ejecución más alto que los demás.

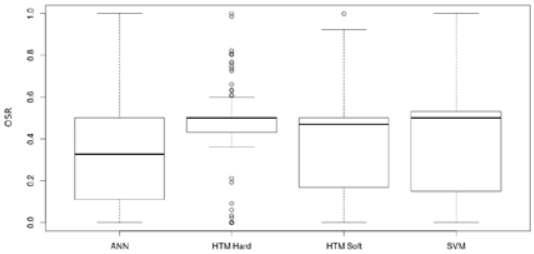


*Fig. 4. Box plot for the significant factor Technique.*

Además, se proporcionó más evidencia para respaldar la idea de utilizar las características de SIFT para el reconocimiento de patrones. Se usó la técnica conocida de SIFT denso y debido a los resultados dados, se encontró que es una herramienta excelente para la extracción de características. También, se demostró que con pocos puntos SIFT los resultados son satisfactorios. No es necesario procesar gran cantidad de información para obtener resultados aceptables mediante SIFT.

## V. TRABAJO FUTURO

Sería interesante usar otra combinación de tecnologías para el algoritmo HTM Soft. Para esta investigación, se combinó el temporal pooler con el clasificador kNN. Sin embargo, el spatial pooler y el temporal pooler se pueden combinar con diferentes tecnologías tales como redes neuronales recurrentes, redes bayesianas, aprendizaje por refuerzo, aprendizaje de diccionarios dispersos, algoritmos genéticos, etc.

Además, se puede usar con video de alta resolución. Aquí se utilizó la resolución de 50×50 pixeles. Y también se puede valorar el utilizar más puntos SIFT (en este caso se utilizó solamente 12).

## VI. REFERENCIAS

[1] C.H. Chen. Handbook of Pattern Recognition and Computer Vision. USA:WSPC, 2016.

[2] Allan Costa. Nupic Classifier MNIST. 2015. URL: http://github. com/allanino/nupic-classifier-mnist (visited on 08/08/2015).

[3] Bing Han, Dingyi Li, and Jia Ji. “People Detection with DSIFTAlgorithm”. In: (2011).

[4] David G. Lowe. “Object recognition from local scale-invariant features.” In: International Conference on Computer Vision (1999), pp. 1150–1157. DOI: 10.1109/ICCV.1999.790410.

[5] Emilio Maggio and Andrea Cavallaro. Video Tracking: Theory and Practice. UK: Wiley: a John Wiley and Sons, Ltd, 2011.

[6] D.L. Massart et al. “Visual Presentation of Data by Means of Box Plots”. In: (2005).

[7] Numenta. Numenta: nupic vision. 2017. URL: http://github. com/ numenta/nupic.vision (visited on 11/05/2017).

[8] Anand Singh Jal al and Vrijendra Singh. “The State-of-the-Art in Visual Object Tracking”. In: Informatica: An International Journal of Computing and Informatics 36.3 (2011), pp. 227–247.

[9] Jan Erik Solem. Programming Computer Vision with Python. USA: O’Reilly Media, 2012.