AJUSTE DE HIPERPARÁMETROS DE UNA RED NEURONAL CONVOLUCIONAL PARA EL RECONOCIMIENTO DE LENGUA DE SEÑAS

Jahaziel Anthony Hernández[1], César Javier Ortiz[1], Diego Antonio Legarda[1], Elizabeth Alzate[1]
[1]Tecnoacademia del Oriente Antioqueño

Resumen

El reconocimiento de imágenes es un área de creciente interés en gran medida impulsado por los recientes avances en Inteligencia Artificial. En este trabajo, se realiza un ajuste de hiperparámetros de una Red Neuronal Convolucional (CNN) para el reconocimiento de manos. Se propone una arquitectura preliminar y se diseña un experimento para evaluar el porcentaje de clasificación en el reconocimiento de mano derecha e izquierda usando diferentes tamaños de filtros en cada una de las capas de Convolución. Los resultados obtenidos han permitido ajustar la red para lograr una clasificación del 100% en pocas épocas de entrenamiento.

El presente trabajo hace parte de una investigación en desarrollo para el reconocimiento de la Lengua de Señas Colombiana (LSC) en donde se usará la arquitectura de Red Neuronal propuesta en esta fase para el reconocimiento de los símbolos usados para formar las letras del abecedario.

Palabras Claves: Reconocimiento de imagen, Red Neuronal Convolucional, hiperparámetros, lengua de señas.

Hyper-parameter Tuning of a Convolutional Neural Network for Sign Language Recognition

Abstract

Image recognition is an area of growing interest largely driven by recent advances in Artificial Intelligence. In this paper, a hyper-parameter tuning of a Convolutional Neural Network (CNN) for hand recognition is performed. A preliminary architecture is proposed and an experiment is designed to evaluate the classification rate in right and left hand recognition using different filter sizes in each of the Convolutional layers. The results obtained have allowed adjusting the network to achieve 100% classification in a few training epochs. The present work is part of a research in development for the recognition of the Colombian sign language where the Neural Network architecture proposed in this phase will be used for the recognition of the symbols used to form the letters of the alphabet.

Keywords: Image recognition, Convolutional Neural Network, hyper-parameters, sign language.

Introducción

Las personas con limitaciones auditivas requieren un lenguaje adaptado a sus necesidades. Para esto, se ha desarrollado la lengua de señas que tiene variaciones en cada país. Para el caso de Colombia, la lengua de señas fue reconocida oficialmente en el año 1996 mediante la ley 324 [1]. Gracias a sus particularidades lingüísticas y estructura gramatical, el Ministerio de Cultura, incluyó la lengua de señas colombiana al grupo de lenguas nativas del país, esto significa, que la lengua de señas hace parte del patrimonio inmaterial, cultural y lingüístico de Colombia, lo que garantiza su preservación y divulgación [2].

Según el censo realizado en Colombia en 2015, la comunidad sorda colombiana está compuesta por aproximadamente 455.718 personas [3]. En cuanto a la competencia lingüística, el 17% de los sordos no sabe leer, el 48% lee sólo avisos, el 18% lee, pero no entiende, el 14% lee y entiende y el 3% no responde. El 16% no sabe escribir, el 43% sólo escribe palabras, el 29% escribe frases, el 11% escribe párrafos y el 1% no responde [4].

El papel de la lengua de señas en la conformación de los sordos como grupo social es definitivo. Su carácter particular hace de los usuarios de la Lengua de Señas Colombiana personas diferentes a los hablantes de español como primera lengua. Para declararse una minoría los sordos deben validar su identidad a partir de una historia y una lengua común [4].

Actualmente, existen en Colombia varios proyectos que buscan por medio de la tecnología, tener

mayor inclusión con la población sorda, para esto se requiere un traductor que interprete el movimiento de las manos y lo convierta en palabras conocidas en el alfabeto. Uno de los mayores desarrollos lo tiene la aplicación *Centro de relevo* creada por el ministerio de las TIC y que según datos en su página oficial, en el primer trimestre del año 2019 el centro facilitó 51.754 comunicaciones desde la aplicación, la cual fue premiada en el año 2018 con el *Zero Project* como "mejor práctica innovadora que aprovechan las tecnologías de la información y garantizan el goce de derechos de las personas con discapacidad en todo el mundo" [5]. En esta aplicación el usuario realiza un registro y establece una conexión con un intérprete humano por un tiempo predeterminado de 30 minutos. Aplicaciones como esta podrían verse sustancialmente mejoradas si se desarrollan e implementan sistemas de reconocimiento automático basados en técnicas de visión artificial, de esta forma el tiempo de conexión y los horarios de disponibilidad se incrementan significativamente. No obstante, el reconocimiento de la lengua de señas conlleva retos importantes desde el punto de vista tecnológico, para poder identificar este lenguaje se necesita entender no solo la posición de las manos sino también los gestos en ciertos tipos de palabras para una correcta interpretación y para esto, se debe tener un reconocimiento del movimiento de las manos [3].

Las técnicas de visión artificial que se han ido desarrollando durante los últimos 50 años han dado como resultado actual un gran avance en los sistemas de reconocimiento, principalmente después del surgimiento de la Inteligencia Artificial (IA), que se ha convertido en el estado del arte en la mayoría de los casos. Aplicaciones como el reconocimiento de rostros, la clasificación de plantas, las etiquetas automáticas, reconocimiento de iris, reconocimiento de huellas, entre otras, son algunos ejemplos de la evolución de la IA [6].

Actualmente una de las técnicas de IA que mayor crecimiento y nivel de precisión tienen son las Redes Neuronales Convolucionales (CNN), inspiradas en la estructura biológica que procesa la información visual.

Dentro de los trabajos de orden nacional podemos encontrar diversas aplicaciones de las redes neuronales convolucionales en el procesamiento de imágenes como una herramienta para diversas aplicaciones. Siguiendo esta línea, el trabajo de Pallares et al. [7], en donde se emplean las redes neuronales para la prevención y detección de la enfermedad Sigatoka negra en plantaciones de banano, con el objetivo de reducir costos en el proceso de control de la enfermedad. En la propuesta de Torres-Galindo eta al [8], el cultivo para estudio es el de palma aceitera y se busca la detección de patologías de esta especie empleando imágenes multiespectrales. En el trabajo de Suat-Rojas [9], se utilizan las CNN para el reconocimiento del abecedario de señas para la lengua empleada en Colombia, en donde se realiza solamente un proceso de clasificación de imágenes que representan las letras del abecedario, estableciéndose como un punto de partida para el desarrollo del presente trabajo, en donde el principal reto es la identificación de los gestos a través del movimiento de las manos.

En las CNN es importante diferenciar entre los parámetros y los hiperparámetros. Los parámetros hacen referencia a los pesos de la Red Neuronal que son ajustados de forma automática por el algoritmo de entrenamiento, comúnmente el algoritmo de *backpropagation* (propagación hacia atrás) [10].

Por otra parte, los hiperparámetros se refieren a la configuración de la arquitectura a usar, por ejemplo: el número de capas, el número de neuronas por capa, entre otras. Para el caso de las CNN, un hiperparámetro importante es el tamaño de los filtros (*kernel*) en cada capa de convolución. A diferencia de los parámetros, estos valores de hiperparámetros no son ajustados de forma automática por el algoritmo, sino que son elegidos por el diseñador de la arquitectura.

Encontrar los hiperparámetros adecuados influye significativamente en la capacidad de la CNN en aprender patrones. Dada la importancia que tiene encontrar una arquitectura adecuada de CNN y el ajuste de los hiperparámetros, en esta primera etapa del proyecto se propone una arquitectura de CNN y se hace un estudio de la influencia que tiene el ajuste del tamaño de *kernel* en cada capa de convolución. El artículo está presentado de la siguiente forma: en la sección de metodología se presenta la base teórica de las CNN y con base a esto, se indica cuál ha sido el proceso realizado para seleccionar una arquitectura particular, así como la creación de la base de datos de imágenes para el entrenamiento y validación en la sección de resultados y discusión.

Metodología

La metodología del presente trabajo abarca la selección de la arquitectura de la CNN, la creación de la base de datos y el *hardware* usado. En las siguientes subsecciones se explica a detalle cada uno de los procesos metodológicos para los fines descritos.

Red Neuronal Convolucional

Una Red Neuronal Convolucional es una arquitectura inspirada en el córtex visual del ojo humano. Esta arquitectura les da a los computadores la capacidad de "ver" e identificar objetos en las imágenes que se le presentan [11]. Las convoluciones consisten en operaciones para obtener el producto escalar entre secciones de la imagen original y los filtros conocidos como *kernel*. Estas a su vez son procesadas usando funciones de activación y submuestreo para reducir la dimensionalidad de la imagen de entrada. La figura 1. es una representación para ejemplificar la extracción de características de una imagen a partir de operaciones convolucionales.

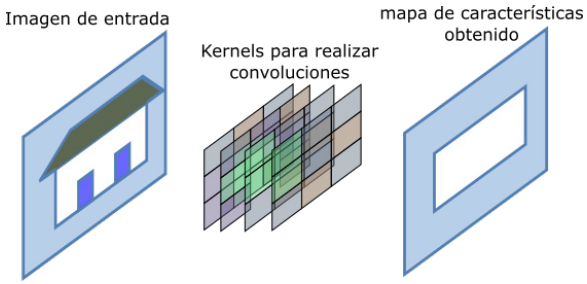

Figura 1: Proceso para establecer un mapa de características a partir de una imagen.

La figura 2 muestra la operación de convolución entre una capa de partida y un filtro determinado, en esta se muestra como las operaciones básicas de multiplicación y suma van generando la capa convolucionada.

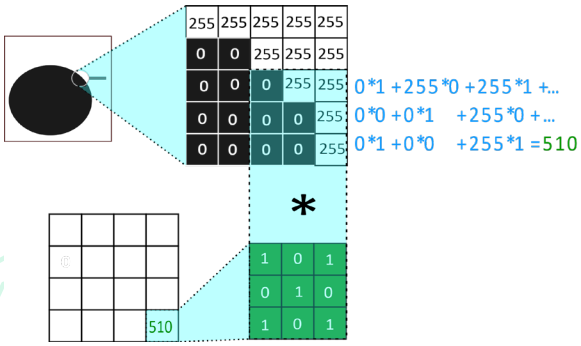

Figura 2: Proceso de convolución en una imagen de entrada y un filtro (3,3).

Este proceso se puede repetir conectando la salida de la primera capa convolucional con una nueva. Cada capa tiene la capacidad de detectar patrones cada vez más complejos. El entrenamiento de una CNN consiste en hacer pasar las imágenes de entrada cada una con una etiqueta que le indique a la red que clase es la imagen, se calcula el error en la salida y este error usando un algoritmo de propagación hacia atrás ajusta los valores de los *kernel*. Una vez la CNN está entrenada, puede reconocer imágenes nuevas que no entraron en la etapa de entrenamiento, la precisión en el reconocimiento depende si el entrenamiento fue bueno y la CNN logró generalizar los patrones relevantes del objeto a clasificar.

Arquitectura de la Red Neuronal

Acorde a la literatura consultada, se elige una arquitectura inicial con cuatro capas convolucionales, 512 filtros de tamaño (3,3) en la primera capa, 512 filtros de tamaño (3,3) en la segunda capa, 256 filtros de tamaño (1,1) en la tercera capa y 256 filtros de tamaño (2,2) en la cuarta capa. En cada capa convolucional hay una función de activación ReLU y una función *MaxPooling2* tal como se muestra en la figura 3.

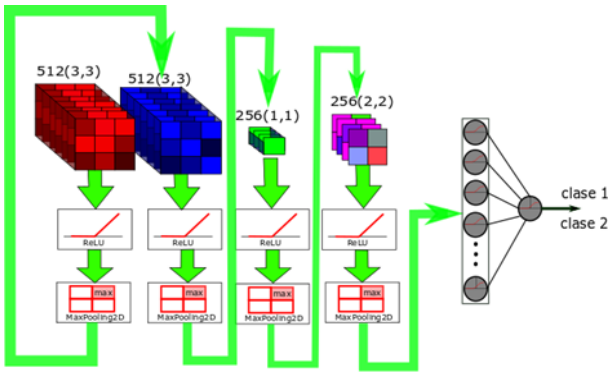

Figura 3: Arquitectura de la Red Neuronal Convolucional propuesta.

La metodología usada para realizar el ajuste de los hiperparámetros seleccionados que pueden influir en mayor medida en el entrenamiento de la CNN consistió en probar diferentes tamaños de filtro (*kernel*) en cada capa convolucional, empezando con valores iniciales de (1,1) lo que lleva a obtener resultados de tipo cuantitativo correlacional entre los porcentajes de clasificación para cada tamaño de filtro estudiado.

Creación de la base de datos

Para la creación de la base de datos con la cual se

entrenó la red, se usó una cámara digital convencional, se tomaron 200 fotos de la mano derecha, 400 fotos para los datos de entrenamiento, 200 por cada clase (mano derecha, mano izquierda). Otras 100 fotos son usadas como datos de validación una vez la CNN está entrenada. Las fotos son tomadas en diferentes condiciones de luminosidad y diferentes distancias desde la cámara a la mano con el fin de buscar una mejor generalización en el aprendizaje. En la figura 4 se muestra un ejemplo de las imágenes usadas para el entrenamiento.

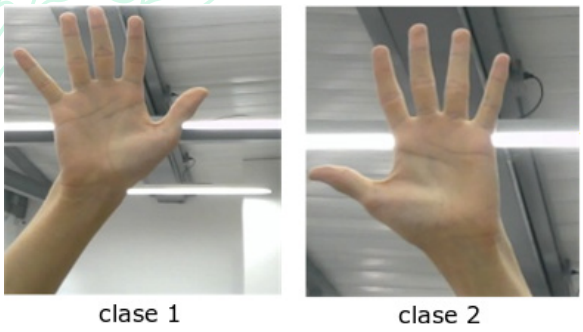

Figura 4: Ejemplo de imágenes usadas para el entrenamiento y validación de la CNN.

Hardware y *Software* usado

Para la implementación de la CNN se usó la librería de *TensorFlow* y *Keras*. El entrenamiento de la red fue realizado en un computador con tarjeta gráfica NVIDIA 1060 con 16 GB de memoria RAM y 6 GB de memoria VRAM suministrado por la Tecnoacademia del Oriente Antioqueño.

Resultados y Discusión

Se entrenó la CNN durante 30 épocas para cada valor de *kernel* elegido. Se analizaron los datos de entrenamiento y validación para los diferentes tamaños de *kernel* propuestos. Las siguientes gráficas (Fig. 5-14) muestran la evolución en los porcentajes de clasificación, para los datos de entrenamiento y validación de la CNN propuesta.

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(1,1)	(1,1)	(1,1)	(1,1)

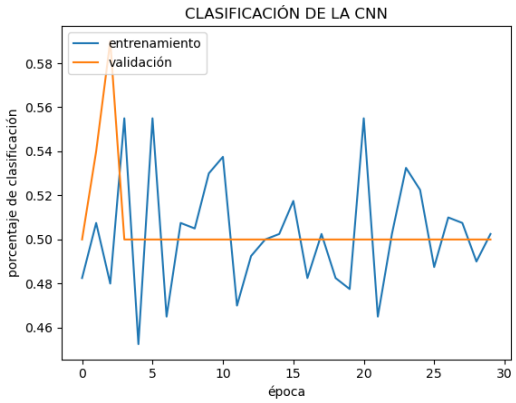

Figura 5: Entrenamiento y validación de la CNN para (1,1)- (1,1)-(1,1)-(1,1).

Se observó que cuando el tamaño del *kernel* de la primera capa es (1,1), la CNN no logra encontrar los patrones que le permitan diferenciar entre las clases.

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(3,3)	(1,1)	(1,1)	(1,1)

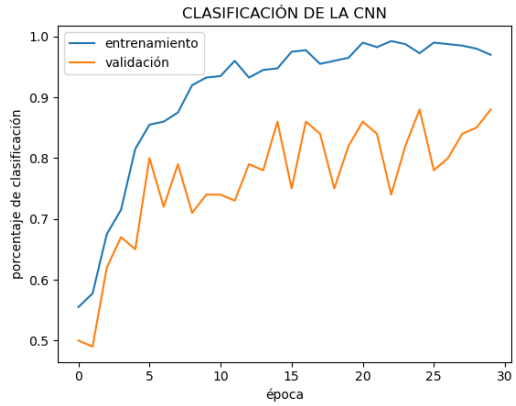

Figura 6: Entrenamiento y validación de la CNN para (3,3)-(1,1)-(1,1)-(1,1).

Con el *kernel* de la primera capa de tamaño (3,3), la CNN logra aprender los datos de entrenamiento en valores cercanos al 100%, mientras que los datos de validación oscilan entre 70% y 90%.

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(5,5)	(1,1)	(1,1)	(1,1)

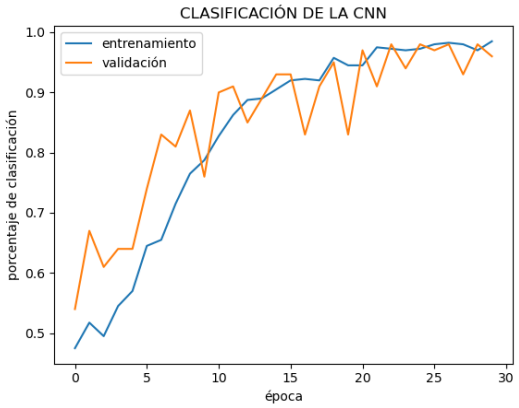

Figura 7: Entrenamiento y validación de la CNN para (5,5)-(1,1)-(1,1)-(1,1).

Con el *kernel* de la primera capa de tamaño (5,5), la CNN logra aprender los datos de entrenamiento en valores cercanos al 100%, mientras que los datos de validación oscilan entre 90% y 99% después de 20 épocas de entrenamiento. Este resultado sugiere que el tamaño del *kernel* en la primera capa de convolución mejora el aprendizaje aumentando su tamaño. Para validar esta hipótesis se elige un *kernel* mayor en la siguiente prueba.

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(9,9)	(1,1)	(1,1)	(1,1)

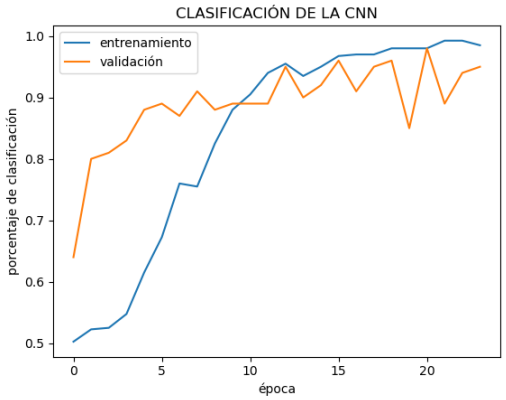

Figura 8: Entrenamiento y validación de la CNN para (9,9)-(1,1)-(1,1)-(1,1).

En este caso la CNN logró aprender los datos de entrenamiento y generalizó también los datos de validación; sin embargo, se nota mayor oscilación. Esto sugiere que el tamaño del kernel en la primera capa debe estar en un rango de tamaño entre (3,3) y (9,9). Se toma entonces un valor fijo de (7,7) para la primera capa y se realizan variaciones para las otras capas.

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(7,7)	(3,3)	(1,1)	(1,1)

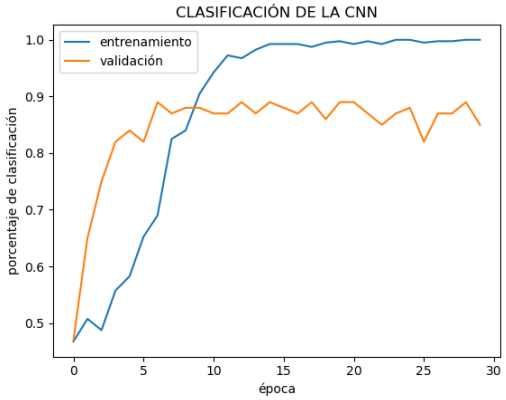

Figura 9: Entrenamiento y validación de la CNN para (7,7)-(3,3)-(1,1)-(1,1).

En este caso la CNN logró aprender los datos de entrenamiento, no obstante, los datos de validación llegaron hasta un valor máximo de 90%.

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(7,7)	(5,5)	(1,1)	(1,1)

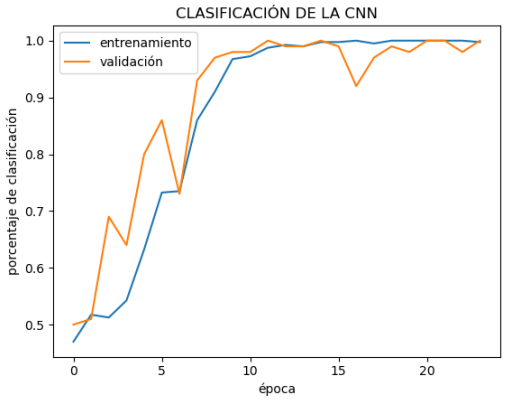

Figura 10: Entrenamiento y validación de la CNN para (7,7)-(5,5)-(1,1)-(1,1).

Aumentando el tamaño del *kernel* de (3,3) a (5,5) en la segunda capa, se evidencia un incremento en el porcentaje de clasificación aumentando de 90% a 99% en 10 épocas de entrenamiento.

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(7,7)	(7,7)	(1,1)	(1,1)

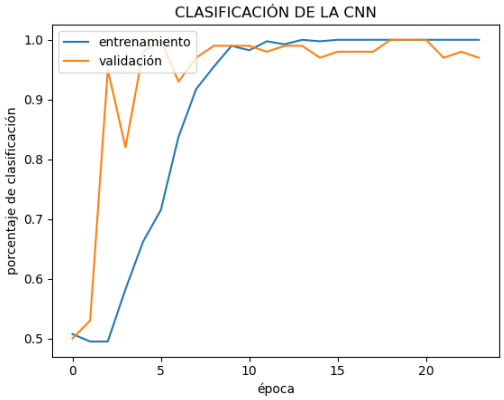

Figura 11: Entrenamiento y validación de la CNN para (7,7)-(7,7)-(1,1)-(1,1).

Del ajuste del tamaño de *kernel* el resultado obtenido usando dos capas convolucionales con tamaño (7,7) dio un resultado en donde con pocas épocas de entrenamiento logra aprender bien los datos de entrenamiento y los datos de validación.

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(7,7)	(7,7)	(3,3)	(1,1)

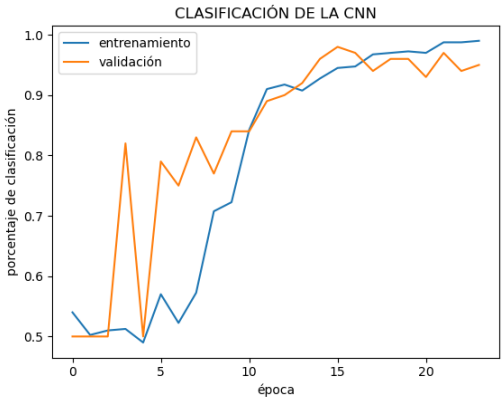

Figura 12: Entrenamiento y validación de la CNN para (7,7)-(7,7)-(3,3)-(1,1).

Al aumentar el tamaño del *kernel* en la tercera capa de convolución se notó una disminución en la velocidad de aprendizaje en comparación a otros resultados logrados con el *kernel* de esta capa en (1,1).

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(7,7)	(7,7)	(3,3)	(3,3)

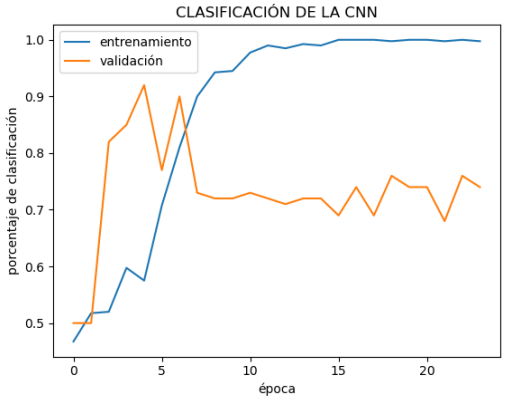

Figura 13: Entrenamiento y validación de la CNN para (7,7)-(7,7)-(3,3)-(3,3).

Después de probar otras combinaciones, se encontró un ajuste del tamaño de *kernel* dado por los valores:

kernel 1	*kernel 2*	*kernel 3*	*kernel 4*
(7,7)	(3,3)	(1,1)	(2,2)

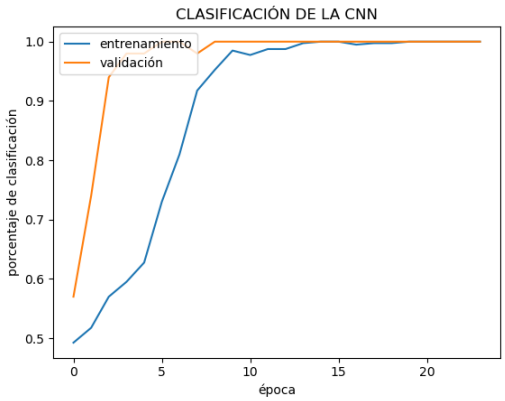

Figura 14: Entrenamiento y validación de la CNN para (7,7) - (3,3)-(1,1)-(2,2).

En este caso, la CNN aprendió rápidamente los datos de entrenamiento y generalizó los datos de validación logrando la clasificación de 100% en pocas épocas.
De los resultados obtenidos en esta etapa de la investigación, se encuentra la importancia de realizar una búsqueda no solamente de una arquitectura de Red Neuronal, sino también de realizar el ajuste de los hiperparámetros de dicha arquitectura, con el fin de optimizar la búsqueda de patrones que le permitan a la Red una mejor generalización para los datos futuros.

En el caso particular del ajuste del tamaño de los *kernel* en las capas convolucionales, se evidencia

que son un hiperparámetro relevante para lograr el reconocimiento de diferentes objetos. Para el reconocimiento de la lengua de señas es importante, primero garantizar que la arquitectura de red neuronal seleccionada tenga la capacidad de reconocer dos clases como en este caso, en donde la CNN logra identificar claramente entre la mano derecha y la mano izquierda, para luego lograr una clasificación multiclase para las diferentes posiciones que dan forma a las letras del alfabeto en la lengua de señas.

Del estudio realizado en la influencia del tamaño de los *kernel* en las capas convolucionales, se muestra experimentalmente que el tamaño de las primeras capas debe ser mayor y a medida que se va profundizando en las capas, el tamaño de *kernel* debe ir disminuyendo proporcionalmente.

En esta etapa del proyecto se puede decir que se tiene definida una arquitectura de CNN con los hiperparámetros bien ajustados para encontrar patrones con una alta tasa de aprendizaje y de generalización para las manos humanas. Sin embargo, se debe tener en cuenta que en este trabajo solo se estudió la influencia del tamaño de *kernel,* dejando estáticos otros hiperparámetros también importantes, tales como la elección del número de capas ocultas, el número de filtros en cada capa, el tamaño del *pool-size,* entre otras.

Para continuar con el proyecto de investigación, actualmente se están recopilando las imágenes para crear una base de datos con todos los símbolos usados en la Lengua de Señas Colombiana. Estas imágenes serán procesadas en la CNN propuesta en esta primera etapa y se verificará la capacidad de la arquitectura de lograr clasificar más de dos clases.

Conclusiones

En este trabajo, previo al reconocimiento de lenguaje de señas usando técnicas de aprendizaje automático, se realizó un estudio sistemático de los hiperparámetros que influyen en el porcentaje de clasificación, principalmente centrados en el tamaño de los filtros de convolución en una arquitectura CNN. La arquitectura propuesta y el ajuste de dichos hiperparámetros han permitido un aprendizaje en pocas épocas de entrenamiento, logrando 100% de clasificación, tanto en los datos de entrenamiento como en los datos de validación.
Del presente estudio, se puede inferir que el tamaño de los *kernel* en las primeras dos capas convolucionales son un hiperparámetro determinante en la capacidad de aprendizaje de una CNN. Se puede observar también que los tamaños de *kernel* que presentan mejores resultados van disminuyendo de tamaño en capas posteriores.

Como trabajo futuro se espera clasificar imágenes de todas las letras del abecedario propias de la Lengua de Señas Colombiana.

Agradecimientos

Los autores agradecen a la Tecnoacademia del Oriente Antioqueño por facilitar los equipos de cómputo para la adquisición de datos e implementación de los algoritmos propuestos.

Referencias:

L. Hurtado (2016) “Inclusión educativa de las personas con discapacidad en Colombia,” 2016.

“La lengua de señas colombiana hace parte del patrimonio inmaterial, cultural y
lingüístico del país.” www.insor.gov.co/home/la-lengua-de-senas-colombiana-hace-parte-del-patrimonio-inmaterial-cultural-y-linguistico-del-pais. Accessed: 2021-08-15.

D. J. Botina-Monsalve, M. A. Domínguez-Vásquez, C. A. Madrigal-González, and A. E. Castro-Ospina, (2018) “Clasificación automática de las vocales en el lenguaje de señas colombiano,” Tecnológicas .

Hurtado Tarazona, A. (2003). Entre la integración y la diferenciación-la lucha por la reivindicación de los sordos como comunidad lingüística en Colombia (Bachelor's thesis, Uniandes).

“Centro de relevo: app más valiosa del mundo para las personas sordas.” https://mintic.gov.co/portal/inicio/Sala-de-Prensa/MinTIC-en-los-Medios/100388:Centro-de-Relevo-app-mas-valiosa-del-mundo-para-las-personas-sordas
Accesed: 2021-08-16.

Trillas, E. (1998). La inteligencia artificial: máquinas y personas. Temas de Debate, SA.

Pallares, C. J., Lallemand, K. S., & Visbal, F. D. (2021). Control preventivo de sigatoka negra en cultivo banano apoyado en redes convolucionales.

Torres-Galindo, A., Camacho-Tamayo, J., Torres-León, J., & Cruz-Roa, A. Análisis preliminar de detección de patologías en cultivos de palma aceitera usando Redes Neuronales Convolucionales.

Suat-Rojas, N., Pinzón, E., Rodríguez, O., & Montoya, B. Reconocimiento del abecedario de la lengua de señas colombiana por medio de Imágenes y Redes Neuronales Convolucionales.

Valencia Reyes, M. A. (2007). Algoritmo Backpropagation para Redes Neuronales: conceptos y aplicaciones. Instituto Politécnico Nacional. Centro de Investigación en Computación.

Fu, H., Niu, Z., Zhang, C., Ma, J., & Chen, J. (2016). Visual cortex inspired CNN model for feature construction in text analysis. Frontiers in computational neuroscience, 10, 64.