XVIII Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 2018)

CAEPIA 7:
DEEP LEARNING

Time Series Decomposition for Improving the Forecasting Performance of Convolutional Neural Networks*

*Note: The full contents of this paper have been published in the volume *Lecture Notes in Artificial Intelligence 11160* (LNAI 11160)

Iván Méndez-Jiménez
Centro de Investigaciones Energéticas Medioambientales y Tecnológicas
Madrid, Spain
ivan.mendez@ciemat.es

Miguel Cárdenas-Montes
Centro de Investigaciones Energéticas Medioambientales y Tecnológicas
Madrid, Spain
miguel.cardenas@ciemat.es

***Abstract*—Time Series forecasting is of high interest in the Big Data ecosystem. A larger data volume accessible in industry and science, and a higher profit from more accurate predictions have generated a growing application of Deep Learning techniques in the Time Series forecasting. In this work, the improvement of the forecasting capacity of Convolutional Neural Networks and Recurrent Neural Networks when using as input the trend, seasonal and remainder time series generated by the Seasonal and Trend decomposition using Loess, instead of the original time series observations, is evaluated. The benchmark used in this work is composed of eight seasonal time series with different lengths and origins. Besides, Convolutional Neural Networks and Recurrent Neural Networks, comparisons with Multilayer Perceptrons are also undertaken. As a consequence, an improvement in the forecasting capacity when replacing the original observations by their decomposition in Convolutional Neural Networks-based forecasting is stated.**

***Index Terms*—Time Series Analysis, Deep Learning, Forecasting, Convolutional Neural Networks, Recurrent Neural Networks, Seasonal and Trend Decomposition using Loess**

Comparing Deep Recurrent Networks Based on the MAE Random Sampling, a First Approach*

*Note: The full contents of this paper have been published in the volume *Lecture Notes in Artificial Intelligence 11160* (LNAI 11160)

Andrés Camero, Jamal Toutouh, Enrique Alba
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Málaga, Spain
andrescamero@uma.es,{jamal,eat}@lcc.uma.es

***Abstract*—Recurrent neural networks have demonstrated to be good at tackling prediction problems, however due to their high sensitivity to hyper-parameter configuration, finding an appropriate network is a tough task. Automatic hyper-parameter optimization methods have emerged to find the most suitable configuration to a given problem, but these methods are not generally adopted because of their high computational cost. Therefore, in this study we extend the MAE random sampling, a low-cost method to compare single-hidden layer architectures, to multiple-hidden-layer ones. We validate empirically our proposal and show that it is possible to predict and compare the expected performance of an hyper-parameter configuration in a low-cost way.**

***Index Terms*—Deep learning, Recurrent neural network, MAE random sampling**

Background modeling for video sequences by stacked denoising autoencoders*

*Note: The full contents of this paper have been published in the volume *Lecture Notes in Artificial Intelligence 11160* (LNAI 11160)

Jorge García-González, Juan M. Ortiz-de-Lazcano-Lobato, Rafael M. Luque-Baena,
Miguel A. Molina-Cabello, Ezequiel López-Rubio
Department of Computer Languages and Computer Science
University of Málaga
Málaga, Spain
{jorgegarcia,jmortiz,rmluque,miguelangel,ezeqlr}@lcc.uma.es

***Abstract*—Nowadays, the analysis and extraction of relevant information in visual data flows is of paramount importance. These images sequences can last for hours, which implies that the model must adapt to all kinds of circumstances so that the performance of the system does not decay over time. In this paper we propose a methodology for background modeling and foreground detection, whose main characteristic is its robustness against stationary noise. Thus, stacked denoising autoencoders are applied to generate a set of robust characteristics for each region or patch of the image, which will be the input of a probabilistic model to determine if that region is background or foreground. The evaluation of a set of heterogeneous sequences results in that, although our proposal is similar to the classical methods existing in the literature, the inclusion of noise in these sequences causes drastic performance drops in the competing methods, while in our case the performance stays or falls slightly.**

***Index Terms*—Background modeling, deep learning, autoencoders**

Predicción ordinal de rampas de viento usando *Echo State Networks* de complejidad reducida

M. Dorado-Moreno
Dpto. de Informática y Análisis Numérico
Universidad de Córdoba
Córdoba, España
manuel.dorado@uco.es

P. A. Gutiérrez
Dpto. de Informática y Análisis Numérico
Universidad de Córdoba
Córdoba, España
pagutierrez@uco.es

S. Salcedo-Sanz
Dpto. de Teoría de la Señal y Comunicaciones
Universidad de Alcalá
Alcalá de Henares, España
sancho.salcedo@uah.es

L. Prieto
Dpto. de Recursos Energéticos
Iberdrola
Madrid, España

C. Hervás-Martínez
Dpto. de Informática y Análisis Numérico
Universidad de Córdoba
Córdoba, España
chervas@uco.es

***Resumen*—Las *Renovables* son la fuente de energía que más ha crecido en los últimos años a nivel mundial. En particular, la energía eólica en Europa es actualmente la que tiene un mayor crecimiento, estando su capacidad de producción en la segunda posición, por detrás del gas natural. Existen una serie de problemas que complican la integración del recurso eólico en la red eléctrica. Uno de ellos es conocido como rampas de viento, que son incrementos o decrementos de gran magnitud en la velocidad del viento en un tiempo reducido. Estas rampas de viento pueden dañar las turbinas en los parques eólicos, así como reducir los ingresos generados a partir de la producción del parque. Actualmente, la mejor forma de afrontar este problema es predecir estas rampas de viento, de forma que se puedan parar las turbinas con suficiente antelación, evitando así los daños que puedan producirse. Para realizar esta predicción, se suelen utilizar modelos que puedan aprovechar la información temporal. Uno de los modelos más conocidos con estas características son las redes neuronales recurrentes. En este trabajo utilizaremos las conocidas como *Echo State Networks* (ESNs), las cuales han demostrado obtener un buen rendimiento en predicción de series temporales. En concreto, proponemos utilizar ESNs de complejidad reducida para afrontar un problema de predicción de rampas de viento en tres parques eólicos en España. A nivel metodológico, se comparan tres arquitecturas diferentes de red, dependiendo de la configuración de las conexiones de la capa de entrada con el *reservoir* o directamente con la capa de salida. Los resultados muestran que, por lo general, los mejores resultados son obtenidos por la estructura *Delay Line Reservoir with Feedback* (DLRB) y que el aumento en el rendimiento obtenido por la arquitectura de Doble *reservoir* con respecto a la arquitectura de Simple *reservoir* es mínima, y teniendo en cuenta el gran aumento de complejidad computacional de la arquitectura Doble, concluimos que los mejores resultados son obtenidos por la combinación de la estructura DLRB con la arquitectura Simple.**

Este trabajo ha sido desarrollado con la financiación de los proyectos TIN2017-85887-C2-1-P, TIN2017-85887-C2-2-P y TIN2017-90567-REDT del Ministerio de Economía y Competitividad de España (MINECO) y fondos FEDER. La investigación de Manuel Dorado-Moreno ha sido financiada por el programa predoctoral FPU (Ministerio de Educación y Ciencia) con referencia FPU15/00647. Los autores agradecen a *NVIDIA Corporation* la cesión de recursos computacionales a través del *GPU Grant Program*.

***Index Terms*—Echo state networks; Energía eólica; Clasificación ordinal; Rampas de viento; Redes neuronales recurrentes.**

I. Introducción

La naturaleza nos ofrece múltiples formas de generar energía de forma sostenible y libres de emisiones contaminantes. Este tipo de energías explotan recursos naturales renovables y actualmente son las que más están creciendo a nivel mundial. Algunas de las más conocidas son la energía solar, la eólica y la marina (mareomotriz, undimotriz, eólica *offshore*), así como sus combinaciones, aunque existen otras alternativas tales como la biomasa o la energía hidráulica. Nuestro trabajo se centrará en este caso en la energía eólica y, dentro de esta, en la producción en parques eólicos, los cuales, generan energía mediante turbinas eólicas de grandes dimensiones. El problema de la gran mayoría de los recursos renovables es que normalmente son intrínsecamente intermitentes, lo que dificulta la completa explotación del recurso, y su incorporación al *mix* energético en igualdad de condiciones con respecto a otros tipos de recursos no renovables. En el caso de la energía eólica, además de su intermitencia intrínseca, aparecen otro tipo de problemas en producción, relacionados con características específicas del recurso. En el caso de los parques eólicos, uno de los problemas más grave son las conocidas como *rampas de viento*, definidas como incrementos o decrementos en la velocidad del viento de gran magnitud en un corto periodo de tiempo. Estas rampas de viento pueden ser positivas, es decir, que se produce un incremento de la velocidad del viento, o negativas, cuando es un decremento. El efecto de las rampas positivas en un parque eólico es, principalmente, el posible daño que pueden causar a las turbinas existentes. Esto puede derivar en un aumento de los costes de mantenimiento del parque. En cuanto a las rampas negativas, su efecto fundamental es un decremento de producción de energía súbito, que puede acarrear problemas

de abastecimiento si este tipo de sucesos no se predice con suficiente antelación.

Muy diversos problemas relacionados con energías renovables han sido abordados mediante técnicas de aprendizaje automático, por ejemplo, en energía solar [2], undimotriz [8] o energía eólica [4]–[6]. En aprendizaje automático, uno de los modelos más conocidos a la hora de tratar con series temporales y realizar predicciones son las redes neuronales recurrentes [12]. La diferencia con las redes neuronales convencionales es la inclusión de ciclos entre sus neuronas, es decir, se permiten conexiones de una neurona consigo misma y con neuronas situadas en las capas anteriores o en su misma capa. De cualquier forma, al incrementar el número de capas (o neuronas) de las redes recurrentes para incrementar su capacidad de cómputo, las redes sufren un problema conocido como desvanecimiento del gradiente [12], debido al cual, al ir enlazando las derivadas a través de los ciclos, estas tienden a cero, por lo que no aportan información al gradiente e impiden la actualización de los pesos. Una de las propuestas más aceptadas para solventar este problema son las *echo state networks* (ESNs) [11], las cuales tienen una capa oculta conocida como *reservoir* en la que se encuentran todos los ciclos de los enlaces entre las neuronas, inicializados de forma aleatoria. Este *reservoir* está totalmente conectado con las entradas y las salidas, y estas últimas conexiones son las únicas que se entrenan. Así, se evita el problema de la tendencia del gradiente a 0, ya que no es necesario utilizar las derivadas para entrenar los pesos del *reservoir*.

Una de las dificultades asociadas a las ESNs es su naturaleza estocástica, ya que parte de su rendimiento depende del azar. Para solventar este problema, en este trabajo vamos a utilizar las ESNs de complejidad reducida propuestas en [14], las cuales establecen los enlaces entre las neuronas siguiendo un patrón razonable y además los inicializan de forma determinista, pudiendo así, justificar su correcto funcionamiento. Además proponemos tres arquitecturas distintas, siguiendo el trabajo realizado en [5], para comprobar las distintas formas en las que el *reservoir* afecta al resultado del modelo dependiendo de las entradas que se conecten a él. Es importante destacar que, debido al orden natural que muestran las distintas categorías a predecir (rampa negativa, ausencia de rampa y rampa positiva), el problema se aborda desde la perspectiva de la clasificación ordinal [10]. Por último, cabe destacar que para este trabajo se utilizarán dos fuentes de datos, a partir de las cuales extraeremos las variables de entrada utilizadas por las ESNs para la predicción de las rampas de viento. La primera incluye mediciones recogidas por los sensores de tres parques eólicos situados en España (ver Figura 1), mientras que la segunda se corresponde datos generados a través de modelos físicos y matemáticos que son conocidos como datos de reanálisis [7]. Estos datos de reanálisis son muy fiables y se calculan a lo largo de todo el mundo cada 0,125 grados (en latitud y longitud) y cada 6 horas, alcanzando así una buena resolución tanto espacial como temporal.

En la Sección II se expondrán las características extraídas de las dos bases de datos y se explicará la generación de la base de datos conjunta. Las distintas arquitecturas propuestas para el modelado se expondrán en la Sección III, justo antes de explicar el diseño experimental en la Sección IV. Para concluir, en la Sección V, se mostrarán los resultados obtenidos y se realizará una discusión de los mismos. La Sección VI expondrá las conclusiones obtenidas tras este trabajo.

II. Base de datos

En esta sección se explican las características de las fuentes de información utilizadas para resolver el problema de la predicción de rampas de viento, las transformaciones llevadas a cabo y la unión de ambas fuentes de información. La primera fuente de información corresponde a medidas de la velocidad de viento, obtenidas cada hora en tres parques eólicos situados en España, como se puede observar en la Figura 1. Calcularemos las rampas de viento como valores objetivo a ser predichos utilizando distintas variables predictivas. La segunda fuente de información de la que obtendremos variables predictivas es el proyecto de reanálisis ERA-Interim [7], que almacena información sobre el clima cada 6 horas. Estos datos se calculan utilizando modelos físicos, es decir, que no dependen de ningún sensor que pueda generar datos perdidos, por lo que se pueden estimar valores futuros de estos modelos para predecir las rampas de viento.

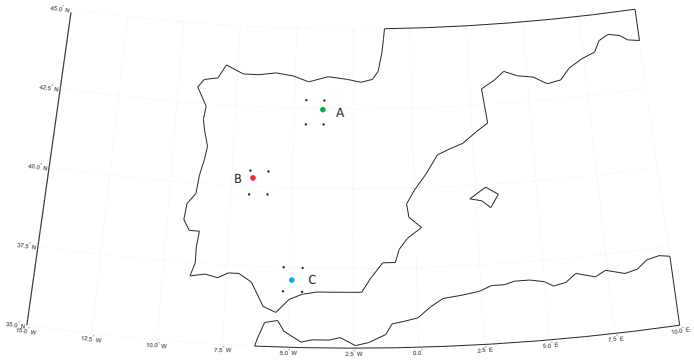


Figura 1: Localización de las tres parques eólicos (A, B y C) y de los nodos de reanálisis considerados en las cercanías de cada parque.

II-A. Definición de Rampa de Viento

La función $S_t : \mathbb{R}^k \rightarrow \mathbb{R}$ será la función evaluada para decidir si se ha producido una rampa de viento, o no, en un determinado periodo de tiempo, donde k será el número de características a considerar. Hay muchas definiciones de S_t [9] y todas de ellas incluyen la producción de energía (P_t) como un criterio del parque eólico o la turbina de viento a considerar. En este artículo utilizamos la siguiente definición:

$$S_t = P_t - P_{t-\Delta t_r},$$

donde Δt_r es el intervalo de tiempo considerado para estudiar la rampa (6 horas en nuestro caso, de acuerdo con la frecuencia de los datos de reanálisis).

Utilizando S_t, podemos definir el problema de clasificación definiendo un umbral (S_0) para discretizar la variable objetivo de la siguiente forma:

$$y_t = \begin{cases} \mathcal{C}_{\mathrm{NR}}, & \text{si } \ S_t \leq -S_0, \\ \mathcal{C}_{\mathrm{NoR}}, & \text{si } \ -S_0 < S_t < S_0, \\ \mathcal{C}_{\mathrm{PR}}, & \text{si } \ S_t \geq S_0. \end{cases}$$

donde $\{\mathcal{C}_{\mathrm{NR}}, \mathcal{C}_{\mathrm{NoR}}, \mathcal{C}_{\mathrm{PR}}\}$ son las diferentes categorías de eventos a detectar, es decir, rampas negativas (NRs), no rampas (NoRs) y rampas positivas (PRs).

En nuestro caso, S_0 se ha definido como un porcentaje de la capacidad de producción de energía del parque eólico (en concreto, un $50\,\%$ siguiendo [9]). La predicción de rampas de viento también implica un vector de variables predictivas: utilizaremos datos de reanálisis climatológicos como datos de entrada ($\mathbf{z}$) (definidos en la siguiente sección), junto con el valor de la velocidad de viento medido por los sensores del parque eólico en el instante anterior al que queremos predecir.

II-B. Datos de reanálisis

Para cada una de los tres parques, tenemos 48 predictores, que corresponden a 12 variables por cada nodo de reanálisis (considerando los 4 nodos más cercanos a la parque eólico, ver Figura 1). Estos nodos están situados cada 15 kilómetros (0.125 grados) en todo el mundo y en esas localizaciones se calculan las variables de reanálisis utilizando modelos físicos que estiman medidas de variables meteorológicas a muy diferentes alturas. Entre las mismas, tenemos la velocidad de viento, presión y temperatura del aire. Para evitar trabajar con tantos datos, que en muchos casos estarán altamente correlados, por lo que introducirán ruido al modelo, realizamos una media ponderada por la distancia de cada nodo de reanálisis al centro del parque eólico. De esta forma reducimos el número de predictores de reanálisis a 12, sin perder la información relativa de cada nodo. Así, en primer lugar calculamos la distancia de cada nodo de reanálisis al parque eólico mediante la distancia de Haversine:

$$\begin{aligned} d(p_0, p_j) = &\arccos(\sin(lat_0) \cdot \sin(lat_j) \cdot \\ &\cdot \cos(lon_0 - lon_j) + \cos(lat_0) \cdot \cos(lat_j)), \end{aligned}$$

donde p_0 es la localización del parque eólico, p_j la localización de cada nodo de reanálisis, y lat y lon serán la latitud y longitud de los puntos, respectivamente. Una vez que la distancia de cada uno de los nodos de reanálisis (los cuatro puntos negros que rodean a cada parque eólico, ver de nuevo Figura 1) al parque eólico ha sido calculada, estas distancias se invierten y normalizan, considerando que cuanto más corta sea la distancia, más grande será el peso de la información de ese nodo de reanálisis en la media ponderada:

$$w_i = \frac{\sum_{j=1}^{4} d(p_0, p_j)}{d(p_0, p_i)}, \quad i = 1, \ldots, 4. \tag{1}$$

Después de calcular estos pesos, se utilizan para obtener una media ponderada de cada una de las 12 variables:

$$\bar{z_i} = \sum_{j=1}^{4} w_j z_{i,j} \quad i \in \{0, 1, \cdots, 11\}$$

siendo i el índice de cada una de las variables de reanálisis, j cada uno de los nodos de reanálisis y w_j el peso correspondiente, calculado en la Ecuación (1).

III. Arquitecturas propuestas

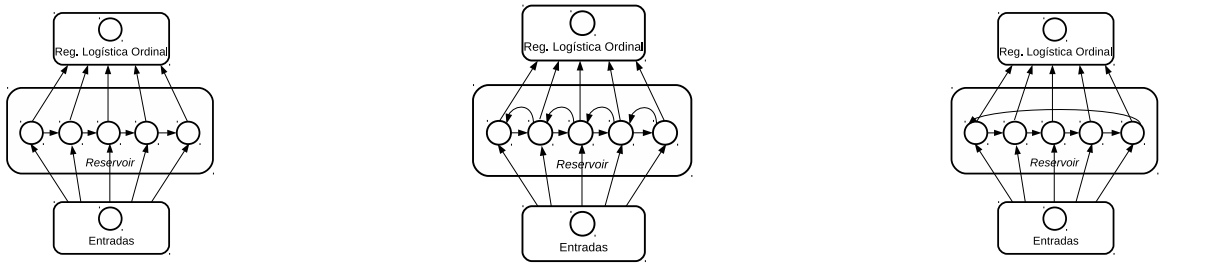
En este artículo proponemos una modificación de los modelos utilizados en [5]: en lugar de realizar una clasificación binaria, vamos a resolver un problema de clasificación ordinal para tres clases. Por otra parte, vamos a modificar la estructura del *reservoir* basándonos en las distintas estructuras propuestas en [14], las cuales reducen la complejidad del *reservoir* además de eliminar la aleatoriedad en la inicialización de los mismos, sin reducir de forma considerable el rendimiento del modelo. Gracias a estas estructuras de *reservoir* de complejidad reducida, se construyen los ciclos entre las neuronas del *reservoir* de forma determinista, además de otorgarle a la red una capacidad de memoria óptima para cada problema. Un esquema de las tres estructuras de *reservoir* puede analizarse en la Figura 2. En la capa de salida utilizaremos un modelo de regresión logística ordinal basado en umbrales [13], el cual proyecta los patrones en una dimensión y optimiza el valor de los umbrales para separar las distintas clases.

A continuación describiremos las arquitecturas propuestas para resolver la predicción de rampas de viento, que exploran distintas formas de combinar los valores pasados de la velocidad de viento y los datos de reanálisis procedentes del proyecto ERA-Interim. Proponemos tres arquitecturas, la primera (ver Figura 3a) tiene un único *reservoir* cuya entrada es la velocidad de viento recogida en cada parque eólico en el instante anterior al que se quiere predecir, de forma que el resto de variables de reanálisis se utilizarán directamente como entradas a la capa de salida, sin ser procesadas por ningún *reservoir*. La segunda arquitectura (ver Figura 3b) dispone de dos *reservoir* independientes, uno para la velocidad del viento y otro cuyas entradas serán todas las variables de reanálisis. Por último, proponemos una tercera arquitectura (ver Figura 3c) en la que solo disponemos de un *reservoir* cuyas entradas englobarán tanto la velocidad del viento como el resto de variables de reanálisis. Con estas tres arquitecturas, estudiaremos la capacidad de cómputo del *reservoir* así como su utilidad para cada tipo de variables.

En la capa de entrada, incluimos los vectores de entrada con la velocidad de viento y las 12 variables de reanálisis en los instantes t (para la primera) y $t+1$ (para las demás), respectivamente. El uso de $\mathbf{z}_{t+1}$ en la capa de entrada para predecir y_{t+1} es posible, como se ha mencionado en la sección II, debido a que estos datos se calculan mediante modelos físicos que permiten estimarlos de forma fiable 6 horas después del instante actual.

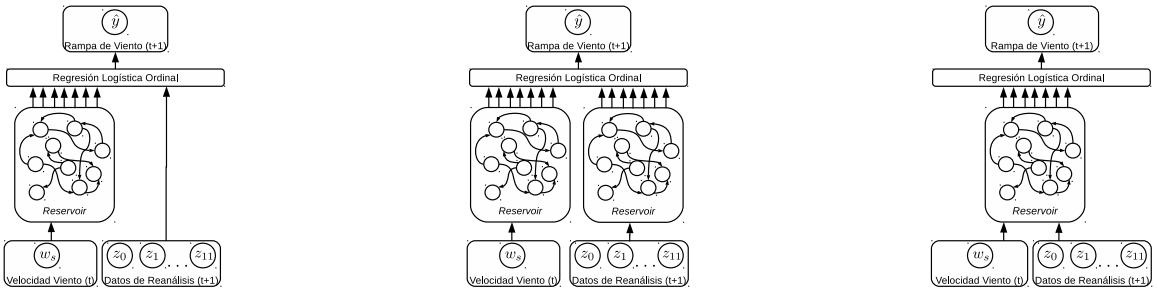
La metodología propuesta para entrenar los distintos modelos propuestos es la siguiente:

1. Crear un *reservoir* de tamaño M, conectando sus neuronas según las restricciones de cada tipo de *reservoir* (DLR, DLRB y SCR) tal y como se indica en [14].
2. Recoger todos los estados del *reservoir*. Para ello, se alimenta el *reservoir* desde el instante $t = 1$ hasta el



(a) *Delay Line Reservoir* (DLR) (b) *Delay Line Reservoir with feedback* (DLRB) (c) *Single Cycle Reservoir* (SCR)

Figura 2: Estructuras de *reservoir* consideradas



(a) *Reservoir* Simple (b) *Reservoir* Doble (c) *Reservoir* Compartido

Figura 3: Distintas arquitecturas de red propuestas

instante $t = M$, de forma tal que todas las conexiones del *reservoir* hayan recibido una señal, permitiendo obtener el vector $\mathbf{x}_t$ completo.

3. Combinar los estados del *reservoir* ($\mathbf{x}_t$) y/o las variables de reanálisis ($\mathbf{z}_{t+1}$).
4. Calcular los pesos de salida entrenando un modelo de regresión logística ordinal, proyectando los patrones sobre una recta y estableciendo los umbrales para distinguir cada una de las tres clases ($\mathcal{C}_{\text{NR}}$, $\mathcal{C}_{\text{NoR}}$ y $\mathcal{C}_{\text{PR}}$).

Una vez que la red ha sido entrenada, se puede utilizar para la predicción en tiempo real de rampas de viento, descartando el patrón correspondiente a $t = 0$ ya que no habría información de instantes anteriores de tiempo para poder predecirlo.

IV. Diseño experimental

En esta sección se describen los experimentos que se han llevado a cabo para comparar las distintas arquitecturas propuestas en la Sección III.

IV-A. Métricas de evaluación

Hay muchas métricas que pueden ser utilizadas para evaluar clasificadores ordinales. Las más comunes en aprendizaje automático son el Error Absoluto Medio (MAE) y el Error Medio Cero-uno (MZE) [10], siendo $MZE = 1 - \text{Acc}$, donde Acc es la precisión o la proporción de patrones bien clasificados. De cualquier modo, estas métricas podrían no ser las mejores opciones, por ejemplo, cuando medimos el rendimiento en bases de datos desequilibradas (como es nuestro caso, ver Tabla I) [1], y/o cuando los costes de diferentes errores varían notablemente. Es por ello que, para poder evaluar correctamente el rendimiento de los clasificadores, hemos incluido la media geométrica de las sensibilidades (GMS), que es una métrica más estricta a la hora de penalizar la mala clasificación de las clases minoritarias, siendo 0 cuando una de las clases está mal clasificada por completo. Así, hemos considerado 3 métricas para evaluar estos modelos:

- La proporción de patrones bien clasificados (CCR) que se define como:

$$CCR = \frac{1}{N}\sum_{i=1}^{N}(I(y_i^* = y_i)),$$

donde $I(\cdot)$ es la función de pérdida cero-uno, y_i es la salida deseada para el patrón $\mathbf{x}_i$, y_i^* es la predicción del modelo y N es el número de patrones del conjunto de entrenamiento o *test* en la base de datos. Los valores del CCR varían de 0 a 1 y representa el rendimiento global de la tarea de clasificación.

- Las sensibilidades de cada clase representan la habilidad del modelo para predecir correctamente cada tipo de evento:

$$S_{\text{NR}} = \frac{CC_{\text{NR}}}{N_{\text{NR}}},\ S_{\text{NoR}} = \frac{CC_{\text{NoR}}}{N_{\text{NoR}}},\ S_{\text{PR}} = \frac{CC_{\text{PR}}}{N_{\text{PR}}},$$

donde CC_{NR}, CC_{NoR} y CC_{PR} es el número de eventos de tipo NR, NoR y PR bien clasificados y N_{NR}, N_{NoR} y N_{PR} ($N_{\text{NR}} + N_{\text{NoR}} + N_{\text{PR}} = N$) es el número total de eventos de tipo NR, NoR y PR. La media geométrica de las sensibilidades (GMS) se define como:

$$GMS = \sqrt[3]{S_{\text{NR}} \cdot S_{\text{NoR}} \cdot S_{\text{PR}}}.$$

Incluimos esta métrica debido al alto desequilibrio de la base de datos, de forma que los clasificadores que no se centren en una de las clases sean fácilmente detectables, ya que su GMS será cercano a 0.

- La media del error absoluto medio ($AMAE$) [1] es la media del error de clasificación MAE en cada clase, donde MAE es la media de la desviación absoluta entre la clase predicha y la clase real. Esta métrica es capaz de mitigar el efecto de las distribuciones de clase desequilibradas y se define como:

$$AMAE = \frac{1}{J}\sum_{j=1}^{J} MAE_j,$$

donde

$$MAE_j = \frac{1}{N_j}\sum_{i=1}^{N_j} |\mathcal{O}(y_i) - \mathcal{O}(y_i^*)|,$$

donde $1 \leq j \leq J$, $\mathcal{O}(\mathcal{C}_{\mathrm{NR}}) = 1, \mathcal{O}(\mathcal{C}_{\mathrm{NoR}}) = 2, \mathcal{O}(\mathcal{C}_{\mathrm{PR}}) = 3$. Los valores de MAE van desde 0 hasta $J-1$, al igual que los de $AMAE$. En nuestro caso tendremos $J = 3$.

IV-B. Diseño Experimental

Los tres parques eólicos de la Figura 1 se han utilizado en la comparación de resultados de las distintas arquitecturas propuestas. Para evaluar los resultados, las tres bases de datos se han dividido de la misma forma: los últimos 365 días son utilizados para el conjunto de test y el resto de la base de datos para el conjunto de entrenamiento. Todas las bases de datos comienzan el 2/3/2002 y finalizan el 29/10/2012. Con esta partición de los datos, los patrones por clase de cada una de las tres bases de datos se muestra en la Tabla I, especificando el tipo de evento recogido (rampa negativa, NR, no rampa, NoR, y rampa positiva, PR).

Tabla I: Número de patrones por clase de las distintas bases de datos consideradas en la experimentación

Parque	Conjunto	#NR	#NoR	#PR
A	Entrenamiento	753	12469	886
	Test	67	1288	105
B	Entrenamiento	1161	11804	1074
	Test	117	1220	123
C	Entrenamiento	661	12768	679
	Test	58	1340	62

Las diferentes arquitecturas presentadas en la sección III han sido comparadas entre si, comparando además las distintas estructuras internas del *reservoir* de acuerdo a [14]. Deseamos identificar la arquitectura con mejor rendimiento y comprobar si el *reservoir* de complejidad reducida es suficiente para nuestro problema.

Debido al desequilibrio del problema, realizamos un sobre-muestreo mediante la metodología SMOTE [3] a las salidas del *reservoir*, tal y como se explica y justifica en [4]. Para cada clase minoritaria ($\mathcal{C}_{\mathrm{NR}}$ y $\mathcal{C}_{\mathrm{PR}}$), generaremos tantos patrones sintéticos como se indique mediante una proporción del número de patrones de la clase mayoritaria (en nuestro caso, un 60 % de los patrones de la clase mayoritaria), evitando así obtener clasificadores triviales. El sobre-muestreo se realiza únicamente sobre el conjunto de entrenamiento.

Si no se controlan los pesos de la regresión logística ordinal, estos pueden llegar a ser muy grandes, sobre-ajustando la red o impidiendo clasificar los patrones de las clase minoritarias. Para paliar este efecto, incluimos un término de regularización, que obliga a ajustar un parámetro α que controla su importancia. Este parámetro ha sido ajustado mediante una validación cruzada interna de tipo 5-*fold* sobre el conjunto de entrenamiento. Los valores valores explorados han sido $\alpha \in \{2^{-5}, 2^4, ..., 2^{-1}\}$. La selección final del mejor valor se realiza en base a maximizar la mínima sensibilidad, es decir, $MS = \min\{S_{\mathrm{NR}}, S_{\mathrm{NoR}}, S_{\mathrm{PR}}\}$.

El resto de parámetros se han configurado de la siguiente forma: el número de neuronas del *reservoir* es $M = 50$, asumiendo que es un tamaño suficiente para abordar el problema sin suponer un coste computacional demasiado alto. Los valores de los enlaces del *reservoir* se establecen según una distribución uniforme entre los valores $[-0{,}9, 0{,}9]$.

V. Resultados

Esta sección expone y discute los resultados obtenidos por las distintas arquitecturas de red y las diferentes estructuras de *reservoir* propuestas. En la Tabla II, se pueden observar los resultados de las tres arquitecturas con el *reservoir* de estructura DLR, en la Tabla III los resultados con la estructura DLRB y en la Tabla IV los resultados obtenidos usando la estructura SCR (ver Figuras 2a, 2b y 2c, respectivamente).

Tabla II: Resultados obtenidos con las tres arquitecturas propuestas con estructura DLR

Parque	Modelo	GMS	AMAE	CCR	MS
A	Simple	*0,6607*	*0,3485*	0,7212	*0,5671*
	Doble	**0,6951**	**0,3060**	**0,7411**	**0,5820**
	Compartido	0,3056	0,6207	*0,7328*	0,1791
B	Simple	**0,6394**	**0,3850**	*0,7082*	**0,5811**
	Doble	*0,6311*	*0,3903*	0,7000	*0,5726*
	Compartido	0,3185	0,5921	**0,7630**	0,0813
C	Simple	**0,6344**	**0,3768**	0,7383	*0,5689*
	Doble	*0,6227*	*0,3931*	*0,7452*	**0,5862**
	Compartido	0,2443	0,6598	**0,7636**	0,0967

El mejor resultado se muestra en negrita y el segundo mejor en cursiva

Tal y como se observa en la Tabla II, la arquitectura simple gana en dos de los tres parques eólicos, la doble gana en uno y la compartida obtiene los peores resultados. Como se ha comentado anteriormente en la sección IV-A, se obtiene un alto valor de CCR pero a coste de un valor muy bajo de GMS, clasificando incorrectamente las clases minoritarias.

Por el contrario, en la Tabla III, la arquitectura con *reservoir* doble gana en dos de los tres parques eólicos, mientras que la arquitectura con un único *reservoir* para la velocidad de

Tabla III: Resultados obtenidos con las tres arquitecturas propuestas con estructura DLRB

Parque	Modelo	GMS	AMAE	CCR	MS
A	Simple	*0,6715*	0,3389	0,7294	*0,5970*
	Doble	**0,6971**	**0,3057**	*0,7397*	**0,6268**
	Compartido	0,3630	0,5484	**0,7863**	0,1343
B	Simple	**0,6397**	**0,3847**	*0,7089*	**0,5726**
	Doble	*0,6352*	*0,3912*	0,7006	*0,5470*
	Compartido	0,1648	0,6634	**0,7821**	0,0427
C	Simple	*0,6290*	*0,3871*	0,7376	*0,5645*
	Doble	**0,6437**	**0,3733**	*0,7486*	**0,5862**
	Compartido	0,1922	0,6454	**0,8445**	0,0645

El mejor resultado se muestra en negrita y el segundo mejor en cursiva

viento obtiene el segundo resultado en estos dos parques y el mejor resultado en el restante. El mal comportamiento de la arquitectura con *reservoir* compartido se repite.

Tabla IV: Resultados obtenidos con las tres arquitecturas propuestas con estructura SCR

Parque	Modelo	GMS	AMAE	CCR	MS
A	Simple	*0,6607*	0,3485	0,7212	*0,5671*
	Doble	**0,6951**	**0,3060**	**0,7411**	**0,5970**
	Compartido	0,3056	0,6207	*0,7328*	0,1492
B	Simple	**0,6394**	**0,3850**	*0,7082*	**0,5726**
	Doble	*0,6311*	*0,3903*	0,7000	*0,5641*
	Compartido	0,3185	0,5921	**0,7630**	0,1623
C	Simple	**0,6344**	**0,3768**	*0,7383*	**0,5689**
	Doble	*0,6227*	*0,3931*	*0,7383*	*0,5517*
	Compartido	0,2443	0,6598	**0,7636**	0,1290

El mejor resultado se muestra en negrita y el segundo mejor en cursiva

Por último, los resultados obtenidos con la estructura SCR, que podemos observar en la Tabla IV, siguen la misma dirección que los obtenidos con la estructura DLR, lo que indica que en dos de las tres bases de datos el modelo simple obtiene los mejores resultados.

Comparando las tres tablas, la estructura de *reservoir* que mejor rendimiento obtiene para la predicción de rampas de viento es la DLRB. A su vez, la arquitectura de *reservoir* doble solo mejora los resultados para la estructura DLRB (y no para las otras dos). Si consideramos el incremento de complejidad que se introduce en el entrenamiento de la regresión logística ordinal (62 entradas para el modelo simple frente a 100 entradas para el modelo doble), podemos afirmar que la arquitectura de *reservoir* simple es la más adecuada para este problema.

VI. Conclusiones

Este artículo propone y evalúa tres arquitecturas distintas de redes recurrentes, y cada una de estas tres arquitecturas se crea con una estructura diferente de *reservoir* de complejidad reducida. Estas propuestas se utilizan para realizar una predicción ordinal en tres clases de rampas de viento, donde también se considera el alto nivel de desequilibrio de la base de datos.

Las arquitecturas propuestas cambian la forma en que se procesan los datos de entrada, que son una combinación de la velocidad del tiempo medida en el parque eólico y 12 variables de reanálisis. Por una parte tenemos solo la velocidad de viento (medida en cada parque eólico) procesada por el *reservoir* mientras que todas las variables de reanálisis se introducen directamente en la regresión logística ordinal. Otra de las arquitecturas dispone de dos estructuras de *reservoir*, una para procesar la velocidad del viento y otra para procesar las variables de reanálisis. La última dispone de un único *reservoir* para procesar todas las entradas.

Para evitar los modelos triviales (que clasifican todo en la clase mayoritaria), hemos aplicado sobre-muestreo a las activaciones del *reservoir*, mejorando así los resultados. Podemos concluir que el modelo con un único *reservoir* para la velocidad de viento obtiene el mejor rendimiento, siendo pocos los casos en los que la arquitectura de doble *reservoir* funciona mejor que la simple. Con respecto a las estructuras de *reservoir* de complejidad reducida que han sido comparadas, los mejores resultados se obtienen con la estructura *Delay Line Reservoir with feedBack*).

Referencias

[1] S. Baccianella, A. Esuli, F. Sebastiani, Evaluation measures for ordinal regression, in: Proc. of the 9th Int. Conf. on Intelligent Systems Design and Apps, pp. 283-287 (2009).

[2] S. Basterrech and T. Buriánek, "Solar irradiance estimation using the Echo State Network and the flexible neural tree," *Intelligent data analysis and its Applications, Volume I*, Springer, pp. 475-484 (2014).

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research* vol. 16, pp. 321-357 (2002).

[4] M. Dorado-Moreno, A.M. Durán-Rosal, D. Guijo-Rubio, P.A. Gutiérrez, L. Prieto, S. Salcedo-Sanz and C. Hervás-Martínez,"Multiclass Prediction of Wind Power Ramp Events Combining Reservoir Computing and Support Vector Machines," *Conference of the Spanish Association for Artificial Intelligence, Lecture Notes in Computer Science*, vol. 9868, pp. 300-309 (2016).

[5] M. Dorado-Moreno, L. Cornejo-Bueno, P.A. Gutiérrez, L. Prieto, S. Salcedo-Sanz and C. Hervás-Martínez,"Combining *reservoir* computing and over-sampling for ordinal wind power ramp prediction," *International Work-conference on Artificial Neural Networks, Lecture Notes in Computer Science*, vol. 10305, pp. 708-719 (2017).

[6] M. Dorado-Moreno, L. Cornejo-Bueno, P.A. Gutiérrez, L. Prieto, C. Hervás-Martínez and S. Salcedo-Sanz, "Robust estimation of wind power ramp events with reservori computing," *Renewable Energy*, vol. 111, pp. 428-437 (2017).

[7] D. P. Dee, S. M. Uppala, A. J. Simmons, P. Berrisford, P. Poli et al., "The ERA-Interim reanalysis: configuration and performance of the data assimilation system," *Quart. J. of the Royal Met. Society*, vol. 137, pp. 553-597 (2011).

[8] J. C. Fernandez, S. Salcedo-Sanz, P. A. Gutiérrez, E. Alexandre y C. Hervás-Martínez. "Significant wave height and energy flux range forecast with machine learning classifiers," *Engineering Applications of Artificial Intelligence,* vol. 43, pp. 44-53 (2015).

[9] C. Gallego-Castillo, A. Cuerva-Tejero and O. López-García, "A review on the recent history of wind power ramp forecasting, "*Renewable and Sustainable Energy Rev.*, vol. 52, pp. 1148-1157, (2015).

[10] Pedro Antonio Gutiérrez, María Pérez-Ortiz, Javier Sánchez-Monedero, Francisco Fernández-Navarro, and César Hervás-Martínez, "Ordinal regression methods: survey and experimental study," *IEEE Trans. on Knowledge and Data Engineering*, vol.28, pp. 127-146 (2016)

[11] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks," *GMD Report* 148, German National Research Center for Information Technology, pp. 1-43, (2001).

[12] M. Lukosevicius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127-149 (2009).
[13] P. McCullagh, "Regression Modelos for Ordinal data," *J. of the Royal Statistic Society*, vol. 42, no. 2, pp. 109-142 (1980).
[14] A. Rodan and P. Tiňo, "Minimum Complexity Echo State Network," *IEEE Trans. Neural Networks*, vol. 22, no. 1, pp. 131-144 (2011).

Detección de variedad y estado de maduración del ciruelo japonés utilizando imágenes hiperespectrales y aprendizaje profundo

F. Chávez, B. Rodríguez-Puerta, F. J. Rodríguez-Díaz
Dpto. de Ingeniería en Sistemas Informáticos y Telemáticos
Universidad de Extremadura
06800 Mérida, España.
Email: {fchavez, brpuerta, fjrodriguez}@unex.es

Rafael M. Luque-Baena
Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga
29071, Málaga, España
Email: rmluque@lcc.uma.es

***Resumen*—En la actualidad, España ocupa el séptimo puesto como productor de ciruelas a nivel mundial y el tercero a nivel europeo según la Organización de las Naciones Unidas para la Alimentación y la Agricultura. La importancia que tiene el cultivo de esta fruta en nuestro país es evidente, siendo mayor en Comunidades Autónomas como la Extremeña, que centran su actividad económica en el sector primario. Lo que debe diferenciar una producción es su calidad, pero la calidad de los frutos tradicionalmente se hace en base a la experiencia de los agricultores y técnicos, basándose únicamente en su percepción visual. Esto puede generar errores en la determinación de la fecha óptima de recolección.**

En este trabajo se propone un método novedoso basado en el análisis de imágenes hiperespectrales de los frutos del ciruelo japonés que, mediante técnicas de aprendizaje profundo (Deep Learning) y utilizando para ello redes neuronales convolucionales, se obtienen eficaces clasificadores de los frutos por su variedad y su fecha de maduración. Los resultados presentados en este trabajo permiten afirmar que es posible dotar a los agricultores y técnicos agrícolas de herramientas que les ayuden a la correcta toma de decisón en relación a la fecha de maduración de sus frutos, para poder obtener productos de mayor calidad y ser más competitivos en el sector.

I. INTRODUCTION

La Organización de las Naciones Unidad para la Alimentación (FAO) sitúa a España como el séptimo productor de ciruelas del mundo y el tercero a nivel europeo[1]. La importancia económica que tiene el cultivo de este fruto en nuestro país es evidente. Si nos centramos en la comunidad autónoma de Extremadura, que es donde se está desarrollando el estudio aquí presentado, el Ministerio de Agricultura, Pesca, Alimentación y Medio Ambiente Español, cifra en 6500Has el territorio dedicado a esta actividad. Una mejora en el proceso de recolección, que puede ser recolectar el fruto en su momento óptimo con las herramientas que aquí se presentan, puede suponer una gran ventaja competitiva de las Empresas Extremeñas sobre sus competidoras.

La calidad de un producto es percibida por el consumidor como un conjunto de atributos que son evaluados de forma subjetiva, con el fin de escoger el mejor. Si nos centramos en la fruta, esta calidad la mediremos por: apariencia, aroma, sabor, etc. Pero, ¿cómo obtenemos frutas de mayor calidad?, podemos utilizar técnicas para mejorar los procesos de cultivo o recolección del fruto en su momento óptimo. De esta forma, cuando llegue la fruta al consumidor, alcanzará la calidad deseada.

La clasificación de la ciruela por su estado de maduración es un proceso que se realiza de manera manual, lo que puede llevar a clasificaciones erróneas. Este proceso históricamente ha sido realizado por operarios humanos, agricultores o técnicos, que con la experiencia adquirida durante los años de trabajo, son capaces, de manera visual, de clasificar las ciruelas por su calidad. Este método tiene grandes limitaciones ya que las decisiones que se toman dependen en gran medida de la experiencia de éstos. Se trata pues de un método altamente subjetivo, que puede derivar en errores en la cosecha, ya sea por recoger el fruto antes de tiempo o incluso con una fecha posterior a su estado óptimo. Para intentar solucionar este problema y ayudar a los operarios del sector, se están introduciendo nuevas técnicas [1] que tienen como finalidad la de servir de apoyo en la correcta toma de decisiones. Entre estas técnicas novedosas se encuentran las técnicas basadas en visión por computador [3], [4] y algoritmos de *machine learning* [2]. Este tipo de algoritmos requieren de un proceso intensivo de aprendizaje, que una vez concluido, genera potentes clasificadores que ayudarán a los agricultores y técnicos a tomar la decisión más acertada en función a los parámetros óptimos de cosecha.

La correcta clasificación de las ciruelas por su estado de maduración es de gran importancia, debido principalmente a que ciertas propiedades internas del fruto (sólidos solubles, brix, firmeza...) están directamente relacionadas con su estado de maduración. Para poder conocer el estado de estas variables es necesario la utilización de técnicas análiticas en un laboratorio. Los principales inconvenientes de estas técnicas se resumen en la necesidad de la destrucción del fruto para conocer sus propiedades químicas y los complejos y costosos materiales que se utilizan para ello.

La incorporación al proceso de técnicas de *machine learning*

[1] http://www.fao.org/home/en/

por las industrias alimentarias en los procesos selectivos es habitual [2], aplicando desde bosques aleatorios (*random decision forests*) [5], máquinas de vectores de soporte [6] y redes neuronales en la evaluación de la calidad del producto [7].

El objetivo del trabajo que aquí se presenta es la detección de la variedad de la ciruela en su fase temprana de maduración, así como su estado de maduración. Las variedades seleccionadas en este estudio son:

- Black Splendor
- Angeleno
- OwenT

El análisis de imágenes se realizará a través de imágenes hiperespectrales de las diferentes variedades de ciruelas, divididas en diferentes semanas de maduración. El uso de imágenes hiperespectrales para el análisis de calidad en productos agroalimentarios es ampliamente utilizado [9], [11], obteniendo buenos resultados. Se pretende localizar espectros de frecuencia que nos permitan, a través de las propiedades fisicoquímicas que describen, clasificar las ciruelas por variedad y maduración. Los resultados serán comparados con técnicas previas donde únicamente se utilizaban imágenes tipo RGB [15].

El resto del trabajo se divide de la siguiente forma: la sección II describe el estado del arte de técnicas similares a las descritas en este trabajo. La sección III describe la metodología que se ha utilizado en este trabajo. Por último, los resultados obtenidos junto con las conclusiones, son presentados en las secciones IV y V, respectivamente.

II. Estado del arte

Entre los trabajos más destacados en esta línea de investigación se encuentra [1], el cual tiene como objetivo el estudio y detección de la maduración en los frutos con hueso.

La aplicación de técnicas de visión artificial para el análisis de los alimentos ha aumentado considerablemente en los últimos años [2]. La diversidad de las aplicaciones depende, entre otras cosas, del hecho de que los sistemas de visión artificial proporcionan información sustancial acerca de la naturaleza y los atributos de los objetos presentes en una escena. Otra característica importante de tales sistemas es que abren la posibilidad de estudiar estos objetos en las regiones del espectro electromagnético donde el ojo humano es incapaz de operar, como en el ultravioleta (UV), infrarrojo cercano (NIR) o infrarrojo (IR).

Investigadores de todo el mundo han estudiado el potencial de diversas técnicas para conocer la calidad de la fruta [4]. Una de las más utilizadas han sido las diferentes técnicas de espectroscópicas, como la NIR [13]. El éxito de la utilización de estas técnicas reside en las ventajas que aportan a los investigadores, estas son las siguientes:

- El proceso de medición es simple y rápido.
- Se trata de una técnica no destructiva.
- Permiten conocer varias propiedades de la fruta a la vez.

El inconveniente que tienen las técnicas de espectroscopia es que solamente nos aportan información de los componentes químicos y físicos de la fruta en el punto medido, para poder obtener más información, es necesario combinar esta técnica con la adquisición de imágenes. Por un lado, se tiene la información obtenida de la adquisición de la imagen del fruto, que se trata de una información espacial: morfología, tamaño, etc. Por otro lado, mediante la espectroscopia, se obtiene información sobre los componentes químicos y propiedades físicas que la componen. Sin embargo, si nos centramos en las técnicas espectrales de imágenes, nos permiten la obtención de imágenes de frutas e información espectral simultáneamente, con las ventajas de una alta resolución espectral y múltiples bandas de ondas. De acuerdo con la resolución espectral, la espectroscopia de imágenes se puede dividir en imágenes multiespectrales, imágenes hiperespectrales e imágenes ultraespectrales. Las imágenes multiespectrales y las imágenes hiperespectrales son factibles para la medición de los parámetros de calidad de la fruta [12].

Una nueva técnica ha surgido en los últimos años con fuerza en el campo del aprendizaje profundo (Deep learning en inglés) [14], se trata de las redes neuronales convolucionales (CNN en inglés) [7]. Estas redes basan su funcionamiento en un aprendizaje jerarquizado en el cual estructuras de alto nivel son construidas de manera automáticas a partir de estructuras de más bajo nivel llamadas capas, comenzado por los datos sin procesar: los píxeles de una imagen. Deep learning surge como una alternativa frente a una mayoría de técnicas de aprendizaje que están basadas solamente en una o, a lo sumo, dos capas de transformaciones no lineales de características.

El aprendizaje profundo a través de CNNs [14] es una alternativa a los métodos clásicos de clasificación que requerían de una cuidadosa selección de las características realizada a mano. Los métodos clásicos han demostrado ser bastante eficaces para resolver problemas simples o problemas bien delimitados, pero tropiezan con dificultades para hacerles frente con problemas complejos del mundo real tales como objetos y reconocimiento de voz. Es en estos problemas complejos donde el aprendizaje profundo está resultando ser verdaderamente efectivo, siendo el problema aquí presentado como un problema complejo de visión.

III. Metodología

Será necesaria la generación de un conjunto de imágenes hiperespectrales de los frutos que se van recolectando en diferentes semanas de maduración. Para ello, se requerirá el uso de hardware especializado que nos permita la captura de imágenes de las frutas en un entorno controlado de laboratorio. Así como un software específico de control para todo el hardware, el cual se describe a continuación:

- Cabina de iluminación Matcher Modelo MM-4e equipada con cuatro fuentes de luz: Simulador de luz diurna 6500K y de 5000K y una fuente ultravioleta para medir la fluorescencia si es necesario.
- Cámara hiperespectral Cuber UHD 285. Dicha cámara cuenta con un rango de longitud de onda comprendido entre los 450 - 950 nm con un intervalo de submuestreo cada 4 nm. Los fabricantes garantizan 125 canales de

información hiperespectral, aunque la cámara es capaz de ofrecer hasta los 138 canales.
- Plataforma giratoria diseñada y controlada por arduino para girar 90 grados cada pieza de fruta de forma automática.
- Software diseñado con MatLab para el control semi-automático de la plataforma giratoria y toma de imágenes hiperespectrales.

Para poder entender mejor el aspecto de las frutas capturadas y las mínimas diferencias que existen para el ojo humano, en la Figura 1 podemos observar un conjunto de imágnes RGB de las frutas empleadas en este estudio.

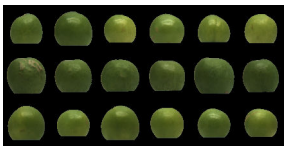


Figura 1. Ejemplo imágenes RGB. Angeleno, Black Splendor y OwenT

Las imágenes que captura la cámara hiperespectral son imáganes tipo CUE, que deben ser post-procesadas para obtener imágenes tipo PNG, utilizadas en este trabajo, que representan las 138 capas del espectro que ofrece la cámara. La Figura 2 muestra un ejemplo de ciertas bandas convertidas a imagen PNG.

Figura 2. Ejemplo imágenes procesadas fichero tipo CUE

Una vez creados los conjuntos de datos correspondientes, se han utilizado técnicas de aprendizaje profundo para generar modelos de clasificación basados en CNNs, en concreto se ha utilizado *Alexnet* [16], que a través de un proceso de optimización, se han obtenido los clasificadores destinados a detectar la variedad y la maduración de los frutos del ciruelo japonés.

El entranamiento de este tipo de redes requiere de una estructura muy específica que posibilita todo el proceso de aprendizaje de la red. En el trabajo aquí presentado se ha utilizado el framework llamado Caffe[2], que nos permite realizar el proceso de aprendizaje.

Todo proceso estocástico, como el que nos compete en este trabajo, requiere realizar el mismo proceso de aprendizaje un determinado número de veces, para poder consolidar los resultados obtenidos. Para ello, en este trabajo hemos utilizado la técnica llamada K-fold cross validation, donde K=5. Esta técnica nos obliga a dividir el conjunto de imágenes en 5 particiones, donde usará un conjuto compuesto por 4 de estas particiones para entrenar a la red y 1 para validar los resultados. El proceso debe repitese tantas veces como subconjuntos se hayan realizado, en nuestro caso 5. Posteriormente, con cada subconjunto se repite nuevamente un total de 6 veces para el ajuste de la red, obteniendo así un total de 30 ejecuciones, lo que nos permite consolidar los resultados que arroje la red, a través de la media de los mismos.

[2]http://caffe.berkeleyvision.org

IV. Resultados

Entre los objetivos citados en este estudio se encuentra el análisis de la información hiperespectral de las imágenes de fruta, para obtener las longitudes de onda más prometedoras para su clasificación por variedad y madurez. Esto nos permitirá desechar gran cantidad de información hiperespectral y centrarnos en las bandas más interesantes que, como se puede ver en esta sección, ofrecen resultados muy robustos en relación a la clasificación de la variedad de la fruta, así como sobre su maduración.

En este apartado se presentan los resultados obtenidos por los diferentes clasificadores que se han optimizado utilizando *Alexnet* como CNN base. La Tabla I muestra una descripción de los los diferentes *datasets* utilizados.

Tabla I
Características de las ciruelas

Dataset	Número de Imágenes[3]	Fecha de recolección	Semanas de maduración
MW1	121 x 138	9-13 Mayo	6 Angeleno, 7 Owent y 7 BlackSplendor
MW2	147 x 138	23-27 Mayo	8 Angeleno, 9 Owent y 9 BlackSplendor
MW3	127 x 138	13-17 Junio	11 Angeleno, 12 Owent y 12 BlackSplendor
MW4	130 x 138	4-9 of Julio	14 Angeleno, 15 Owent y 15 BlackSplendor
All_MW	525 x 138	-	-

El trabajo que se ha llevado a cabo ha constado de la optimización de 138 clasificadores diferentes, ya que cada uno de ellos se debía especializar en una de las bandas obtenidas de las imágenes tomadas con la cámara hiperespectral. Para la optimización de este trabajo se ha utilizado un equipo compuesto por una GPU Tesla K20.

A continuación se presentan los resultados obtenidos por los clasificadores especializados en la variedad de la fruta, así como los especializados en la maduración.

IV-A. Resultados clasificación por variedad

El objetivo de esta tarea es analizar si en diferentes estados de maduración, podemos encontrar diferentes longitudes de onda que ayuden a la clasificación por variedad de las ciruelas. Se presenta un conjunto de gráficas que resumen el resultado de estos clasificadores, así como se puede observar en cada uno de ellos la presencia de ciertas bandas espectrales que obtienen excelentes resultados de clasificación.

Los resultados presentados en la Fig. 3 se corresponden con las diferentes semanas de maduración indicadas en la Tabla I. Si analizamos los resultados atendiendo a su semana de maduración, puede observarse que para el *dataset MW1* todos

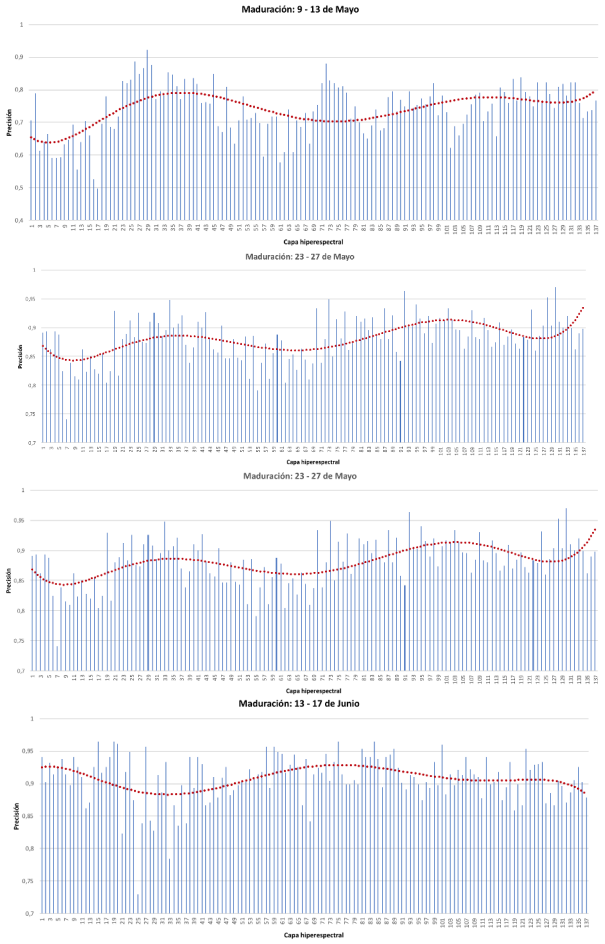


Figura 3. Resultado de clasificación por variedad. *Datasets*: arriba izquierda MW1, arriba derecha MW2, abajo izquierda MW3, abajo derecha MW4

los frutos se encuentran en una etapa muy temprana de su ciclo de maduración. Esto tiene como consecuencia que las ciruelas se encuentran poco desarrolladas y el parecido entre ellas es elevado, independiente de la variedad de las mismas. Como puede observarse en la figura, la línea de tendencia mostrada en los resultados, nos indica que entre las capas 25 y 35 se muestra una tendencia de mayor precisión en la clasificación, siendo la mejor de ellas la capa número 29 con una precisión de un 92,32 %.

Si atendemos a los datos del *dataset MW2*, se obtienen mejores resultados que con el *dataset MW1*. Esto es debido a que las frutas van avanzando en su proceso de maduración, lo que hace que se encuentren más desarrolladas y por tanto las diferencias entre variedades empiecen a ser palpables. Como puede observarse, atendiendo nuevamente a la línea de tendencia mostrada, las últimas capas, entre las 129 y 137, ofrecen los mejores resultados de clasificación por variedad, teniendo en cuenta que todas las capas se obtienen una precisión mayor al 75 %. En este proceso de optimización del clasificador por variedad, se ha alcanzado el óptimo en la capa 130 con un 96,95 % de aciertos.

A medida que avanzamos en la semana de maduración, *dataset MW3*, el clasificador mejora su precisión en todas sus capas respecto a los resultados obtenidos con los *datasets* anteriores. Casi todas las capas hiperespectrales del *dataset MW3* obtienen una precisión superior al 75 %. La línea de tendencia nos ayuda a confirmarlo. A diferencia de los *dataset* anteriores, en éste se encuentran varias capas con una precisión superior al 96,48 %.

Los resultados obtenidos con el último *dataset* utilizado, *MW4*, aportan buenos resultados globales de todas las capas en cuanto a precisión media para clasificar las variedades de ciruelo estudiadas. En esta última fase de maduración las diferencias entre las variedades son notorias a simple vista y esto permite que su clasificación sea más sencilla, es por ese motivo por el cual los resultados globales obtenidos son mejores. La línea de tendencia presente en la gráfica muestra que todas las capas obtienen buenos datos, estando la mayoría de las capas por encima del 80 %. El óptimo se encuentra en la capa 55 con un valor de precisión del 100,00 % en la clasificación por variedad.

Hasta aquí, podemos observar que a medida que la fruta madura, es más sencillo detectar su variedad, obteniendo un conjunto de capas hiperespectrales con mayor precisión. Estas características también pueden deberse a las diferentes fases de maduración que tienen las variedades que se han utilizado en este estudio. La variedad *Angeleno* es la que tiene mayor ciclo de maduración, de ahí que la fruta presenta menos cambios a lo largo del periodo estudiado, pero el resto si que presenta modificaciones, lo que permite a la red clasificar mejor las variedades.

En el estudio que se detalla a continuación, se han mezclado todas las fases de maduración de las diferentes variedades, para poder desestacionar la componente del tiempo de maduración. Con esto se pretende no facilitar a la red la detección gracias a grandes cambios en ciertas variedades y cambios pocos significativos en las variedades de ciclo largo. La Figura 4 muestra los resultados obtenidos de este nuevo estudio.

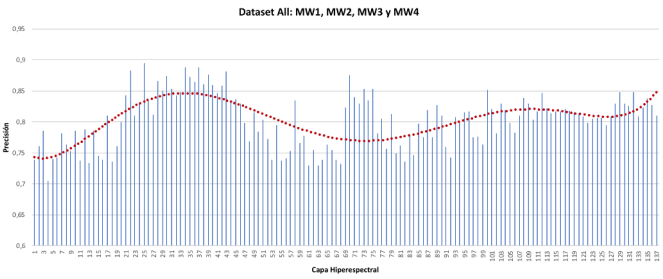


Figura 4. Resultado clasificación por variedad,*Dataset* ALL

El *dataset* utilizado donde la fecha de maduración del fruto no es relevante, arroja resultados del 70 % de precisión como mínimo. Si observamos la línea de tendencia presente en la Fig. 4 se aprecia que entre las capas 17 - 47 se obtienen muy buenos resultados, siendo la mejor capa la 25 con un 85 % de precisión. Este dato nos indica que es posible la clasificación de ciruelas por su variedad, gracias a la precisión obtenida del

85 %, siendo indiferente la fase de maduración en la que se encuentre el fruto.

La Tabla II muestra un resumen de las propuestas estudiadas con los diferentes *dataset* empleados. En esta tabla se resumen las capas que han obtenido mayores resultados, así como el intervalo de capas donde se aprecian diferencias con respecto al resto de capas.

Tabla II
Proposición capas para clasificador

Semanas de maduración	*Dataset*	Capas	% acierto máximo
9–13 Mayo	MW1	29	92,32 %
23–27 Mayo	MW2	92, 130	96,95 %
13–17 Junio	MW3	15, 19, 75, 84, 101	96,48 %
4–9 of Julio	MW3	55	100,00 %
-	All_MW	25	89,45 %

Podemos observar, según los datos presentados, que a diferentes fases de maduración de los frutos, la información relevante para poder clasificar las variedades según las imágenes hiperespectrales, se encuentran en diferentes bandas del espectro.

Por otro lado, también podemos observar que los resultados son excelentes, si nos centramos en la capacidad de clasificación de las redes optimizadas, ya que, independientemente de las capas, es posible generar una buena clasificación de las variedades empleadas.

Si comparamos los datos obtenidos en el estudio que aquí se presenta, con los datos obtenidos en estudios similares, pero utilizando imágenes RGB, podemos observar ciertas diferencias. Anteriormente se ha abordado la problemática de la clasificación de ciruelas por su variedad mediante el uso de la misma CNN, *Alexnet* [15], pero con la diferencia que en este trabajo, las imágenes utilizadas eran imágenes RGB. Esta circunstancia nos permite comparar ambos estudios, resumiendo los datos en la Tabla III.

Tabla III
Resumen de resultados utilizando imágenes RGB frente a imágenes hiperespectrales

Dataset	RGB	Hiperespectral - Número capa
MW1	0.8960 ± 0.010	0.9232 ± 0.060 - 29
MW2	0.9299 ± 0.015	0.9695 ± 0.040 - 130
MW3	0.9739 ± 0.008	0.9648 ± 0.030 - 19
MW4	0.9674 ± 0.005	1.0000 ± 0.010 - 55
All_MW	0.9071 ± 0.010	0.8945 ± 0.030 - 25

Como puede observarse, los resultados obtenidos al clasificar ciruelas usando imágenes hiperespectrales son superiores a los resultados obtenidos anteriormente, independientemente si lo que buscamos es un conjunto de capas que nos indiquen propiedades físico-químicas de las ciruelas que nos permitan realizar la clasificación, aunque si realizamos un test estadísitico para comparar ambas técnicas nos ofrece resultados no significativos, como indica la Tabla IV-A.

Tabla IV
Test de Wilcoxon para comapración de técnicas

VS	R^+	R^-	P-value	P-value Asintótico
Hiper	8.0	7.0	≥ 0.2	0.787406

Atendiendo a la capa que obtiene mejor resultado de clasificación, al utilizar imágenes hiperespectrales, y la misma red CNN -*Alexnet*-, los resultados son mejores que al utilizar imágenes RGB. Se podría afirmar que el estudio de la imágen hiperespectral, al contener propiedades físico-químicas del fruto, ofrece mejores resultados frente a una clasificación por imagen convencional, donde únicamente la forma, tamaño y color serían las componentes que intervienen en dicha clasificación.

IV-B. Resultados clasificación por estado de maduración

De igual forma que con los resultados presentados en la sección anterior, donde hemos obtenido buenos clasificadores que se centran en la variedad de la fruta analizada, en esta sección se presenta un nuevo estudio donde se optimizan clasificadores orientados a determinar la fecha de maduración de la misma.

En este caso, los *dataset* con los que se ha trabajado están agrupados por variedad y agrupados en 4 clases diferentes, cada una de las semanas de maduración cuando fueron recogidos los frutos. Se utilizan para este estudio nuevamente las imágenes hiperespectrales, donde se intenta encontrar un conjunto de bandas que nos permitan clasificar las frutas por su estado de maduración.

La Figura 5, muestra los resultados obtenidos.

Si observamos los resultados obtenidos por el clasificador de la variedad Black Splendor por semana de maduración, se observa que existe una gran diferencia entre las diferentes capas de las imágenes hiperespectrales. Como se puede ver en la Figura 5, la precisión obtenida se puede dividir en tres bloques que son claramente diferenciables. Un primer bloque, de la capa 1 - 65, con un buen índice de precisión, entorno al 96 %. El segundo bloque, rango comprendido entre la capa 66 - 113, donde la precisión decae hasta valores cercanos al 40 %. Un último bloque, 114 - 137, donde la precisión se sitúa de nuevo en el 96 %.

Por otro lado, los resultados obtenidos al clasificar imágenes hiperespectrales de la variedad *OwenT* muestran, que al igual que sucedía con los resultados obtenidos en la variedad *Black Splendor*, sus resultados pueden dividirse en tres bloques. En el primer bloque, que comprende de la capa 1 hasta la capa 17 tienen una precisión cercana al 98 %. El segundo bloque, capa 18 - 57, obtiene mala precisión, algunas de sus capas tienen una precisión del 25 %. De la capa 58 a la 137 la precisión crece situándose en un rango comprendido entre el 80 % y el 98 %.

Por último, a diferencia de lo que ocurre con las variedades *OwenT* y *Black Splendor* los resultados obtenidos al clasificar la variedad *Angeleno* por maduración son muy similares en todas sus capas. La precisión en ninguna de las capas es

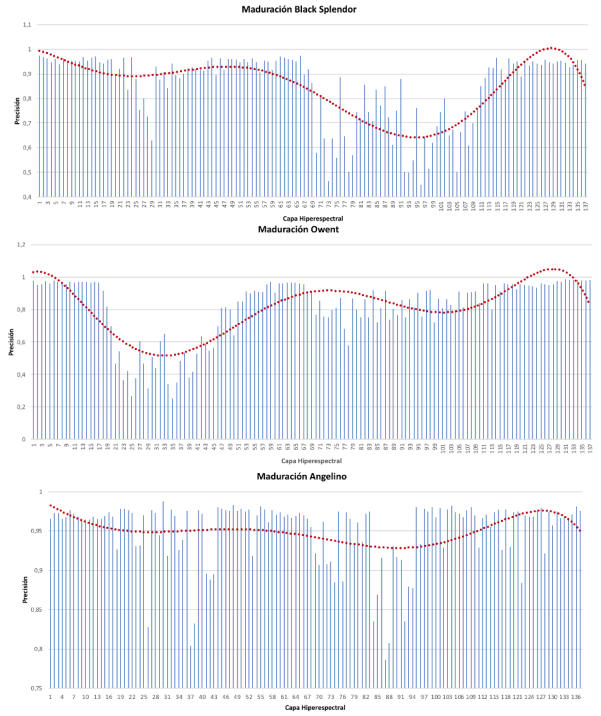


Figura 5. Clasificación por maduración: *Black Splendor*, *OwenT*, *Angeleno*.

inferior al 80 %. La capa que ofrece mejor precisión es la capa 30 con un 98,78 %. Esto puede ser debido a que esta variedad es una variedad de ciclo muy largo, y las fechas en las que se ha recolectado los frutos, debido a su ciclo de maduración, no son lo suficientemente significativas en el tiempo.

A continuación, en la tabla V se muestra un resumen de las capas que obtienen los mejores resultados para cada una de las variedades.

Tabla V
PROPOSICIÓN CAPAS PARA CLASIFICAR POR ESTADO DE MADURACIÓN

Variedad	Rango de capas	% máximo de acierto
Angeleno	94 - 137	96 %
Black Splendor	1 - 65 / 114 - 137	98 %
OwenT	1 - 17 / 58 - 137	98,78 %

V. Conclusiones

Los resultados presentados en este trabajo muestran que es posible clasificar, con un alto rango de acierto, las variedades de ciruelo japonés seleccionadas para este estudio por medio de las imágenes hiperespectrales tomadas en un entorno de laboratorio. Los datos obtenidos demuestran que es posible clasificar la variedad de la ciruela atendiendo a su información hiperespectral y la semana de maduración en la que se encuentra, se han obtenido resultados del 92,32 %, 96,95 %, 96,48 % y 100,00 % respectivamente, para las cuatro fases de maduración empleadas en este estudio. Por otro lado, si la fase de maduración no se tiene en cuenta, se obtiene unos resultados del 89,45 %. Estos datos superan un estudio previo donde se emplearon imágenes RGB, permitiendo obtener clasificadores más eficaces.

Estos resultados nos permiten afirmar que es posible clasificar la variedad de una ciruela gracias al estudio de su espectro. Además, los datos obtenidos y presentados, nos permiten vislumbrar ciertas zonas de interés en el espectro de las diferentes variedades, atendiendo a su fecha de maduración, donde, con un estudio más profundo, se podrán obtener propiedades físico-químicas de los frutos a través de su espectro.

Agradecimientos

Agradecemos el apoyo del Ministerio de Economía y Competitividad proyecto TIN2017-85727-C4-{2,4}-P, Junta de Extremadura, Consejería de Comercio y Economía, proyecto IB16035 a través del Fondo Europeo de Desarrollo Regional, "Una manera de hacer Europa".

Referencias

[1] Wang, H., Peng, J., Xie, C., Bao, Y., He, Y. *"Fruit quality evaluation using spectroscopy technology: a review,"* Sensor 15(5), 11889-11927, 2015.

[2] Sergio Cubero, Nuria Aleixos, Enrique Moltó, Juan Gómez-Sanchis, and Jose Blasco. *"Advances in machine vision applications for automatic inspec- tion and quality evaluation of fruits and vegetables,"* Food and Bioprocess Technology, 4(4):487–504, 2011.

[3] M.T. Riquelme, P. Barreiro, M. Ruiz-Altisent, and C. Valero. *"Olive classification according to external damage using image analysis,"* Journal of Food Engineering, 87(3):371–379, 2008.

[4] P.B. Pathare, U.L. Opara, and F. A. J. Al-Said. *"Colour measurement and analysis in fresh and processed foods: A review,"* Food and Bioprocess Technology, 6(1):36–60, 2013.

[5] Pal, M. *"Random forest classifier for remote sensing classification,"* International Journal of Remote Sensing, 26(1), 217-222. 2005.

[6] Danfeng Wang, Xichang Wang, Taiang Liu and Yuan Liu, *"Prediction of total viable counts on chilled pork using an electronic nose combined with support vector machine,"* Meat Science, Volume:90, pag:373 - 377, 2012.

[7] Krizhevsky, A., Sutskever, I., Hinton *"Imagenet classification with deep convolutional neural networks. In: Bartlett, P., Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.),"* Advances in Neural Information Processing Systems 25, pp. 1106–1114, 2012

[8] C. Yang, W.S. Lee, and P. Gader. *"Hyperspectral band selection for detecting different blueberry fruit maturity stages,"* Computers and Electronics in Agriculture, 109:23–31, 2014.

[9] Lu, R., Ariana, D. P. *"Detection of fruit fly infestation in pickling cucumbers using a hyperspectral reflectance/transmittance imaging system,"* Postharvest Biology and Technology, 81, 44-50, 2013.

[10] Haff, R. P., Saranwong, S., Thanapase, W., Janhiran, A., Kasemsumran, S., Kawano, S. *"Automatic image analysis and spot classification for detection of fruit fly infestation in hyperspectral images of mangoes,"* Postharvest Biol. Technol, 86, 23-28, 2013

[11] Dubey, S. R., Jalal, A. S. *"Adapted approach for fruit disease identification using images,"* arXiv preprint arXiv:1405.4930, 2014

[12] Hailong Wang, Jiyu Peng, Chuanqi Xie, Yidan Bao y Yong He *"fruit quality evaluation using spectroscopy",* Sensor, 21 mayo 2015

[13] S. Edward Law. *"Scatter of near-infrared radiation by cherries as a means of pit detection,"* Journal of Food Science, 38(1):102 – 107, 1973.

[14] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. *"Deep learning"* Nature 521, no. 7553 436-444. 2015

[15] Rodríguez, Francisco J., García, Antonio, Pardo, Pedro J., Chávez, Francisco, Luque-Baena, Rafael M., *"Study and classification of plum varieties using image analysis and deep learning techniques"* Progress in Artificial Intelligence, 7(2), 119-127, 2018

[16] Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E., *"ImageNet Classification with Deep Convolutional Neural Networks"* NIPS'12, 1097–1105, 2012

Preprocesamiento guiado por luminosidad para la detección automática de armas blancas en video vigilancia con Deep Learning

Alberto Castillo, Siham Tabik , Francisco Pérez, Roberto Olmos, and Francisco Herrera
*Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain.
Email: albertocl@decsai.ugr.es, siham@ugr.es, fperezhernandez@ugr.es, herrera@decsai.ugr.es

***Resumen*—La detección automática de armas blancas empuñadas por una o varias personas presentes en los vídeos de vigilancia pueden ayudar a reducir los delitos. Sin embargo, la detección de este tipo de objetos en vídeos se enfrenta a varios problemas, tales como la producida por la variabilidad de la luz ambiental junto con la la reflectancia de la superficie de las armas blancas. El objetivo de este trabajo es doble: i) Elaborar un modelo de detección automático de armas blancas para la videovigilancia mediante redes neuronales convolucionales (CNN) y ii) reforzar su robustez frente a diferentes condiciones lumínicas proponiendo una metodología de preprocesamiento guiado por luminosidad llamada DaCoLT (*Darkening and Contrast at Learning and Test stages*) para abordar condiciones de luminosidad perjudiciales.**

I. Introducción

Según la Organización Mundial de la Salud [1], cada año mueren más de 15,000 personas en crímenes violentos. Alrededor del 40 % por ciento de estos homicidios se cometen con navajas y armas blancas punzantes. En la videovigilancia, los agentes de seguridad tienen que detectar visualmente la presencia de armas en escenas monitorizadas y tomar decisiones en muy poco tiempo. Una de las soluciones más efectivas ante este problema es equipar las cámaras de vigilancia con un sistema automático y preciso de detección de armas blancas.

La mayoría de los estudios previos abordaron la detección de armas en rayos X, imágenes milimétricas o RGB utilizando métodos clásicos de aprendizaje automático [5], [6], [15], [16], [17]. Actualmente, los modelos de detección de objetos más precisos son basado en técnicas de Deep Learning, particularmente modelos basados en CNNs. El primer trabajo para abordar la detección de armas en vídeos utilizando CNNs fue [11]. Este trabajo se centró en las pistolas y fue evaluado en vídeos de películas de los noventa.

Por lo que sabemos, el presente estudio es el primero en desarrollar un sistema de detección de armas blancas usando Deep Learning y abordando el problema de la luminosidad aplicado en vídeos de vigilancia grabados en escenarios interiores. La detección de armas blancas en los vídeos de vigilancia en escenas de interior afronta varios desafíos:

- Las armas blancas pueden manejarse de diferentes formas y una gran parte del arma puede ser ocluida. Además, las armas blancas comunes, como los cuchillos, son pequeñas y la distancia entre el cuchillo y la cámara puede ser grande, lo que hace que la detección sea más difícil.
- El proceso de diseño de un nuevo conjunto de datos para entrenar con éxito el modelo de detección es manual y lleva mucho tiempo.
- La detección es sensible a la luz ambiental ya que, en general, las armas blancas, como los cuchillos, tienen superficies reflectantes.

Nos enfocamos en detectar con precisión los tipos más usados de armas blancas en delitos. Construimos un nuevo conjunto de datos que permite que el modelo aprenda con éxito las características distintivas de las armas blancas. A continuación, desarrollamos un modelo de detección de armas blancas apropiado para escenarios interiores y bajo diferentes condiciones de luz. Estudiamos las condiciones de luminosidad que afectan al rendimiento de la detección y proponemos una nueva metodología de preprocesamiento para solucionar los problemas de luminosidad.

Las principales contribuciones de este trabajo pueden resumirse como sigue:

- Construir una nueva base de datos etiquetada para detección de armas blancas, guiado por el proceso de clasificación .
- Analizar la mejor combinación de clasificadores basados en CNN y técnicas de selección de región para la detección automática de armas blancas en vídeos de vigilancia en escenarios interiores.
- Proponer una nueva metodología de preprocesamiento guiada por luminosidad, llamada Darkening and Contrast at Learning and Test time (DaCoLT), para superar las condiciones de luminosidad perjudiciales.

Nuestro estudio experimental muestra que el modelo de detección más preciso entrenado en nuestra nueva base de datos es R-FCN basado en ResNet-101, proporciona una medida F1 de 93 %. El F1 obtenido por el modelo en diferentes condiciones de luminosidad empeoraron hasta en un 15 % y usando nuestra metodología DaCoLT lo redujimos al 3 %.

Este documento está organizado de la siguiente manera. La sección II da un breve análisis de los estudios de investigación

[1] http://www.euro.who.int/__data/assets/pdf_file/0012/121314/E94277.pdf

más relacionados. La sección III describe el procedimiento para construir nuestra nueva base de datos de calidad para detección. La sección IV selecciona el modelo más adecuada para su uso como sistema de detección automática. Sección V analiza el rendimiento de detección en diferentes condiciones de luminosidad y propone la metodología DaCoLT. Finalmente, las conclusiones se resumen en la sección VI.

II. Trabajos relacionados

El problema de detectar un cuchillo empuñado por una persona en videovigilancia está estrechamente relacionado con (i) la detección de objetos pequeños en imágenes y ii) detección general de objetos mediante modelos de aprendizaje profundo.

El área tradicional de detección de armas en imágenes ha utilizado a menudo métodos clásicos supervisados de aprendizaje automático que requieren un alto nivel de supervisión humana, por ejemplo, FAST [2], SIFT [5], AAM [6], Harris [15]. Los medios utilizados son principalmente rayos X o imágenes milimétricas [16], [17] para armas ocultas y RGB para armas visibles [2], [6], [7]. En general, estos métodos proporcionan buenas precisiones pero sufren de varias limitaciones, son invasivas, necesitan costosas sistemas de detección de metales [5] como los sistemas utilizados en el acceso al aeropuerto, no puede detectar múltiples armas [9], [15] y son lentos para usar en sistemas de detección en tiempo real [2].

Los modelos de detección de objetos de última generación se basan en Convolutional Neural Networks y muestran resultados prometedores en los dos desafíos de detección más prestigiosos. El modelo de detección más preciso de ILSVRC 2017 (Large Scale Visual Recognition Challenge) [4] alcanzó una precisión media de alrededor del 73 % [2] en un benchmark de 527892 imágenes dispuestas en 200 clases de objetos, con un promedio de 2500 imágenes por clase. El modelo de detección más preciso en el benchmark de detección de 80 objetos comunes Common Objects in Context (COCO) [10] también alcanzó una precisión media de alrededor del 73 %. Los rendimientos más altos en COCO tiene una precisión del 60 % y un recall del 80 % pero fueron obtenidos en objetos grandes, y de menor rendimiento con una precisión del 30 % y un recall del 50 % en objetos pequeños [3].

Por lo que sabemos, el primer sistema automático de detección de armas de fuego basado en Deep Learning fue [11]. Este trabajo demostró ser preciso en las películas (descargadas de YouTube) con mejor calidad, es decir, mejor resolución, contraste y luminosidad que los vídeos comunes de vigilancia. Los mejores resultados reportados en este trabajo fueron obtenidos por el modelo de detección Faster R-CNN [12] basado en VGGNet [14] y con una velocidad de cinco fotogramas por segundo (fps), que es una tasa baja para un sistema de tiempo real.

[2]http://image-net.org/challenges/LSVRC/2017/
[3]http://cocodataset.org/#detections-leaderboard

III. Procedimiento de construcción de la base de datos para la detección de armas blancas

Nuestro objetivo es construir una base de datos que permita al modelo de detección distinguir con precisión entre cuchillos y todos los objetos que puedan confundirse con cuchillos. Con este fin, primero comenzamos con un conjunto de datos de clasificación inicial, Database-1, y lo ampliamos progresivamente con nuevas clases de objetos para mejorar el número de true positives (#TP), falses positives (#FP), true negatives (#TN) y false negatives (#FN) producidos por un modelo de clasificación simple (VGG-16). Este análisis nos permite entender qué objetos son críticos en el proceso de aprendizaje y considerarlos como objetos en el fondo de las imágenes de la base de datos a la hora de construir la base de datos final para detección.

Extendimos la base de datos en tres pasos:

- Database-1 incluye 2 clases, la clase de cuchillos contiene imágenes de cuchillos de diversos tamaños y la otra clase con diversos fondos.
- Database-2 contiene 28 clases e incluye nuevas clases de objetos que a menudo están presentes como fondo en la clase cuchillos de Database-1.
- Database-3 incluye clases de objetos que pueden manejarse de forma similar a un cuchillo, por ejemplo, bolígrafo, o teléfono móvil, ver cuatro ejemplos en la figura 1.

Las imágenes utilizadas para construir la Database-1, -2 y -3 fueron descargadas de diversos sitios web. Las características de las tres bases de datos auxiliares, Database-1, 2 y 3, se muestran en la Tabla I

Para evaluar el rendimiento de la clasificación y detección sobre las bases de datos propuestas, hemos construido dos conjuntos de pruebas, Test-clas y Test-det.

- Test-clas se utiliza para evaluar el modelo de clasificación, consta de 512 imágenes, 260 imágenes contienen la clase cuchillo y 252 imágenes contienen otras clases de objetos.
- El Test-det se utiliza para evaluar los modelos de detección, contiene 388 imágenes, 378 contienen al menos un cuchillo. Test-det incluye fotogramas tomados por una cámara IP de videovigilancia (Hikvision DS-2CD2420F-IW 1080p para vídeo, ratio de frames 30 fps, campo de visión 95^o y compresión MJPEG).

Figura 1: Imágenes de ejemplo de cuatro clases de objetos de la base de datos 3, (a) clase cuchillo, (b) clase bolígrafo, (c) clase teléfono móvil y (d) clase cigarrillos.

(a) (b) (c) (d)

Usamos Keras API 2.0.4 [3] para los experimentos. Medimos el rendimiento, precisión, recall, y F1, obtenidos por el

Tabla I: Características de las bases de datos.

Database-	clases	total img	img arma	otras img	enfoque
1	2	1654	598	1056	clasificación
2	28	5538	598	4940	clasificación
3	100	10039	618	9421	clasificación
4	1	1250	1250	-	detección
Test-clas	-	512	260	252	clasificación
Test-det	-	388	378	10	detección

Tabla II: Resultados del modelo de clasificación para la clase cuchillo.

Database-	#TP	#FN	#TN	#FP	Precisión	Recall	F1 score
1	181	79	174	78	69,88 %	69,62 %	69,75 %
2	209	51	228	24	89,70 %	80,38 %	84,78 %
3	213	47	228	24	**89,87 %**	**81,92 %**	**85,71 %**

modelo de clasificación cuando se entrena en Database-1, -2 y -3 se muestra en la Tabla II. El rendimiento de la clase de cuchillo ha aumentado al ampliar el conjunto de datos con más clases de objetos. El mejor rendimiento se obtiene cuando el modelo es entrenado en Database-3, pero no puede ser usada directamente para entrenar el modelo de detección ya que el detector requiere una estrategia de anotación diferente.

Como paso final, construimos el conjunto de entrenamiento, Database-4, teniendo en cuenta todas las clases de objetos, de Database-1,-2 y -3, que mejoran el aprendizaje porque se manejan de la misma manera que un cuchillo o tienen características similares a las de un cuchillo. A diferencia de la clasificación de imágenes, el proceso de anotación para la detección requiere indicar la clase de objeto utilizando un cuadro delimitador. Consideramos dos clases, el cuchillo como la clase verdadera y el resto de objetos como fondo. Incluimos imágenes de i) armas blancas de diversos tipos, formas, colores, tamaños y hechos de diferentes materiales ii) cuchillos ubicados cerca y lejos de la cámara, iii) cuchillos ocluidos parcialmente por la mano, iv) objetos que pueden ser empuñados de la misma manera que los cuchillos y v) imágenes capturadas en escenarios de interior y exterior, conjunto de datos de 1250 imágenes. Figura 2 muestra ejemplos de Database-4.

Las imágenes utilizadas para construir esta base de datos fueron descargadas de Internet, algunos fotogramas fueron extraídos de vídeos de Youtube y vídeos de vigilancia. En el resto del trabajo usaremos Database-4 para entrenar el modelo de detección.

Figura 2: Imágenes de ejemplo de Database-4. Estas imágenes muestran un contexto más rico.

IV. Análisis del enfoque de Deep Learning para la detección de armas blancas

En esta sección, analizamos el rendimiento de varias combinaciones de los modelos de clasificación más avanzados y algoritmos de selección de regiones con el objetivo de encontrar el mejor modelo de detección para la videovigilancia. En particular, analizamos estas combinaciones:

- SSD basado en Inception-v2
- R-FCN basado en ResNet101
- Faster R-CNN basado en: Inception-ResNetV2, ResNet50, ResNet101 y Inception-V2

Todos los modelos de clasificación y detección se construyeron utilizando TensorFlow [1]. Para evaluar la detección usamos Tensorflow Object Detection API [8]. Todos los experimentos fueron llevados a cabo en una GPU NVIDIA Titan Xp.

Todos los modelos de detección fueron inicializados usando los pesos pre-entrenados en el conjunto de datos COCO integrado por más de 200.000 imágenes etiquetadas. Utilizamos fine-tuning mediante el entrenamiento de las dos últimas capas completamente conectadas de la red. El proceso de entrenamiento dura de tres a cuatro horas.

Tabla III: Analisis comparativo de los modelos de detección del estado del arte.

Detector	modelo base CNN	#TP	#FP	Precisión	Recall	F1	fps
Faster R-CNN	Inception-ResNetV2	345	0	100 %	91,27 %	**95,44 %**	1,3
Faster R-CNN	ResNet101	332	8	97,65 %	89,73 %	93,52 %	4,8
Faster R-CNN	Inception-V2	329	3	99,1 %	87,04 %	92,64 %	12,8
Faster R-CNN	ResNet50	326	2	99,39 %	86,24 %	92,35 %	4,4
R-FCN	ResNet101	335	0	100 %	88,62 %	93,97 %	10
SSD	InceptionV2	245	0	100 %	64,81 %	78,65 %	20,4

El rendimiento de los modelos de detección se mide en términos de true positives, false positives, precision, recall, F1 y tasa de tiempo de inferencia (frames per second). El entrenamiento y el test se llevaron a cabo en Database-4 y test-det respectivamente. En general, los modelos de detección logran un alto rendimiento como se puede ver en la Tabla III. Esto se explica por el hecho de que el aprendizaje transferido de COCO ha sido muy beneficioso para el proceso de aprendizaje, ya que COCO incluye la clase cuchillos compuesta por unas 8.500 imágenes. Al centrarnos en la videovigilancia, la detección debe ser precisa y rápida al mismo tiempo. Por lo tanto, seleccionamos R-FCN_ResNet101 para construir nuestro detector de armas blancas. Utilizando 100 regiones de interés R-FCN_ResNet101 logra una buena precisión $100\,\%$, recall $88,62\,\%$ y F1 $93,97\,\%$, lo que está cerca del mejor modelo y proporciona una tasa razonable de inferencia.

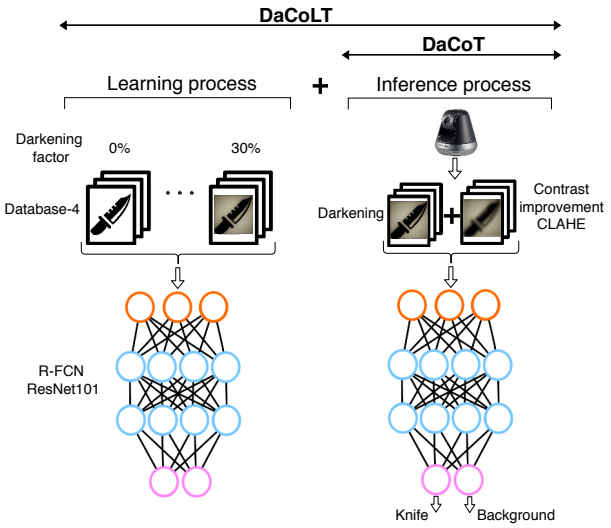
Todo el proceso de detección utiliza R-FCN_ResNet101 en una resolución por fotograma de Full HD, 1920×1080-píxeles, que es dos veces más rápido que el detector de pistola propuesto en [11]. Esto permite que el detector de armas blancas pueda ser usado en tiempo real para la videovigilancia.

V. Preprocesamiento guiado por luminosidad: metodología DaCoLT

Para resolver los problemas de luminosidad, proponemos dos alternativas:

- DaCoT: durante el proceso de prueba, los fotogramas con luminosidad alta son oscurecidos por un factor específico, luego su contraste es mejorado usando el algoritmo CLAHE.
- DaCoLT: durante el proceso de aprendizaje, el modelo de detección se entrena utilizando una técnica específica de *d*ata augmentation para oscurecimiento. Luego, DaCoT se aplica durante el tiempo de prueba. La diferencia entre estos dos procedimientos se ilustra en la figura 3.

Figura 3: Una ilustración de nuestro procedimiento, DaCoT se aplicó en el tiempo de test y DaCoLT se aplicó tanto en el tiempo de aprendizaje como en el tiempo de test.

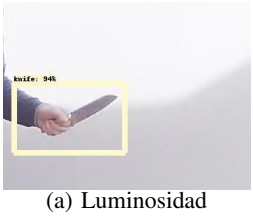


En particular, primero determinamos el rango de luminosidad que afecta la calidad de la detección (sección V-A), a continuación, se analiza y mejorar la detección simulando las condiciones ideales de luminosidad en test(sección V-B) y tanto en tiempo de aprendizaje como test(sección V-C).

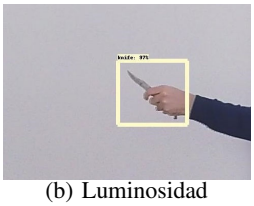
V-A. Análisis del impacto de la luminosidad en el redimiento de detección

A continuación, analizamos el impacto de las condiciones de luminosidad en el rendimiento del modelo de detección R-FCN_ResNet101. Usamos doce vídeos de prueba grabados con una cámara de seguridad IP, Samsung SNH-V6410PN de resolución 1080p, frame rate 30fps y amplitud de vista 96.1^o. Los vídeos de prueba se dividen en cuatro grupos de diferentes condiciones de luminosidad, luminosidad alta, luminosidad media, luminosidad baja y luminosidad artificial. Para una comparación justa, todos los vídeos muestran a la misma persona repitiendo las mismas acciones al mismo tiempo y distancia de la cámara. Todos los vídeos fueron grabados usando la misma cámara en la misma escena interior. Los vídeos de prueba incluyen tres cuchillos comunes con diferentes tamaños, pequeño, mediano y grande. Pequeño, mediano y grande se refieren a la proporción de la parte no ocluida del arma. Ver ejemplos en la figura 4. Los vídeos de test se pueden encontrar a través de este repositorio en github [4].

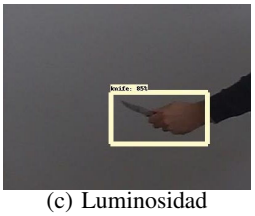
Figura 4: Resultados de detección en cuatro condiciones de luminosidad diferentes.



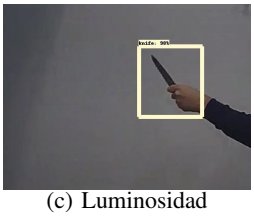
(a) Luminosidad alta



(b) Luminosidad media



(c) Luminosidad baja



(c) Luminosidad artificial

Consideramos que un cuchillo es un ground truth cuando es reconocible por el ojo humano. Los resultados en términos de número total de Ground Truth positivos #GT_P, #TP, #FP, precisión, recall, y F1 en cada video de prueba se muestran en la Tabla IV. Consideramos una detección como TP si el solapamiento entre el área del cuchillo manipulada en el fotograma y la caja delimitadora predicha es mayor que 70 %.

Tabla IV: Rendimiento de detección obtenido en vídeos grabados en diferentes condiciones de luminosidad.

Luminosidad	Tamaño arma	#frames	#GT_P	#TP	#FP	Precisión	Recall	F1
Alta	Grande	121	112	78	0	100 %	69,64 %	82,10 %
	Mediano	107	90	44	0	100 %	48,89 %	65,67 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	56,66 %	71,91 %
Media	Grande	109	98	85	0	100 %	86,73 %	92,89 %
	Mediano	116	98	73	0	100 %	74,49 %	85,38 %
	Pequeño	138	110	64	0	100 %	58,18 %	7356 %
				Promedio		100 %	73,13 %	83,94 %
Baja	Grande	126	114	104	1	99,05 %	92,04 %	95,41 %
	Mediano	114	100	70	0	100 %	70 %	82,35 %
	Pequeño	138	101	74	0	100 %	73,27 %	84,57 %
				Promedio		99,68 %	78,44 %	87,44 %
Artificial	Grande	119	110	95	0	100 %	86,36 %	92,68 %
	Mediano	113	99	75	3	96,15 %	78,13 %	86,21 %
	Pequeño	96	90	65	4	94,20 %	75,58 %	83,87 %
				Promedio		96,78 %	**80,02 %**	**87,59 %**

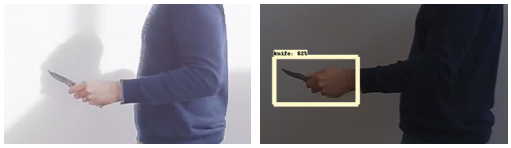
Como se puede observar en la Tabla IV, el rendimiento del modelo de detección es inestable en un escenario de cambio de luminosidad. El peor rendimiento se obtiene en condiciones de alta luminosidad, y el mejor rendimiento con luminosidad artificial. De las condiciones de luminosidad más bajas a las

[4]https://github.com/alcasla/Automatic-Cold-Steel-Detection-Alarm

más altas, el recall promedio disminuyó de 80,02 % a 56,66 %, y el F1 promedio de 87,59 % a 71,91 %.

La Figura 5 muestra un ejemplo de los resultados de detección de escenas muy similares, es decir, la misma pose y contexto, diferentes niveles de luminosidad y contraste, pero diferentes resultados de detección.

Figura 5: Un ejemplo del resultado de la detección en dos situaciones similares con diferentes condiciones de luminosidad.



Luminosidad alta Luminosidad baja

V-B. Darkening and Contrast at Test stage (DaCoT)

Para resolver la inestabilidad del modelo de detección en condiciones de luminosidad variable, primero analizamos el procedimiento llamado DaCoT, Oscurecimiento y Contraste en tiempo de test, el cual simula la condición de luminosidad que produce el mejor desempeño, luminosidad baja y alto contraste. Procedemos de la siguiente manera:

- Primero comprobamos el nivel de luminosidad de cada fotograma. Si el nivel de luminosidad es de medio a alto, oscureceremos el fotograma multiplicando los valores de los píxeles por el factor de oscurecimiento correspondiente. Este factor se calcula en base a la diferencia entre el nivel de luminosidad ideal y el nivel de luminosidad actual de la imagen.
- A continuación, aumentamos el contraste del fotograma obtenido mediante el algoritmo *Contrast-Limited Adaptive Histogram Equalization*(CLAHE) [13].
- A continuación, el fotograma se introduce al modelo de detección para inferencia.

La evaluación del enfoque propuesto en condiciones de luminosidad alta al considerar diferentes factores de oscurecimiento se proporciona en la Tabla V. El rendimiento del modelo de detección ha mejorado cuando se utiliza un factor de oscurecimiento del 30 %. En promedio, con un factor de oscurecimiento de 30 % el recall y F1 han mejorado en 6,53 % y 5,07 % respectivamente en comparación con la condición original de luminosidad alta.

El preprocesamiento propuesto, oscurecimiento más CLAHE, tarda alrededor de 29 ± 3 ms por fotograma en la CPU, lo que no ralentiza el proceso general de detección, ya que esta tarea de preprocesamiento se realiza en paralelo con la tarea de detección en la GPU. Es decir, la hebra de preprocesamiento se ejecuta en la CPU y la de detección se ejecuta en la GPU.

V-C. Darkening and Contrast at Learning and Test stages (DaCoLT)

Del análisis anterior, encontramos que el DaCoT mejora el rendimiento del modelo de detección bajo condiciones de luminosidad alta. En esta sección, analizamos el uso de DaCoLT, que es una extensión del DaCoT, mediante la aplicación de diferentes niveles de oscurecimiento no sólo en la etapa de prueba sino también durante la etapa de aprendizaje del modelo de detección. El método de aumento de datos de oscurecimiento consiste en oscurecer imágenes de entrenamiento individuales seleccionando aleatoriamente un factor de oscurecimiento en el rango [0 % 30 %].

Tabla V: Los resultados de los fotogramas de vídeo grabados originalmente en condiciones de luminosidad alta(es decir, en el peor de los casos) al aplicar DaCoT.

Factor oscurecimiento	Tamaño arma	#frames	#GT_P	#TP	#FP	Precisión	Recall	F1
original luminosidad alta	Grande	121	112	78	0	100 %	69,64 %	82,11 %
	Mediano	107	90	44	0	100 %	48,89 %	65,67 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	56,66 %	71,91 %
10 %	Grande	121	112	81	0	100 %	72,32 %	83,94 %
	Mediano	107	90	52	0	100 %	57,78 %	73,24 %
	Pequeño	137	103	57	1	98,28 %	55,34 %	71,25 %
				Promedio		99,43 %	61,81 %	76,14 %
20 %	Grande	121	112	83	0	100 %	74,11 %	85,13 %
	Mediano	107	90	55	0	100 %	61,11 %	75,86 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	62,23 %	76,31 %
30 %	Grande	121	112	85	0	100 %	75,89 %	86,29 %
	Mediano	107	90	56	0	100 %	62,22 %	76,71 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	**63,19 %**	**76,98 %**
40 %	Grande	121	112	80	0	100 %	71,43 %	83,33 %
	Mediano	107	90	52	0	100 %	57,78 %	62,60 %
	Pequeño	137	103	51	0	100 %	49,51 %	66,23 %
				Promedio		100 %	59,57 %	70,72 %
50 %	Grande	121	112	78	0	100 %	69,64 %	82,11 %
	Mediano	107	90	41	0	100 %	45,56 %	65,67 %
	Pequeño	137	103	50	0	100 %	48,54 %	65,36 %
				Promedio		100 %	54,58 %	71,04 %

Tabla VI: Los resultados al aplicar DaCoT y DaCoLT en vídeos grabados originalmente bajo condiciones de luminosidad alta usando diferentes tamaños de cuchillos, grande, mediano y pequeño.

	Tamaño arma	#frames	#GT_P	#TP	#FP	Precisión	Recall	F1
original luminosidad alta	Grande	121	112	78	0	100 %	69,64 %	82,11 %
	Mediano	107	90	44	0	100 %	48,89 %	65,67 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	56,66 %	71,91 %
guiado lumino. DaCoT (Test)	Grande	121	112	85	0	100 %	75,89 %	86,29 %
	Mediano	107	90	56	0	100 %	62,22 %	76,71 %
	Pequeño	137	103	53	0	100 %	51,46 %	67,95 %
				Promedio		100 %	63,19 %	76,98 %
guiado lumino. DaCoLT (Train+Test)	Grande	121	112	84	0	100 %	75 %	85,71 %
	Mediano	107	90	64	0	100 %	71,11 %	83,12 %
	Pequeño	137	103	74	0	100 %	71,84 %	83,61 %
				Promedio		100 %	**72,65 %**	**84,15 %**

La Tabla VI muestra el impacto al aplicar DaCoLT en el rendimiento de detección para las peores condiciones de luminosidad. La primera parte muestra los resultados del modelo de

detección en vídeos filmados originalmente en condiciones de luminosidad alta y utilizando diferentes tamaños de cuchillos, grandes, medianas y pequeñas. La segunda parte muestra el efecto de aplicar el enfoque de preprocesamiento propuesto en la etapa de inferencia, DaCoT. La tercera parte muestra el efecto de la metodología DaCoLT propuesta durante las etapas de inferencia y aprendizaje. De esta tabla podemos ver que el paso de data augmentation incluido en DACOLT mejora el aprendizaje del modelo de detección bajo condiciones de luminosidad alta. El recall y F1 promedio han mejorado respectivamente en $9{,}46\,\%$ y $7{,}17\,\%$ en comparación con el rendimiento considerando sólo el preprocesamiento en el momento de la inferencia, DaCoT.

Aplicando el preprocesamiento de luminosidad durante los pasos de inferencia y aprendizaje en los vídeos filmados bajo condiciones de luminosidad alta, se mejora el recall en $15{,}99\,\%$ y el F1 en $12{,}24\,\%$ en comparación con las condiciones de luminosidad alta originales.

Como estudio final, mostramos en la Tabla VII los resultados al aplicar la metodología de preprocesamiento DaCoLT en vídeos filmados bajo diferentes condiciones de luminosidad.

Tabla VII: El efecto de aplicar la metodología DaCoLT en vídeos filmados originalmente bajo diferentes condiciones de luminosidad.

Luminosidad	Tam arma	#Frames	#GT_P	#TP	#FP	Precisión	Recall	F1	orig. F1
Alta	Grande	121	112	84	0	100 %	75,00 %	85,71 %	82,10 %
	Mediano	107	90	64	0	100 %	71,11 %	83,12 %	65,67 %
	Pequeño	137	103	74	0	100 %	71,84 %	83,61 %	67,95 %
			Promedio			100 %	72,65 %	84,15 %	71,91 %
Media	Grande	109	98	84	0	100 %	85,71 %	92,31 %	92,89 %
	Mediano	116	98	78	0	100 %	79,59 %	88,64 %	85,38 %
	Pequeño	138	110	75	0	100 %	68,18 %	81,08 %	73,56 %
			Promedio			100 %	77,83 %	87,34 %	83,94 %
Baja	Grande	126	114	103	0	100 %	90,35 %	94,93 %	95,41 %
	Mediano	114	100	74	0	100 %	74,00 %	85,06 %	82,35 %
	Pequeño	138	101	72	0	100 %	71,29 %	83,24 %	84,57 %
			Promedio			100 %	**78,55 %**	**87,74 %**	87,44 %
Artificial	Grande	119	110	95	0	100 %	86,36 %	92,68 %	92,68 %
	Mediano	113	99	73	1	98,65 %	74,49 %	84,88 %	86,21 %
	Pequeño	96	90	63	1	98,44 %	70,79 %	82,36 %	83,87 %
			Promedio			99,03 %	77,21 %	86,64 %	87,59 %

Como se observa, DaCoLT mejora la detección especialmente en las peores condiciones (luminosidad más alta). En otras palabras, DaCoLT permite alcanzar precisiones similares en los vídeos independientemente de su nivel de luminosidad.

VI. Conclusiones y trabajo futuro

Este trabajo presenta un modelo de detección automática de armas blancas para videovigilancia basado en una nueva métodología de preprocesamiento guiado por luminosidad, denominado DaCoLT, que mejora la calidad de la detección. El modelo de detección obtenido muestra un alto potencial incluso en vídeos de baja calidad.

Nuestro sistema de detección de armas blancas puede ser utilizado en varias aplicaciones, por ejemplo, i) detección en tiempo real de armas blancas en videovigilancia y ii) control parental de vídeos o imágenes con contenido violento.

Como trabajo futuro, abordaremos la detección de armas en escenarios al aire libre, donde pueden estar presentes objetos en movimiento y las condiciones climáticas adversas pueden aumentar la dificultad de la detección.

Agradecimientos

Este trabajo contó con el apoyo del Ministerio de Ciencia y Tecnología de España bajo el proyecto TIN2017-89517-P. Siham Tabik contó con el apoyo del Programa Ramón y Cajal (RYC-2015-18136). La GPU Titan X Pascal utilizada para esta investigación fue donado por NVIDIA Corporation.

Referencias

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. *Operating Systems Design and Implementation*, 16:265–283, 2016.

[2] Himanshu Buckchash and Balasubramanian Raman. A robust object detector: Application to detection of visual knives. *IEEE Multimedia and Expo Workshops*, pages 633–638, July 2017.

[3] François Chollet. Keras: Theano-based deep learning library. *Code: github. com/fchollet. Documentation: http://keras.io*, 2015.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[5] Greg Flitton, Toby P Breckon, and Najla Megherbi. A comparison of 3d interest point descriptors with application to airport baggage object detection in complex ct imagery. *Pattern Recognition*, 46(9):2420–2436, 2013.

[6] Andrzej Glowacz, Marcin Kmieć, and Andrzej Dziech. Visual detection of knives in security applications using active appearance models. *Multimedia Tools and Applications*, 74(12):4253–4267, Jun 2015.

[7] Michał Grega, Andrzej Matiolański, Piotr Guzik, and Mikołaj Leszczuk. Automated detection of firearms and knives in a cctv image. *Sensors, dx.doi.org/10.3390/s16010047*, 16(1), 2016.

[8] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Zbigniew Fischer, Ianand Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Tensorflow object detection api. *Code: github.com/tensorflow/models/tree/master/object_detection*, CVPR 2017 (developing).

[9] Marcin Kmieć and Andrzej Glowacz. Object detection in security applications using dominant edge directions. *Pattern Recognition Letters*, 52:72 – 79, 2015.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[11] Roberto Olmos, Siham Tabik, and Francisco Herrera. Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, 275:66–72, 2018.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[13] Ali M. Reza. Realization of the contrast limited adaptive histogram equalization (clahe) for real-time image enhancement. *Journal of VLSI signal processing systems for signal, image and video technology*, 38(1):35–44, Aug 2004.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[15] Rohit Kumar Tiwari and Gyanendra K. Verma. A computer vision based framework for visual gun detection using harris interest point detector. *Procedia Computer Science*, 54:703–712, 2015.

[16] Ivan Uroukov and Robert Speller. A preliminary approach to intelligent x-ray imaging for baggage inspection at airports. *Signal Processing Research*, 4:1–11, January 2015.

[17] Zelong Xiao, Xuan Lu, Jiangjiang Yan, Li Wu, and Luyao Ren. Automatic detection of concealed pistols using passive millimeter wave imaging. In *2015 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–4. IEEE, 2015.