ISSN 1870-4069

Aprendizaje de similitud semántica para el reconocimiento del alfabeto de lengua de señas

Atoany Nazareth Fierro Radilla[1], Regina Alexia Blas Flores[1], Emiliano Vivas Rodriguez[1], Karina Ruby Perez Daniel[2], Gibran Benitez-Garcia[3]

[1] Tecnológico de Monterrey, Campus Cuernavaca, México

[2] Universidad Panamericana, Facultad de Ingeniería, México

[3] The University of Electro-Communications, Japón

afierror@tec.mx, karinaperez@up.edu.mx, gibran@ieee.org

Resumen. La lengua de señas es un importante método de comunicación que utilizan las personas que padecen alguna enfermedad auditiva, especialmente personas con problemas del habla y/o del escucha. En los Estados Unidos, aproximadamente dos millones de personas que viven con discapacidad auditiva utilizan ASL (por sus siglas en ingles American Sign Language). Por lo tanto, el objetivo de este estudio es investigar y desarrollar un sistema de reconocimiento para el alfabeto ASL, haciendo uso de redes neuronales convolucionales (VGG16 y Mobilenet) siamesas (dos arquitecturas iguales) para poder mejorar la comunicación y las relaciones interpersonales con las personas que viven con discapacidades auditivas. La propuesta se basa en implementar el aprendizaje de similitud semántica para reducir la variación intraclase y la similitud interclase de imágenes del alfabeto de la lengua de señas en un espacio euclidiano. Los resultados muestran que el sistema propuesto tiene una precisión promedio de 99 % y 90.1 % en el conjunto de datos MINIST y ASL, respectivamente. Utilizando la técnica de análisis estadística llamada t-SNE se puede demostrar por qué nuestra propuesta supera los algoritmos reportados en la literatura.

Palabras clave: ASL, lengua de señas, aprendizaje de similitudes, CNN, red siamesa.

Semantic Similarity Learning for American Sign Language Alphabet Recognition

Abstract. Sign language is an important manner to convey information among deaf community, and it is primarily used by people who have hearing or speech impairments. In USA, approximately two million of deaf people use ASL (American Sign Language). Therefore, the purpose of this study is to

investigate and develop a system for ASL alphabet recognition using two Convolutional Neural Networks (CNN), such as VGG16 and Mobilenet. The proposal is to implement semantic similarity learning in order to reduce the high intra-class variation and the high inter-class similarity in an euclidean space of sign images. The results show that the proposed system improves the ASL alphabet recognition in a considerable way, yielding an average accuracy of 99% and 90.1% on MNIST Dataset and ASL Dataset, respectively. Using an statistical analysis technique, called t-SNE we can demonstrate why our proposal outperform the methods reported in literature.

Keywords: ASL, sign language, semantic similarity, CNN, siamese network.

1. Introducción

La forma en la que nos conectamos físicamente con el mundo es a través de las manos; realizamos la mayoría de las tareas diarias con ellas. Por otro lado, utilizamos dispositivos periféricos como mouse y el joystick para trabajar con una computadora (Interacción humano-máquina) [24], dicha interacción es un área de investigación multidisciplinaria con diversas aplicaciones en control, robótica, estudio de comportamiento psicológico, realidad virtual, reconocimiento de lengua de señas, visualización científica y, en el Metaverso [24, 2].

Los sistemas de Reconocimiento de Lengua de Señas (RLS) se realizan por medio de la Interacción Humano-Computadora, convirtiendo los gestos y movimientos de manos en comandos de texto y/o de voz, permitiendo la comunicación entre los seres humanos a través de una computadora, entre las personas que presentan una discapacidad auditiva y las personas oyentes [2, 26].

En los Estados Unidos (EE.UU) hay aproximadamente dos millones de personas sordas, algunos de ellos nacen con pérdida auditiva en ambos oídos, mientras que otros pierden la audición debido a factores como la rubéola y la meningitis [24].

La lengua de señas americana (ASL) es el segundo idioma distinto del inglés más utilizado en los EE. UU. después del español, tiene 36 formas de manos, 26 letras y alrededor de 6000 palabras, que consisten en movimientos corporales complejos. Las señas se crean usando la mano derecha, la mano izquierda, ambas manos y expresiones faciales y/o corporales [24, 23].

A pesar de que ASL es el principal modo de comunicación para la mayoría de las personas sordas en EE. UU., siguen existiendo problemas de comunicación con las personas oyentes ya que no comprenden el lenguaje ASL. Si el ASL se pudiera traducir automáticamente en texto o voz en inglés y/o español, será mucho más fácil para las personas sordas sentirse incluidos y tener mayor comunicación [24]. Los prinpales aportes de este trabajo son:

- Desarrollo de un algoritmo para el reconocimiento del alfabeto ASL.
- Comparación de los resultados de clasificación entre arquitecturas simples y siamesas.
- Comprobación de nuestra hipótesis utilizando la técnica t-SNE.

2. Sistema de reconocimiento del alfabeto de lengua de señas

Desde hace más de dos décadas, se han publicado diversos trabajos de investigación para el reconocimiento de la lengua de señas con grandes avances, en especial para los estadounidenses [26], australianos [12], los coreanos [17] y el chino [14], sin embargo, existen grandes dificultades debido a la complejidad de los movimientos de manos y cuerpo en expresiones en lengua de señas (LS) [24].

Los enfoques utilizados para resolver los problemas de SLR se pueden clasificar en dos métodos principales, sensores y visión por computadora [2, 26]. En los enfoques basados en sensores, se suele usar un guante o sensor especial para rastrear la orientación, la posición, la rotación y los movimientos de la mano [24, 26, 3], los cuales proporcionan información precisa de la posición de la mano [24, 23].

Sin embargo, son demasiado pesados e incómodos para el uso diario [26]. Por otro lado, los métodos basados en visión por computadora consisten en técnicas de procesamiento de imágenes y aprendizaje automático para capturar y clasificar el movimiento del cuerpo y la forma de la mano usando imágenes a color sin necesidad de sensores conectados al ser humano [24, 3]. Este documento se centra únicamente en el enfoque basado en la visión.

3. Deletreo en la lengua de señas

El deletreo con los dedos es la representación de cada letra del alfabeto mediante una seña. Los usuarios de ASL usan el alfabeto en inglés deletreado con los dedos y manos (AFA, American Finger-spelled Alphabet), mientras que usuarios de otro alfabeto, como por ejemplo el mexicano o el chino, utilizan diferentes variaciones de deletreo. El AFA consta de 22 configuraciones de mano que cuando se mantienen en ciertas posiciones y/o se producen ciertos movimientos, se representan las 26 letras del alfabeto inglés [24, 23].

Las investigaciones coinciden en que el deletreo en lengua de señas está integrado en ASL de manera muy sistemática [9]. Uno de los principales usos del letreo es principalmente para representar nombres propios, nombre de medicamentos o palabras en inglés sin equivalentes en lengua de señas [5].

Además, el deletreo es una parte importante de la lengua de señas para los nuevos usuarios y ayuda a las personas a abreviar señas más largas, a comunicar dos palabras compuestas [4] y a cerrar la brecha entre el léxico ASL a través de la geografía y las culturas [3]. La incorporación del deletreo en ASL resulta más conveniente en escenarios críticos, como es en el ámbito de la medicina.

4. Retos en el reconocimiento del alfabeto ASL

La clasificación de los signos depende de las configuraciones de las manos, las cuales son captadas por una cámara de color y/o profundidad. Las complejidades de las señas hacen que el reconocimiento del alfabeto ASL sea una tarea difícil debido a dos factores principales, la similitud entre clases y las variaciones dentro de las clases.

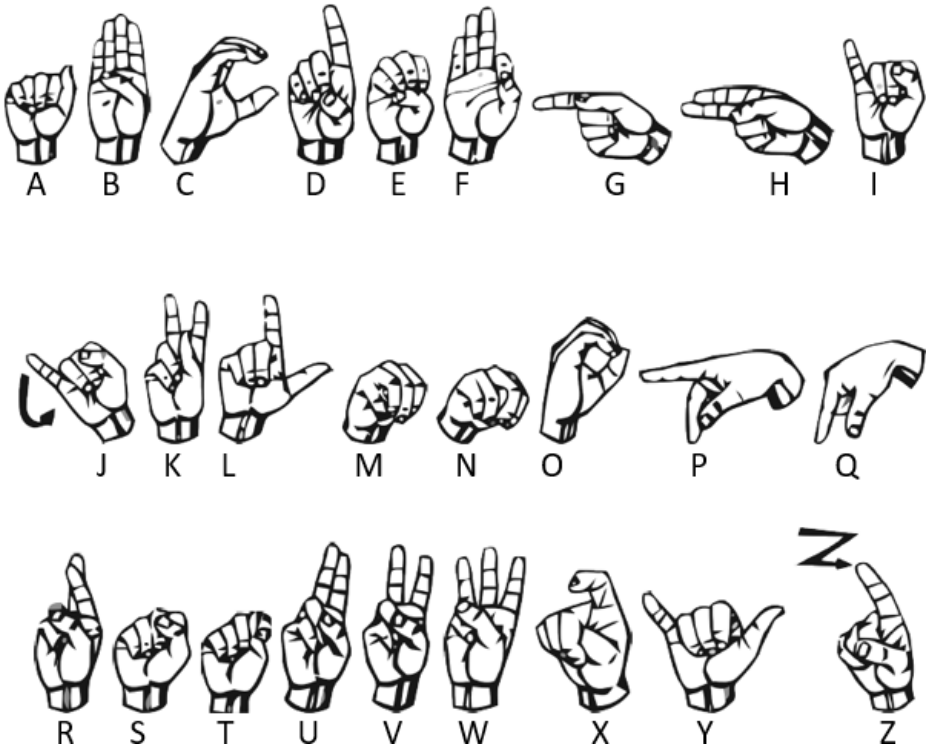

Fig. 1. Alfabeto de la langua de señas americana. Las letras J y Z involucran movimiento.

La similitud entre clases significa que algunas letras están muy estrechamente relacionadas con otras y difieren muy poco en la ubicación de los dedos. Por ejemplo, las letras M y N solo se diferencian entre sí, si el pulgar está entre el primer y el segundo o entre el segundo y el tercer dedo.

Las grandes variaciones intraclase significan que, dentro de una clase, existen diferencias entre las muestras, como la iluminación, el color de la piel, las variaciones del fondo y la posición relativa del usuario con respecto a la cámara.

5. Trabajos relacionados

El reconocimiento del alfabeto ASL se basa principalmente en dos subtareas: extracción de características y clasificación multiclase [26]. En la primer subtarea se realiza la detección y extracción de características locales.

Por otro lado, en la segunda subtarea, éstas características extraídas son comprendidas y caracterizadas para clasificar las muestras [22]. Para el reconocimiento del alfabeto ASL, en la literatura se pueden encontrar dos enfoques que se han utilizado: métodos tradicionales y basados en CNN.

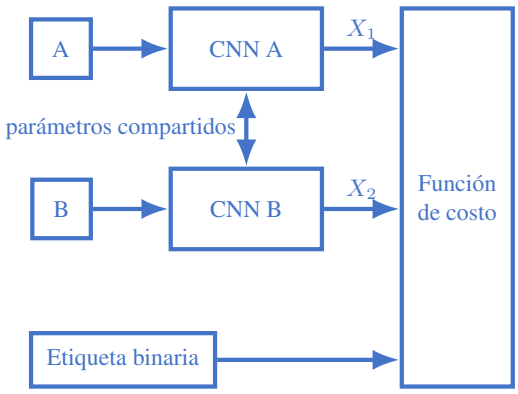

Fig. 2. La arquitectura siamesa está compuesta por dos redes idénticas (Red A y Red B). Estas redes se alimentan con las imágenes A y B. La capa de función de costo se alimenta con los descriptores visuales generados por dichas redes y una etiqueta binaria; esta etiqueta binaria indica si el par es positivo o negativo.

5.1. Métodos tradicionales

En inglés se les conoce como Handcrafted Methods ya que el algoritmo de extracción de características ha sido manualmente construido [16]. Para los enfoques tradicionales es necesario diseñar el mejor algoritmo de extracción de características que mejor se adapte; además, una vez que las características se seleccionan, se debe elegir un clasificador de tal manera que se ajuste con la etapa de extracción de características.

Uno de los primeros sistemas de reconocimiento del alfabeto ASL utiliza los filtros de Gabor para la selección de características y random forest para la clasificación [22], donde los autores obtuvieron un 49 % de precisión.

En [27] los autores popusieron utilizar SP-EMD (Super Pixel Earth Movers Distance, en inglés), el cual es un algoritmo que mide la similitud de las características de forma, textura y profundidad para las imágenes de señas, obteniendo una precisión del 75.8 %. Por otro lado, en lugar de extraer similitudes, los autores en [20] extrajeron textura utilizando espaciogramas volumétricos a partir de los patrones binarios locales (VS-LBP) y utilizando una máquina de soporte vectorial (SVM) como clasificador, obtuvieron una precisión del 83.7 %.

Algunos otros autores [6, 21, 18] consideraron trabajar con la información de profundidad en lugar del color y como clasificador propusieron utilizar bosques aleatorios, obteniendo una precisión de clasificación del 81.1 %, 87 % y 90 %, respectivamente. Los trabajos relacionados mencionados arriba básicamente consisten de dos tareas separadas, extracción de características y clasificación.

Esto produce lo que nosotros llamamos ”fenómeno de desacoplamiento”, donde cierta información se pierde en la etapa de la extracción de características y no logra propagarse hacia la tarea de clasificación.

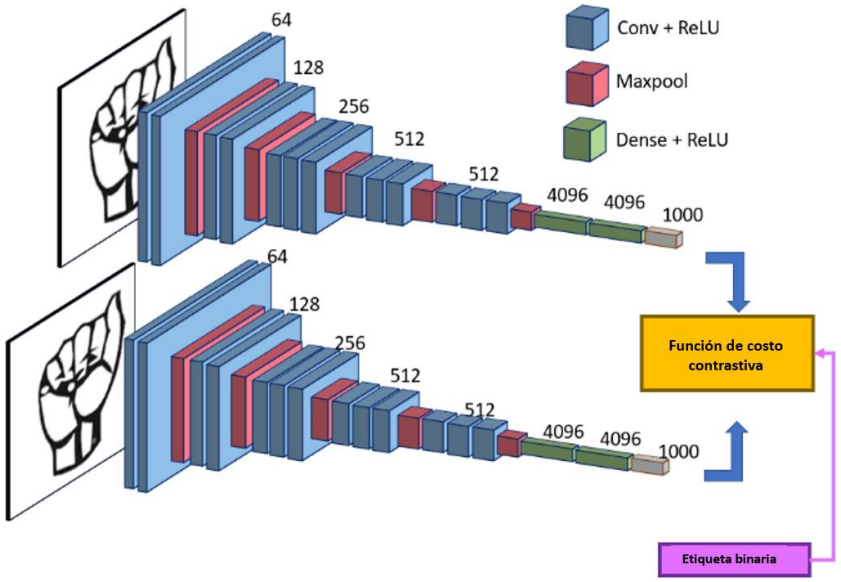

Fig. 3. Arquitectura VGG siamesa. La etiqueta binaria indica la similitud de la imagen. La función de costo contrastiva permite generar descriptores visuales de acuerdo a la similitud entre las imágenes.

5.2. Métodos basados en redes neuronales convolucionales

En el año 2012, las redes neuronales convolucionales (CNN por sus siglas en Inglés) se volvieron muy populares entre la comunidad de visión por computadora debido al éxito de la red Alexnet en la competencia ILSVRC (ImageNet Large Scale Visual Recognition Challenge).

Una de las ventajas de utilizar redes CNN con respecto a los métodos tradicionales es que la extracción de características y la clasificación se realiza en un solo algoritmo sin la intervención humana. En otras palabras, la red aprende qué características son las mejores para extraerse y tener un mejor resultado en la clasificación.

Una red CNN está compuesta por capas convolucionales y capas densas; las primeras permiten obtener representaciones no lineales de imágenes para la extracción de características, mientras que las capas densas son las responsables de la clasificación. Uno de los métodos basados en CNN encontrado en la literatura es [1], donde se propone utilizar una red CNN de dos entradas, una para las imágenes de color y la otra para imágenes de profundidad.

Las convoluciones de estas dos etradas se concatenan y se introducen a las capas densas las cuales obtienen una precisión de clasificación del 80.3 %. Por otro lado, los autores en [26] proponen una estrategia novedosa la cual utiliza solamente imágenes de profundidad para generar una nube de puntos en 3 dimensiones.

En [3] las información de profundidad es también utilizada; los autores utilizaron un sensor de Microsoft Kinnect como extractor de características las cuales fueron clasificadas utilizando PCANet, obteniendo una precisión de 84.5 %.

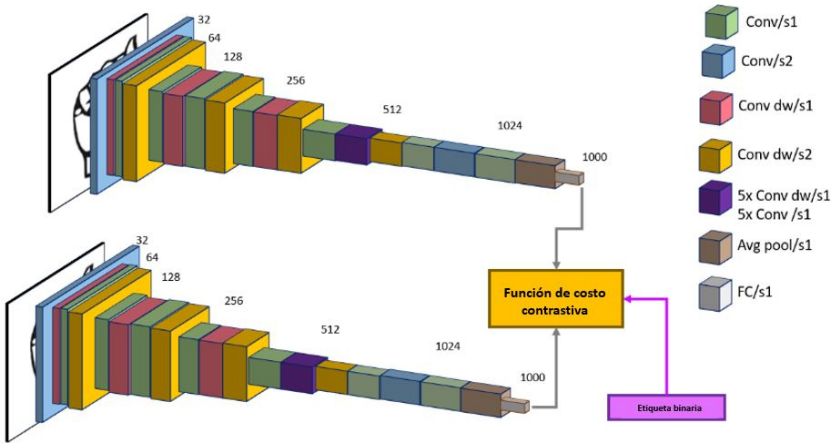

Fig. 4. Arquitectura Mobilenet siamesa. La etiqueta binaria indica la similitud de la imagen. La función de costo contrastiva permite generar descriptores visuales de acuerdo a la similitud entre las imágenes. La última capa coloreada en gris se utiliza como descriptor visual.

A diferencia de los métodos arriba mencionados, en este trabajo utilizamos una arquitectura doble la cual nos entrega dos descriptores cuya distancia entre ellos depende de la similitud de las imágenes. Esta arquitectura doble, mejor conocida como red siamesa se explica a continuación.

6. Arquitectura CNN siamesa

Las redes CNN necesitan entrenarse con una gran cantidad de información, sin embargo, es muy dificil en algunas aplicaciones conseguir tal cantidad de información. En una red neuronal siamesa, el set de entrenamiento se genera mediante combinaciones de pares de imágenes; de esta manera, no se necesita una gran cantidad de muestras. Estos pares de imágenes pueden ser positivos (imágenes de la misma clase) o negativos (imágenes de diferente clases).

Esto hace que las redes neuronales siamesas sean más robustas frente conjuntos de datos sin balancear. Una red siamesa consiste en dos redes neuronales convolucionales idénticas que comparten sus parámetros y se utilizan para aprender similitudes semánticas. La hipótesis de este artículo es que, usando una arquitectura siamesa, se puede reducir la alta similitud interclase y las altas variaciones intraclase, mejorando así el reconocimiento del alfabeto de lengua de señas.

7. Sistema propuesto

Los experimentos se realizaron utilizando dos arquitecturas CNN diferentes, VGG16 y Mobilenet, así como sus versiones siamesas, en dos conjuntos de datos diferentes, Alphabet Dataset [15] y Sign Language MNIST [25]. El entrenamiento se hizo utilizando Keras y Tensorflow en una máquina Intel Core i7 con una GPU NVIDIA GeForce RTX 2070 SUPER.

Tabla 1. Reporte de clasificación de la arquitectura VGG16.

	VGG16		VGG16 Siamesa	
Métrica	**MNIST**	**ASL Alphabet**	**MNIST**	**ASL Alphabet**
Exactitud	0.58	0.30	0.99	0.89
Precisión	0.71	0.47	0.99	0.88
Exhaustividad	0.58	0.30	0.99	0.89
Puntaje F1	0.56	0.28	0.99	0.89

7.1. Datasets

Para los experimentos, se utilizaron los conjuntos de datos del lenguaje de señas MNIST y ASL Alphabet. El conjunto de datos MNIST de lenguaje de señas está compuesto por 34,627 imágenes de tamaño 28x28 píxeles divididas en 24 clases (de la A a la Z, no contienen muestras para J y Z debido involucran movimientos de gestos). Por otro lado, el conjunto de datos de ASL Alphabet está compuesto por 87,000 imágenes de 200x200x3 divididas en 29 clases (de la A a la Z) y 3 clases más etiquetadas como ”SPACE”, ”DEL” y ”NOTHING”; en este trabajo, “J” y “Z” se consideran signos estáticos.

7.2. VGG16

VGG16 se propuso por primera vez en [25] y logró una precisión de prueba del 92,7 % en ImageNet [7], que es un conjunto de datos compuesto por más de 14 millones de imágenes de 1000 clases diferentes. Este modelo participó en el reto ILSVRC-2014, mejorando el rendimiento de Alexnet al reducir el tamaño de los kernel a 3×3. La función de pool utilizada por la red VGG16 es una capa de de 2×2 con un paso de 2.

Esta arquitectura tiene 3 capas densas, seguidas de una capa softmax como salida. La principal contribución de VGG16 es que, en lugar de tener una gran cantidad de hiperparámetros, los autores se enfocaron en tener capas convolucionales de 3×3 con un stride de 1 ($s = 1$). En el entrenamiento de ambos conjuntos de datos, las imágenes se redimensionaron a $224 \times 224 \times 3$ debido a los requisitos de entrada de la red.

La red se entrenó primero en el conjunto de datos del lenguaje de señas del MNIST durante 100 épocas. Se utilizaron 24,720 imágenes como conjunto de entrenamiento y 2,735 para el conjunto de validación, logrando una precisión de entrenamiento de 0.9231, pérdida de entrenamiento de 5.3653, precisión de validación de 0.9872 y pérdida de validación de 0.3970.

Por otro lado, para el entrenamiento con el conjunto de datos ASL Alphabet, el número de épocas fue de 50 debido a la capacidad de la memoria de la máquina. En este experimento se utilizaron 78,300 imágenes de $224 \times 224 \times 3$ como conjunto de entrenamiento y 8,700 imágenes de $224 \times 224 \times 3$ como conjunto de validación. Los resultados utilizando el conjunto de entrenamiento fueron: exactitud de 0.9285, y error de 5.5736. Para el conjunto de validación se obtuvo una exactitud de 0.8477 y un error de 16.2988.

Tabla 2. Reporte de clasificación de la arquitectura Mobilenet.

	Mobilenet		**Mobilenet Siamesa**	
Métrica	**MNIST**	**ASL Alphabet**	**MNIST**	**ASL Alphabet**
Exactitud	0.08	0.04	1.00	0.91
Precisión	0.08	0.07	1.00	0.91
Exhaustividad	0.08	0.04	1.00	0.91
Puntaje F1	0.04	0.01	1.00	0.91

7.3. Mobilenet

Mobilenet fue propuesto en [13]. Este modelo ligero utiliza convoluciones separables en profundidad, lo que reduce el número de parámetros en comparación con las redes con convoluciones regulares con la misma profundidad. Además, en Mobilenet se realiza una sola convolución en cada canal de color en lugar de combinarlos y aplanarlos. Como aplica su nombre, Mobilenet está diseñado para ser utilizado en aplicaciones móviles.

Las dimensiones de las imágenes fueron de $224 \times 224 \times 3$ para realizar los experimentos en las mismas condiciones que en VGG16. Para el conjunto de datos de lengua de señas de MNIST, se utilizaron 24,720 imágenes como conjunto de entrenamiento y 2,735 imágenes como conjunto de validación; utilizando el conjunto de entrenamiento se obtuvo una exactitud de 0.9989 y un error de 0.0042.

Con el conjunto de validación se obtuvo un valor de exactitud de 1.00 y un error de 0.0003. En el caso del conjunto de datos del Alfabeto ASL, el cual está conformado por 78,300 imágenes de entrenamiento y 8,700 imágenes de validacion. Utilizando el set de entranamiento se obtuvo una exactitud de 0.9947 y un error de 0.0158. Por otra parte, con el set de validación se obtuvo un valor de exactitud de 0.9291 y un error de 0.2754.

7.4. VGG siamesa

Para el aprendizaje de similitud semántica, se utilizaron dos VGG16 tipo D idénticas; estas dos redes comparten sus parámetros. La etiqueta binaria indica la similitud del par de imágenes. El número de neuronas en la última capa es 1000 para ambos conjuntos de datos.

Aquí, en el entrenamiento siamés, la última capa no contiene la probabilidad de que una imagen pertenezca a una determinada clase, sino un descriptor visual el cual es una codificación para representar el contenido de la imagen en un espacio euclidiano. Cuanto mayor sea la dimensión de la codificación de la imagen, mejor será la representación de la imagen; sin embargo, los descriptores visuales de grandes dimensiones representan una mayor demanda de recursos informáticos.

Después de varios experimentos, 1000 neuronas en la última capa mostraron el mejor rendimiento teniendo en cuenta la compensación entre la complejidad del cálculo, la precisión y las limitaciones del hardware.

En el caso del conjunto de datos de lenguaje de señas del MNIST, el tamaño de la imagen original es $28 \times 28 \times 1$; sin embargo, el tamaño de entrada mínimo para VGG16 es $32 \times 32 \times 1$ y, debido a que VGG16 es una red profunda, se decidió usar $64 \times 64 \times 1$

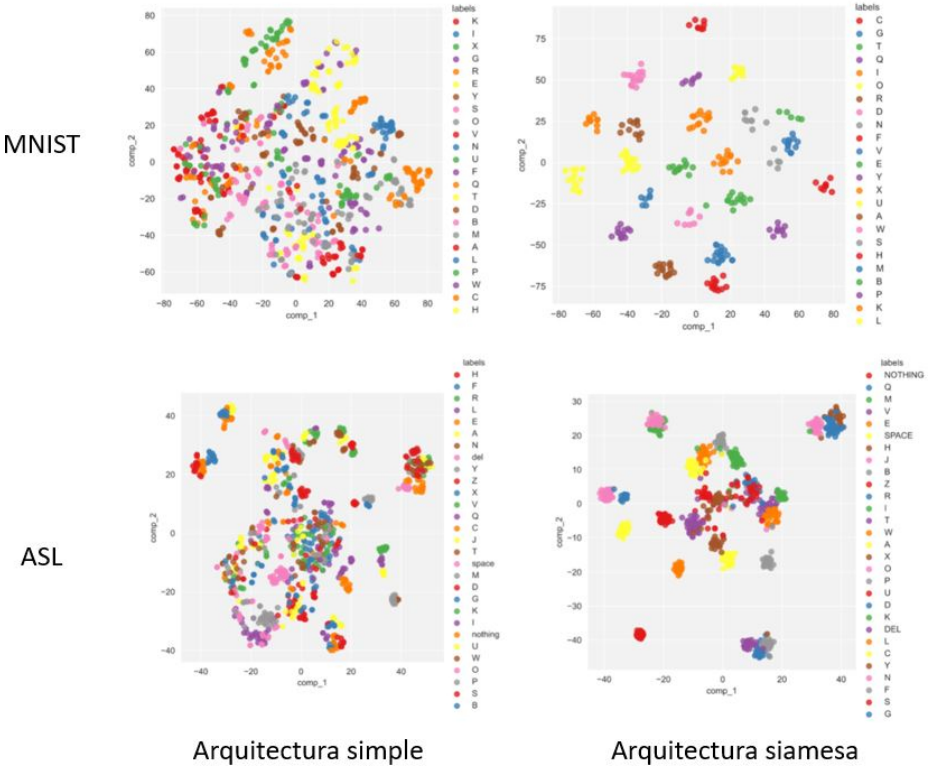

Fig. 5. TSNE.

como tamaño de imagen de entrada para para no perder información importante a través de las capas convolucionales. Los conjuntos de entrenamiento y validación están compuestos por 24,720 y 2,735 imágenes. La versión siamesa de VGG16 entrenada en el conjunto de datos MNIST Sing Language obtuvo un valor de exactitud de 0.9861 y un valor de 0.0199 de error, utilizando el set de entrenamiento. Utilizando el conjunto de validación se obtuvo una exactitud de 0.9834 y un error de 0.0237.

7.5. Mobilenet siamesa

Se utilizaron dos redes Mobilenet idénticas; la última capa contiene 1000 neuronas. Debido a limitaciones de hardware, las imágenes se redimensionaron a 64×64 para ambos conjuntos de datos; los hiperparámetros son básicamente los mismos que se usaron para la VGG16 siamesa, excepto por la cantidad de épocas.

Para el conjunto de datos de lengua de señas de MNIST, el conjunto de entrenamiento y validación estuvo compuesto por 24,709 y 2,746 imágenes, respectivamente. Los resultados de utilizando el conjunto de entrenamiento son los siguientes: error de 4.42E-4 y exactitud de 1.0. Con el conjunto de validación se obtuvo un error de 0.0061 y una exactitud de 1.0.

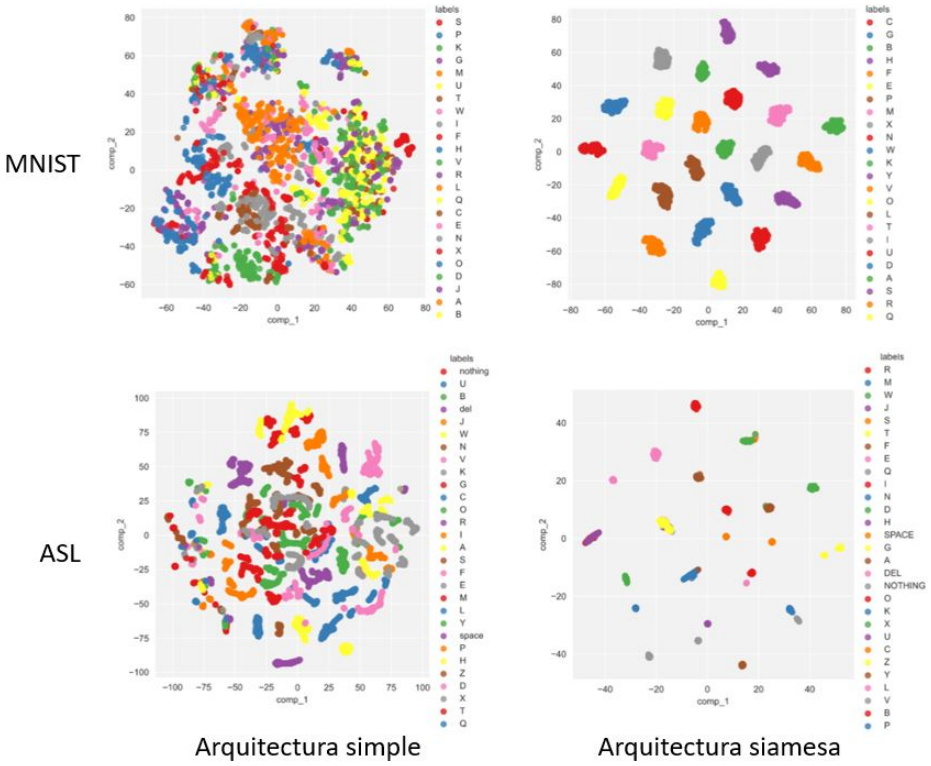

Fig. 6. TSNE.

Por otro lado, para el conjunto de datos de ASL Alphabet, solo se usó el 10 % del conjunto de datos debido a limitaciones de hardware; el conjunto de entrenamiento estuvo compuesto por 7,830 imágenes mientras que el conjunto de validación de 870 imágenes. Los resultados utilizando el conjunto de entrenamiento son los siguientes: error de 0.0064 y exactitud de 0.9925. Por otro lado, utilizando el conjunto de validación, se obtuvo un error de 0.0102 y una exactitud de 0.9888.

8. Resultados experimentales

El reconocimiento del alfabeto ASL se realiza mediante una tarea de clasificación, y para ello se utilizaron los modelos generados del entrenamiento de las redes mencionadas anteriormente. Para los modelos VGG16 y Mobilenet, la tarea de clasificación tiene como base alimentar una imagen del conjunto de prueba (muestras nunca vistas por la red) y dejar que la red prediga el alfabeto.

Las predicciones se compararon con los ejemplos reales. El rendimiento de la clasificación se midió utilizando las métricas más comúnmente utilizadas para la evaluación de la clasificación, como la exactitud, la precisión, recall y puntaje F1. En las Tablas 1 y 2 se muestran los resultados cuantitativos por parte de las arquitecturas simples y siamesas utilizando las dos bases de datos anteriormente mencionadas.

Tabla 3. Comparación de trabajos.

Referencia	**Trabajo**	**Conjunto** de **datos**	**Exactitud**
García et al. [8]	GoogleNet	ASL Alphabet	0.70
Hao et al. [10]	Autodestilación	ASL Alphabet	0.80
Método propuesto	VGG16 Siamesa	ASL Alphabet	0.89
Método propuesto	Mobilenet Siamesa	ASL Alphabet	0.91
LeCun et al. [19]	LeNet	MNIST	0.89
Krizhevsky et al. [16]	Alexnet	MNIST	0.94
Kaiming et al. [11]	ResNet18	MNIST	0.98
Kaiming et al. [11]	ResNet50	MNIST	0.98
Método propuesto	VGG16 Siamesa	MNIST	0.99
Método propuesto	Mobilenet Siamesa	MNIST	1.00

Se puede observar que para el caso de las arquitecturas siamesas, el resultado de la clasificación es mejor que el obtenido utilizando las arquitecturas simples, esto es debido a que el aprendizaje de similutd reduce la alta similitud entre clases y la alta variación dentro de la misma clase.

Una manera de comprobar esto es haciendo uso de la técnica t-SNE (t-distributed Stochastic Neighbor Embedding, en inglés) la cual es un algoritmo el cual permite visualizar datos multidimensionales. En la Figura 5 se presentan los resultados de t-SNE sobre la red VGG16 en su versión simple (primer columna) y su versión siamesa (columna 2) En cada renglón se presenta el resultado para cada uno de los conjuntos de datos.

Se puede observar que la arquitectura siamesa genera descriptores visuales de acuerdo a la similitud entre las imágenes, teniendo como resultado una disminución en la similitud entre clases (los grupos de cada letra estan más separados) y la reducción de la variación dentro de la misma clase (los grupos de descriptores visuales de una misma clase están más cerca).

En la Figura 6 podemos observar algo similar, los descriptores visuales que representan imágenes de la misma letra del alfabeto de la lengua de señas aparecen más cerca (se redujo la variación intra clase) y los descriptores visuales de diferentes letras aparacen separados (se redujo la similitud entre clase).

Para la predicción de una letra del alfabeto de la lengua de señas, se calculó el centroide de los descriptores visuales de las imágenes de entrenamiento de cada clase los cuales fueron generados por las redes siamesas. Para la predicción de un alfabeto de la lengua de señas se introdujo una imagen de prueba a la red siamesa, se obtuvo su vector característico y se calculo la distancia euclidiana con cada uno de los centroides de cada clase.

El centroide que esté mas cerca del descriptor visual de la imagen de prueba determinará qué seña el usuario está haciendo. El orden de complejidad de nuestro algoritmo para reconocer una letra nueva es de O(n^2). El sistema propuesto se evaluó con trabajos publicados en la literatura. En la Tabla 3 se muestra la comparación utilizando el conjunto de entrenamiento ASL Alphabet.

9. Conclusiones

La lengua de señas es la forma en que las personas con discapacidades auditivas y del habla se comunican con los demás, sin embargo, la mayoría de las veces, las personas oyentes no conocen esta lengua. Por lo tanto, existe una brecha de comunicación que impacta negativamente a la comunidad sorda. Por lo tanto, en este artículo, se presenta un método para el reconocimiento del alfabeto ASL.

Uno de los mayores desafíos en el reconocimiento del alfabeto ASL es la gran variación intraclase y la gran similitud entre las imágenes. Los métodos se enfocan en encontrar patrones para la clasificación alfabética de ASL. Para solucionar esto, se planteó la hipótesis de que, en caso de que sea posible generar patrones de acuerdo a la similitud de las imágenes, se podría mejorar el reconocimiento del alfabeto ASL.

Para ello, se implementó un aprendizaje de similitud semántica utilizando redes siamesas, que en pocas palabras, son dos redes idénticas que comparten sus parámetros. Los experimentos muestran que los descriptores visuales generados por las arquitecturas siamesas representan mejor a las imágenes al tener en cuenta las similitudes y diferencias entre ellas.

Otro hallazgo en los experimentos fue que a pesar de que el conjunto de entrenamiento para la CNN siamesa fue mucho menor en comparación con el conjunto de entrenamiento utilizado en una CNN simple, las redes siamesas no presentaron sobreajuste, esto se debe a la ayuda del aprendizaje one-shot. El aprendizaje one-shot permite que la red aprenda de solo unas pocas imágenes por clase.

Además, las redes siamesas son resistentes al desequilibrio de clases debido a que, al final del día, la red intenta aprender solo dos clases, pares de imágenes similares y no similares. El tiempo de entrenamiento y los requerimientos de memoria de hardware de las redes siamesas son los mayores inconvenientes porque involucra pares cuadráticos de muestras para aprender.

Referencias

1. Ameen, S., Vadera, S.: A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images. Expert Systems, vol. 34, no. 3, pp. e12197 (2017) doi: 10.1111/exsy.12197
2. Assaleh, K., Shanableh, T., Zourob, M.: Low complexity classification system for glove-based arabic sign language recognition. Neural Information Processing, Springer Berlin Heidelberg, pp. 262–268 (2012) doi: 10.1007/978-3-642-34487-9_32
3. Aly, W., Aly, S., Almotairi, S.: User-independent american sign language alphabet recognition based on depth image and PCANet features, vol. 7, pp. 123138–123150 (2019) doi: 10.1109/access.2019.2938829
4. Baker, S.: The importance of fingerspelling for reading. Visual Language and Visual Learning Science of Learning Center (2010)
5. Battison, R.: American sign language: Lexical borrowing in american sign language. University of California (1978)
6. Dong, C., Leu, M. C., Yin, Z.: American sign language alphabet recognition using Microsoft Kinect. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 44–52 (2015) doi: 10.1109/cvprw.2015.7301347

7. Deng, J., Dong, W., Socher, R., Li, L. J., Kai, L., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009) doi: 10.1109/cvpr.2009.5206848
8. Garcia, B., Viesca, S. A.: Real-time american sign language recognition with convolutional neural networks. Convolutional Neural Network and Visual Recognition, vol. 2, pp. 225–232 (2016)
9. Geddes, K. O., Czapor, S. R., Labahn, G.: Algorithms for Computer Algebra. Springer US (1992) doi: 10.1007/b102438
10. Hao, A., Min, Y., Chen, X.: Self-mutual distillation learning for continuous sign language recognition. In: IEEE/CVF International Conference on Computer Vision, pp. 11303–11312 (2021) doi: 10.1109/iccv48922.2021.01111
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition, pp. 770–778 (2015) doi: 10.48550/ARXIV.1512.03385
12. Holden, E. J., Lee, G., Owens, R.: Australian sign language recognition. Machine Vision and Applications, vol. 16, no. 5, pp. 312–320 (2005) doi: 10.1007/s00138-005-0003-1
13. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient convolutional neural networks for mobile vision applications (2017) doi: 10.48550/ARXIV.1704.04861
14. Kim, J. S., Jang, W., Bien, Z.: A dynamic gesture recognition system for the Korean sign language (KSL). IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), vol. 26, no. 2, pp. 354–359 (1996) doi: 10.1109/3477.485888
15. Kaggle: Dataset homepage (2023) www.kaggle.com/datasets/grassknoted/asl-alphabet
16. Krizhevsky, A., Sutskever, I., Hinton, G. E.: ImageNet classification with deep convolutional neural networks. Communications of the ACM, vol. 60, no. 6, pp. 84–90 (2017) doi: 10.1145/3065386
17. Kumar, P., Gauba, H., Pratim Roy, P., Prosad Dogra, D.: A multimodal framework for sensor based sign language recognition. Neurocomputing, vol. 259, pp. 21–38 (2017) doi: 10.1016/j.neucom.2016.08.132
18. Kuznetsova, A., Leal-Taixe, L., Rosenhahn, B.: Real-time sign language recognition using a consumer depth camera. In: IEEE International Conference on Computer Vision Workshops (2013) doi: 10.1109/iccvw.2013.18
19. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324 (1998) doi: 10.1109/5.726791
20. Maqueda, A. I., del-Blanco, C. R., Jaureguizar, F., García, N.: Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns. Computer Vision and Image Understanding, vol. 141, pp. 126–137 (2015) doi: 10.1016/j.cviu.2015.07.009
21. Nai, W., Liu, Y., Rempel, D., Wang, Y.: Fast hand posture classification using depth features extracted from random line segments. Pattern Recognition, vol. 65, pp. 1–10 (2017) doi: 10.1016/j.patcog.2016.11.022
22. Nanni, L., Ghidoni, S., Brahnam, S.: Handcrafted vs. non-handcrafted features for computer vision classification. Pattern Recognition, vol. 71, pp. 158–172 (2017) doi: 10.1016/j.patcog.2017.05.025
23. Oz, C., Leu, M. C.: Linguistic properties based on american sign language isolated word recognition with artificial neural networks using a sensory glove and motion tracker. Neurocomputing, vol. 70, no. 16–18, pp. 2891–2901, Oct. 2007. doi: 10.1016 /j.neucom.2006.04.016
24. Oz, C., Leu, M. C.: Recognition of finger spelling of american sign language with artificial neural network using position/orientation sensors and data glove. Advances in Neural Networks, pp. 157–164 (2005) doi: 10.1007/11427445_25

25. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014) doi: 10.48550/ARXIV.1409.1556
26. Tao, W., Leu, M. C., Yin, Z.: American Sign Language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion. Engineering Applications of Artificial Intelligence, vol. 76, pp. 202–213 (2018) doi: 10.1016/j.engappai.2018.09.006
27. Wang, C., Liu, Z., Chan, S. C.: Superpixel-based hand gesture recognition with kinect depth camera. In: IEEE Transactions on Multimedia, vol. 17, no. 1, pp. 29–39 (2015) doi: 10.1109/tmm.2014.2374357