

Reducción de las variaciones intraclase y las similitudes entre clases en el reconocimiento del alfabeto ASL mediante el aprendizaje de similitud semántica

Atoany Nazareth Fierro-Radilla^{1*}, Gibran Benitez-Garcia^{2†}
and Karina Ruby Perez-Daniel^{3†}

^{1*}Engineering and Sciences School, Tecnológico de Monterrey Cuernavaca campus, México 95D km 104, Xochitepec, 62790 , Morelos, Mexico.

²Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu, Tokyo, 182-8585, Tokyo, Japan.

³Engineering Faculty, Universidad Panamericana, 498 Augusto Rodin St., Mexico City, 03920 , CDMX, Mexico.

*Corresponding author(s). E-mail(s): afierro@tec.mx;
Contributing authors: gibran@ieee.org; kperezd@up.mx;

[†]These authors contributed equally to this work.

Resumen .El lenguaje de señas es un método importante que se utiliza para poder comunicarse entre la comunidad que padece alguna enfermedad auditiva, especialmente por personas con problemas del escucha y del habla. 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 dos redes convolucionales (CNN), VGG16 y Mobilenet; para poder mejorar la comunicación y las relaciones interpersonales con las personas que viven con sordera. La propuesta se basa en implementar aprendizaje de semántica similar para reducir la variación intra-clase y la alta similitud interclase en

un espacio euclidiano de las imágenes con signos. 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. Al aumentar la codificación de imágenes de la misma clase y reducirla en diferentes clases y usando la técnica de análisis estadístico t-SNE, el reconocimiento del alfabeto ASL mejora considerablemente, lo cual se demuestra en este trabajo.

Abstract. Sign language is an important method used to communicate among deaf community, especially by people with hearing and speech impairments. In the United States, approximately two million people living with hearing impairment use ASL (American Sign Language). Therefore, the objective of this study is to investigate and develop a system for ASL alphabet recognition using two Convolutional Neural Networks (CNN); VGG16 and Mobilenet, in order to improve communication and interpersonal relationships with people living with deafness. The proposal is based on implementing similar semantic learning to reduce intra-class variation and high inter-class similarity in a Euclidean space of signed images. The results show that the proposed system has an average accuracy of 99 % and 90.1 % in the MINIST and ASL dataset, respectively. By increasing the encoding of images of the same class and reducing it in different classes and using the t-SNE statistical analysis technique, ASL alphabet recognition is considerably improved, which is demonstrated in this work.

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 humana-máquina) [?], dicha interacción es un área de investigación multidisciplinaria con diversas aplicaciones en control de robótica, estudio de comportamiento psicológico, realidad virtual, reconocimiento de lengua de señas, visualización científica y, en el Metaverso [? ?], está constantemente definiendo nuevas modalidades de comunicación. El Reconocimiento de Lengua de Señas (RLS) se realiza por medio de la Interacción Humana-Computadora convirtiendo los gestos y movimientos en comandos de texto y/o de voz, permitiendo la comunicación entre los seres humanos a través de una computadora entre las personas que viven una discapacidad auditiva y las personas oyentes [? ?].

En los Estados Unidos (US) 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 [?]. El lenguaje de señas americano (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 [? ? ?].

A pesar de que ASL es el principal modo de comunicación para la mayoría de las personas sordas en EE. UU., sigue existiendo problemas de comunicación con las personas oyentes ya que no comprende el lenguaje ASL. Si la 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. [?].

El RLS requiere la combinación de información multimodal, por ejemplo, postura y movimiento de la mano, expresión facial y postura del cuerpo humano[? ?]. En general, ASL tiene cuatro aspectos importantes (Figura 1): (1) gestos de palabras para palabras de uso común; (2) deletreo con los dedos, a través del cual se asocian gestos manuales únicos a una letra en el alfabeto inglés (u otro idioma natural). El deletreo con los dedos se usa para comunicar palabras especializadas para las cuales los signos de ASL no existen o son desconocidos, como términos médicos; (3) emoción o expresión, a través de la cual se representan conceptos abstractos como intensidad y expresiones más finas; y (4) variaciones individuales o personalizadas [?]. En este artículo se propone un nuevo método de reconocimiento de deletreo manual basado en el aprendizaje de similitud semántica utilizando arquitecturas siamesas.

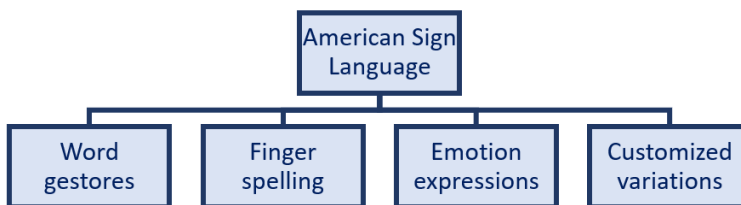


Figura 1 Los gestos de palabras, la ortografía con los dedos, las expresiones de emoción y las variaciones personalizadas son los cuatro componentes de ASL que un hablante usa simultáneamente.

Sistema de reconocimiento ASL

hace mas de dos décadas se han iniciado diversos trabajos de investigacion para el reconocimiento del lenguaje de señas, con grandes avances, en especial para los [? ? ?] estadounidenses, [?] australianos, [?] los indús, los coreanos [?], el chino [?], el polaco [?] y el árabe [? ? ?], sin embargo, existen grandes dificultades debido a la complejidad de los movimientos de manos y cuerpo en Expresiones en lengua de señas (LS) [?]. Los enfoques utilizados para resolver los problemas de SLR se pueden clasificar en dos métodos principaleslos cuales estan divididos en sensores e identificación de imagen[? ? ?]. 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 [? ? ?], que puede proporcionar información bastante precisa de la mano [? ?]. Sin embargo, son demasiado pesados e incómodos para el uso diario [?]. Por otro lado, los métodos basados en la identificación de imagen se basan 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 en color sin necesidad de sensores conectados al ser humano [? ? ?]. Este documento se centra únicamente en el enfoque basado en la visión.

Deletreo

El deletreo con los dedos es la representación de cada letra del alfabeto mediante un signo (Figura 2). Los usuarios de ASL usan el alfabeto deletreado con los dedos y manos (AFA), se pueden tener diversas variaciones del alfabeto, así como diferentes señas que representen un significado distinto. El AFA consta de 22 formas de manos que cuando se mantienen en ciertas posiciones y/o se producen con ciertos movimientos, representan las 26 letras del alfabeto inglés [? ?]. Las investigaciones coinciden en que la ortografía digital está integrada en ASL de manera muy sistemática [? ?]. Uno de los usos es principalmente para representar nombres propios o palabras en inglés sin equivalentes en lenguaje de señas [? ? ?]. Además, la ortografía digital es una parte importante del lenguaje de señas para los nuevos usuarios y ayuda a las personas a abreviar señas más largas, a comunicar dos palabras compuestas [?] y a cerrar la brecha entre el léxico ASL a través de la geografía y las culturas [? ?]. La incorporación de la ortografía de los dedos en ASL resulta más conveniente en escenarios críticos, como es en ámbito de la medicina. En la Figura 3 podemos ver un ejemplo de deletreo manual para la palabra ASL.

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 los signos 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. 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 por si el pulgar está entre el primer y el segundo o entre el segundo y el tercer dedo (Figura 4). 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 firmante con respecto a la cámara (Figura 5).

Este desafío puede resolverse mediante el aprendizaje de similitud semántica, que pueden realizar las redes siamesas [?]. La red Siamesa aprende

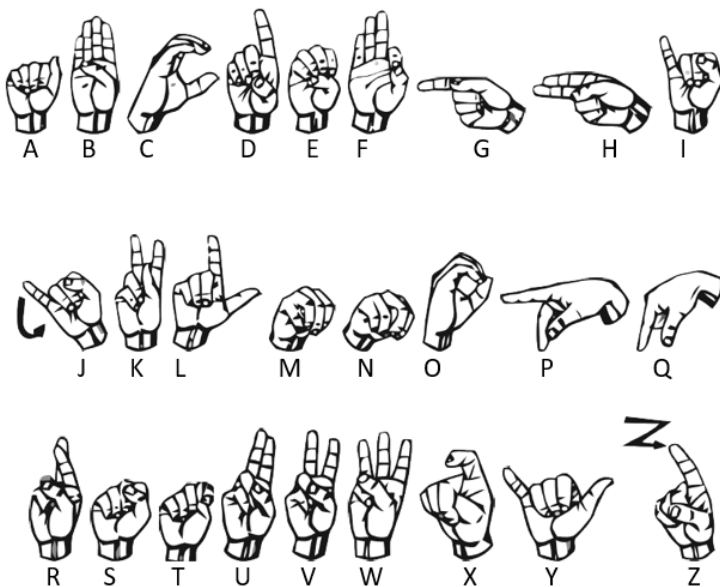


Figura 2 ASL finger spelling alphabet [?]. Letter J and Z involve motion



Figura 3 ASL finger spelling

similitudes semánticas entre clases/conceptos cercanos en niveles más profundos de abstracción, es decir, incrustaciones de imágenes. Esto significa que el rendimiento de las arquitecturas de las subredes tiene una gran influencia en el rendimiento de la red siamesa. Luego, por lo que se propone evaluar el rendimiento de dos arquitecturas mientras se utilizan como subredes en una red siamesa para el reconocimiento del alfabeto ASL. Se realizaron dos Redes Siamesas utilizando VGG16 [?] y Mobilenet [?] como subredes, respectivamente. El rendimiento se presenta de acuerdo con dos conjuntos de datos diferentes. Los conjuntos de datos analizados durante el estudio actual están disponibles en el repositorio de Kaggle, <https://www.kaggle.com/datasets/grassknoted/asl-alphabet> y <https://www.kaggle.com/datasets/datamunge/sign-language-mnist>.

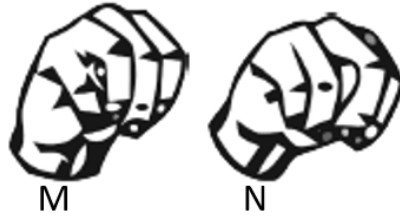


Figura 4 Similitud Interclase: La letra M y la letra N solo difieren en la posición del pulgar

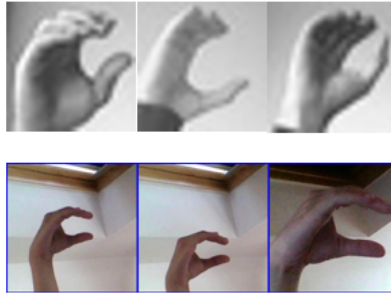


Figura 5 Variación intraclase: la letra C. Para una misma letra, tenemos variaciones en iluminación, color de piel, variaciones de fondo y la posición relativa del firmante a la cámara.

Trabajos relacionados

El reconocimiento del alfabeto ASL se basa principalmente en dos sub-tareas: extracción de características y clasificación multiclase [?]. En el primer proceso se realiza la detección y extracción de características locales. Por otro lado, en este último proceso, estas características extraídas son comprendidas y caracterizadas para clasificar las muestras [?]. Para el reconocimiento del alfabeto ASL, en la literatura se pueden encontrar dos enfoques que se han utilizado: métodos basados en artesanía y basados en CNN.

Métodos hechos a mano

Hecho a mano significa que el algoritmo de extracción de características ha sido manualmente construido [31]. Para los enfoques tradicionales es necesario diseñar el mejor algoritmo de extracción de características que se adapte; además, una vez que la característica se selecciona, se debe elegir un clasificador de tal manera que se ajuste con la etapa de extracción de características. Los primeros sistemas de reconocimiento del alfabeto ASL se proponen utilizando filtros de Gabor para la selección de características y random forest para la clasificación [27], el autor obtuvo un 49 % de precisión. Algunos autores [? ? ?] consideraron trabajar con información de profundidad en lugar de

imágenes en color. Extraen la característica de profundidad de las imágenes y utilizando random forest, se obtuvo un valor de 81,1 %, 87 % y 90 % de precisión, respectivamente. Estos métodos mencionados anteriormente se basan en dos tareas separadas, extracción y clasificación de características, luego, los dos algoritmos deben elaborarse por separado. Esto produce un "fenómeno de desacoplamiento", donde se pierde información en la característica extraída y se propaga en la tarea de clasificación.

Métodos basados en redes neuronales CNN

En 2012, las redes neuronales convolucionales (CNN) se hicieron muy populares entre la comunidad de visión artificial debido al éxito de Alexnet en el desafío de reconocimiento visual ImageNet a gran escala (ILSVRC). Una de las ventajas de CNN sobre métodos artesanales es que las CNN extraen y clasifican características en el mismo algoritmo y no hay necesidad de intervención humana en ninguna de sus etapas. En otras palabras, la red aprende qué características son las mejores para ser extraídas para una mejor resultado de la clasificación. En general, una CNN se compone de capas convolucionales (compuesto por convolución y operaciones de agrupación máxima), totalmente conectado en capas. Los primeros obtienen representaciones no lineales de imágenes (característica extracción), y este último genera la incrustación y realiza la clasificación de las características de la imagen extraída. Uno de los métodos basados en CNN que se encuentra en la literatura es [?], donde los autores propusieron utilizar una red basada en CNN de dos entradas, uno para imágenes en color y el otro para imágenes de profundidad. Las capas FC se alimentan con la concatenación de las representaciones no lineales de imágenes de color y profundidad introducidos en la red de forma separada. La clasificación está hecha por las capas FC obteniendo un 80,3 % de precisión. En la actualidad, un conjunto de vistas adicionales se generan a partir de esas cámaras virtuales distribuidas. En [?] también se utilizan características de profundidad; estos autores utilizan el sensor Microsoft Kinect como extractor de características y usan PCANet para clasificarlas, obteniendo una precisión del 84,5 %. En este artículo hicieron uso de dos ramas de capas como generadores de incrustaciones de imágenes. Esta doble arquitectura, conocida como Siamese Network se explica en detalle a continuación.

Arquitectura Siamesa CNN

CNN se basa en analizar una gran cantidad de datos para funcionar bien. Sin embargo, es complicado para algunas aplicaciones obtener esa gran cantidad de muestras de entrenamiento. La generación del conjunto de entrenamiento para una red siamesa se basa en la selección aleatoria de un par de imágenes; este par de imágenes pueden ser positivas (ambas imágenes pertenecen a la misma clase) y negativas (muestras pertenecientes a diferentes clases). Para un conjunto de datos de muestras de entrenamiento de n , se obtienen un total de C pares, donde C (combinación) se puede calcular como:

$${}_nC_r = \frac{n!}{r!(n-r)!} \quad (1)$$

donde $r = 2$ porque necesitamos combinar las muestras de n en pares. Esto hace que las redes siamesas sean más robustas al desequilibrio de clases. Una red siamesa se basa en dos redes neuronales convolucionales idénticas que comparten sus parámetros (como se muestra en la Figura 6) y se utilizan para aprender similitudes semánticas y encontrar similitudes de los resultados comparando la incrustación que genera cada red [?]. La hipótesis de este artículo es que usando una arquitectura siamesa, se puede reducir la alta similitud interclase y las altas variaciones intraclase.

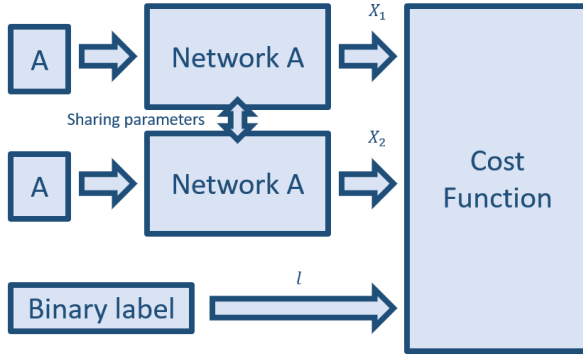


Figura 6 La arquitectura siamesa está compuesta por dos redes idénticas (A y B). Estas dos redes se alimentan con una muestra de imagen. La capa de función de costo se alimenta con incrustaciones de imágenes (generadas por las dos redes) y una etiqueta binaria; esta etiqueta binaria indica si el par es positivo (A y B de la misma clase) o negativo (A y B de diferentes clases)

La CNN siamesa se propuso [?] para la verificación de firmas y luego se usó para el reconocimiento facial [?]. Algunos años más tarde, este tipo de redes se utilizaron en la comparación de imágenes como en [?]. Específicamente, una CNN siamesa (Figura 6) está compuesta por dos CNN idénticas que son alimentadas por una muestra de imagen (A y B). Hay una etiqueta binaria que indica si el par de imágenes es positivo (A y B de la misma clase) o negativo (A y B de diferentes clases). Cada imagen es procesada a través de su rama correspondiente de la red siamesa. La función de costo es la responsable de realizar el aprendizaje de similitud semántica.

La función de costo utilizada en este documento se llama Pérdida contrastiva (presentada por primera vez en [?]) y se define de la siguiente manera: Sean X_1 y X_2 un par de vectores de entrada que se muestran al sistema. Sea Y una etiqueta binaria asignada a este par. $Y = 0$ si X_1 y X_2 son imágenes

de la misma clase, y $Y = 1$ si pertenecen a clases diferentes. Defina la función de distancia parametrizada a aprender D_W entre X_1 y X_2 (la distancia euclidiana):

$$D_W(X_1, X_2) = \sqrt{\sum_{i=1}^n (X_1^i - X_2^i)^2} \quad (2)$$

Para acortar la anotación, $D_W(X_1, X_2)$ se escribirá como D_W . Entonces, la forma general de la función de pérdida contrastiva se definirá como:

$$Loss(W) = \sum_{i=1}^P (L(W, (Y, X_1, X_2))^i) \quad (3)$$

$$(L(W, (Y, X_1, X_2))^i) = (1 - Y)L_S(D_W^i) + YL_D(D_W^i) \quad (4)$$

donde $(Y, X_1, X_2)^i$ es el i -ésimo par de muestras etiquetadas, L_S es la función de pérdida parcial para un par de imágenes similares, L_D la función de pérdida parcial para un par de imágenes diferentes imágenes, y P el número de pares de entrenamiento. L_S y L_D deben diseñarse de manera que minimizar L con respecto a W resulte en valores bajos de D_W para imágenes similares y valores altos de D_W para imágenes diferentes. Por lo tanto, la función de pérdida contrastiva exacta es:

$$L(W, (Y, X_1, X_2)) = (1 - Y)\frac{1}{2}(D_W)^2 + Y\frac{1}{2}(\max(0, m - D_W))^2 \quad (5)$$

donde $m > 0$ es un margen que define un radio alrededor de X_1 o X_2 . Las imágenes diferentes contribuyen a la función de pérdida solo si sus distancias euclidianas están dentro de este radio. La idea de usar una arquitectura siamesa con función de pérdida contrastiva es reducir la alta similitud entre clases y las altas variaciones intraclase en el reconocimiento del alfabeto ASL. En la Figura 7 podemos observar un ejemplo de esto.

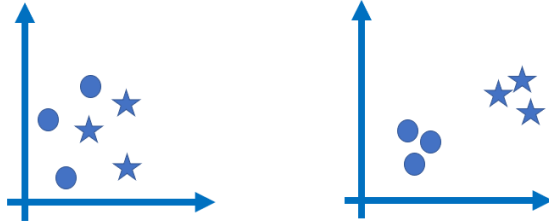


Figura 7 Antes del entrenamiento, las incrustaciones de imágenes de la clase círculo y la clase estrella están muy juntas (alta similitud entre clases y alta variación intraclase), sin embargo, después de la implementación del aprendizaje de similitud semántica, las incrustaciones de imágenes similares aparecen más cerca y las incrustaciones de imágenes diferentes aparecen separadas.

Sistema propuesto para el Reconocimiento del Alfabeto ASL

Los experimentos se realizaron utilizando dos arquitecturas CNN diferentes, VGG16 y Mobilenet, así como sus versiones siamesas, en dos conjuntos de datos diferentes, del alfabeto y lenguaje de señas MNIST. El entrenamiento se hizo utilizando Keras y Tensorflow como marcos en una máquina Intel Core i7 con un GPU NVIDIA GeForce RTX 2070 SUPER.

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 y que no contienen muestras para J y Z debido a 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 papel, "Jz "Z"se consideran signos estáticos.

VGG16

VGG16 se propuso por primera vez en [?] y logró una precisión de prueba del 92,7% entre los cinco primeros en ImageNet [?], que es un conjunto de datos compuesto por más de 14 millones de imágenes de 1000 clases diferentes. Este modelo se envió a ILSVRC-2014, mejorando el rendimiento de Alexnet al reemplazar el tamaño de kernel grande con kernels 3x3. La función de grupo utilizada por la red VGG16 es una capa de grupo máximo de 2x2 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 3x3 con zancada ($s = 1$) y usar el mismo relleno. En la Tabla ?? se presentan los diferentes tipos de red VGG; en este trabajo se utilizó el tipo D.

En el entrenamiento de ambos conjuntos de datos, las imágenes se redimensionaron a 224x224x3 debido a los requisitos de forma de entrada de la red. En Table 1 se muestran los hiperparámetros utilizados. 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 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. A partir de estos resultados de entrenamiento y de la Figura 8 se puede observar que la red no está sobreajustada.

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. En este experimento se utilizaron 78.300 imágenes de 224x224x3 como conjunto

de entrenamiento y 8.700 imágenes de 224x224x3 como conjunto de validación. Los resultados del entrenamiento fueron: precisión de entrenamiento de 0,9285, pérdida de entrenamiento de 5,5736, precisión de validación de 0,8477 y pérdida de validación de 16,2988. De la Figura 9 se puede observar que la red está saturada.

Cuadro 1 Hyper parámetros usados para entrenar la red VGG16

Hyper parámetro	MNIST Lengua de señas Dataset	ASL Alfabeto Dataset
Input shape	224x224x1	224x224x3
Number de épocas	100	50
Batchsize	8	8
Optimizador	SGD	SGD
Initial Learning Rate	1E-2	1E-2
Momentum	0.9	0.9
Loss function	Cross Entropy	Cross Entropy

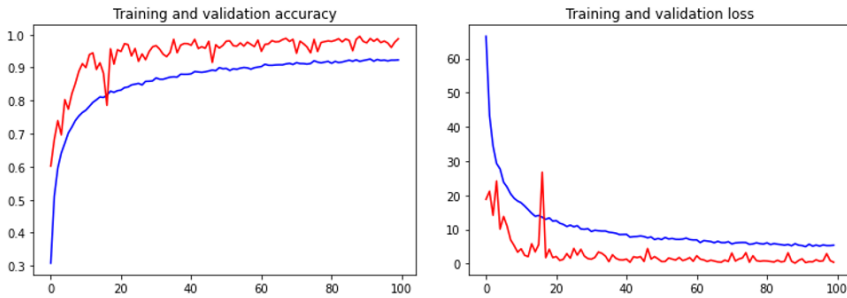


Figura 8 VGG16 type D training curves using MNIST Sign Language dataset. On the left side it is shown the accuracy and on the right side the loss. Training (blue) and validation (red) curves are close to each other, meaning the network is not overfitted

Mobilenet

Mobilenet fue propuesto en [?]. 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. En Table ?? se presenta la arquitectura de mobilenet. La forma de entrada de las imágenes fue de 224x224x3 para realizar los experimentos en las mismas condiciones que en VGG16. Los hiperparámetros utilizados en los experimentos de Mobilenet se presentan en In Table 2. Para el conjunto de datos de lenguaje de señas

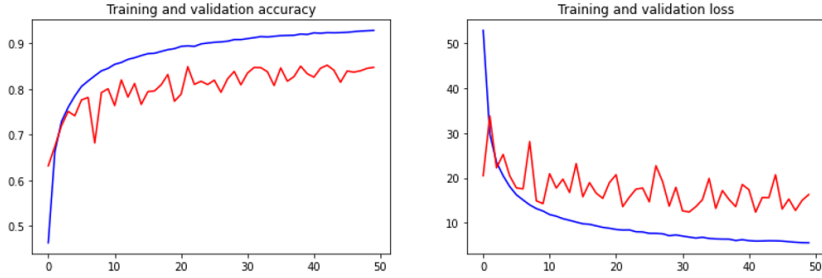


Figura 9 Curvas de entrenamiento VGG16 tipo D utilizando el conjunto de datos ASL Alphabet. En el lado izquierdo se muestra la precisión y en el lado derecho la pérdida. Las curvas de entrenamiento (azul) y validación (roja) están separadas, lo que significa que la red está sobreajustada

del MNIST, se utilizaron 24.720 imágenes como conjunto de entrenamiento y 2.735 imágenes como conjunto de validación, obteniendo un valor de pérdida de entrenamiento de 0,0042, precisión de entrenamiento de 0,9989, pérdida de validación de 0,0003 y precisión de validación de 1,00. Estos resultados se presentan en la Figura 10. Para el entrenamiento del Alfabeto ASL, el conjunto de entrenamiento y el conjunto de validación están compuestos por 78.300 y 8.700 imágenes, respectivamente, obteniendo un valor de precisión de entrenamiento de 0,9947 y un valor de pérdida de entrenamiento de 0,0158; para validación, 0,9291 de precisión y 0,2754 para pérdida. Estos valores muestran que la red está sobreajustada, como se puede ver en la Figura 10. Para el entrenamiento del Alfabeto ASL, el conjunto de entrenamiento y el conjunto de validación están compuestos por 78.300 y 8.700 imágenes, respectivamente, obteniendo un valor de precisión de entrenamiento de 0,9947 y un valor de pérdida de entrenamiento de 0,0158; para validación, 0,9291 de precisión y 0,2754 para pérdida. Estos valores muestran que la red está sobreajustada, como se puede ver en la Figure 11.

Cuadro 2 Hyper parameters used for Mobilenet training

Hyper parameter	MNIST Sign Language Dataset	ASL Alphabet Dataset
Input shape	224x224x1	224x224x3
Number of epochs	50	50
Batchsize	8	8
Optimizer	Adam	Adam
Initial Learning Rate	1E-5	1E-5
Learning rate decay factor	1E-4	1E-4
Loss function	Cross Entropy	Cross Entropy

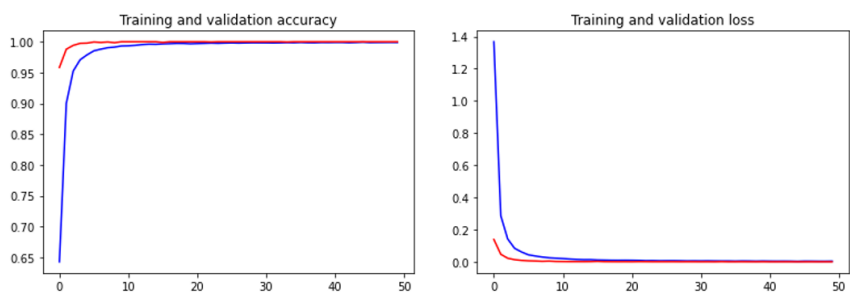


Figura 10 Curvas de entrenamiento de Mobilenet utilizando el conjunto de datos MNIST ASL. Las curvas de precisión de entrenamiento y pérdida de entrenamiento se presentan en azul, mientras que la precisión de validación y la pérdida de validación se presentan en rojo. De acuerdo con esta imagen, se puede notar que la red no está sobreajustada.

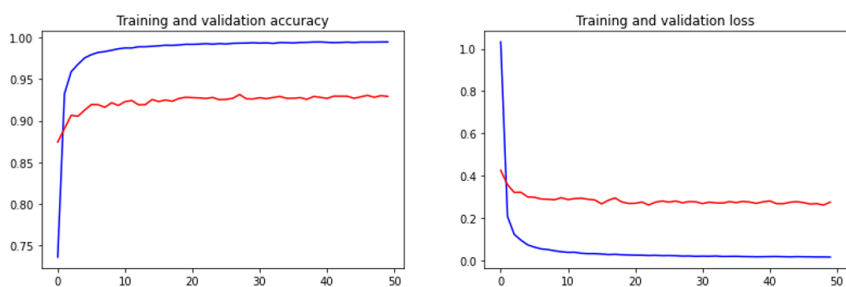


Figura 11 Curvas de entrenamiento de Mobilenet utilizando el conjunto de datos ASL Alphabet. En el lado izquierdo, se presentan las curvas de entrenamiento y validación y, en el lado derecho, las curvas de pérdida correspondientes.

Siamese VGG16

Para el aprendizaje de similitud semántica, se utilizaron dos VGG16 tipo D idénticos; 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 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, las incrustaciones de imágenes grandes 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 la Figura 12 esta capa aparece coloreada en gris.

En el caso del conjunto de datos de lenguaje de señas del MNIST, el tamaño de la imagen original es 28x28x1; 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 $64 \times 64 \times 1$ como tamaño de entrada de imagen 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, respectivamente; la configuración de los hiperparámetros se muestra en Table 3. La versión siamesa de VGG16 entrenada en el conjunto de datos MNIST Sing Language logró 0,9861 de precisión de entrenamiento, 0,0199 de pérdida de entrenamiento, 0,9834 de precisión de validación y 0,0237 de pérdida de validación.

Por otro lado, para el conjunto de datos ASL Alphabet, debido a limitaciones en la memoria del hardware, solo se utilizó el 10 % del conjunto de datos completo; los conjuntos de entrenamiento y validación estuvieron compuestos por 7.830 y 870 imágenes de $224 \times 224 \times 3$ píxeles; los hiperparámetros utilizados para el entrenamiento se presentan en Table 3. La precisión del entrenamiento fue de 0,9529, la pérdida de entrenamiento de 0,0596, la precisión de validación de 0,9332 y la pérdida de validación de 0,0779. En la Figura 13 y la Figura 14 se presentan las curvas de entrenamiento para los conjuntos de datos del lenguaje de señas MNIST y del alfabeto ASL, respectivamente.

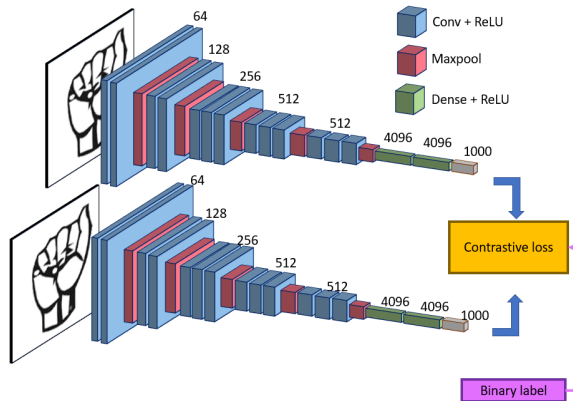


Figura 12 Arquitectura siamesa VGG16. La etiqueta binaria indica la similitud del par de imágenes. La pérdida de contraste hace que las redes generen incrustaciones de imágenes de acuerdo con la similitud del par de imágenes. La última capa, coloreada en gris, se usa como vector de características de la imagen

Siamese Mobilenet

Se utilizaron dos redes Mobilenet idénticas con la misma arquitectura utilizada en la sección 4.3. La última capa contiene 1000 neuronas como se puede ver en la Figura 15. Debido a limitaciones de hardware, las imágenes se redimensionaron a 64×64 para ambos conjuntos de datos; los hiperparámetros se pueden ver en Table 4 que son básicamente los mismos que se usaron para el siamés VGG16, excepto por la cantidad de épocas. Para el conjunto de

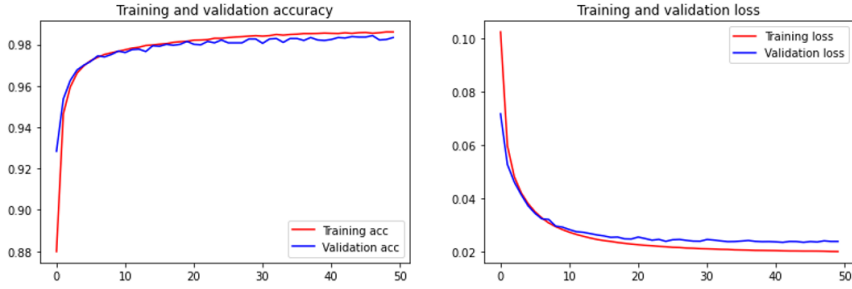


Figura 13 Curvas de entrenamiento del siamés VGG16 en el conjunto de datos de lenguaje de señas del MNIST. Las curvas roja y azul corresponden a conjuntos de entrenamiento y validación, respectivamente.

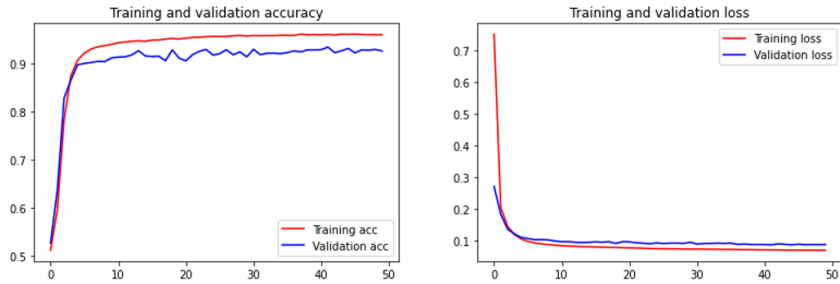


Figura 14 Curvas de entrenamiento del siamés VGG16 en ASL Alphabet Dataset. Las curvas roja y azul corresponden a conjuntos de entrenamiento y validación, respectivamente.

Cuadro 3 Hyper parameters used for Siamese VGG16 type-D network training

Hyper parameter	MNIST Sign Language Dataset	ASL Alphabet Dataset
Input shape	64x64x1	64x64x3
Number of epochs	50	50
Batchsize	4	4
Optimizer	RMSprop	RMSprop
Initial Learning Rate	1E-4	1E-4
Learning rate decay factor	1E-6	1E-6
Loss function	Contrastive loss	Contrastive loss

datos de lenguaje de señas del MNIST, el conjunto de entrenamiento y validación estuvo compuesto por 24 709 y 2746 imágenes, respectivamente. Los resultados del entrenamiento son los siguientes: pérdida de entrenamiento de $4.42\text{E-}4$, precisión de entrenamiento de 1.0, pérdida de validación de 0.0061 y precisión de validación de 1.0.

Por otro lado, para el conjunto de datos de ASL Alphabet, solo se usó el 10 % del conjunto de datos completo 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 del entrenamiento son los siguientes: pérdida de entrenamiento de 0,0064, precisión de entrenamiento de 0,9925, pérdida de validación de 0,0102 y precisión de validación de 0,9888.

Estos resultados para ambos conjuntos de datos se muestran en la Figura 16 y en la Figura 17 donde se puede observar que las redes no están sobreajustadas.

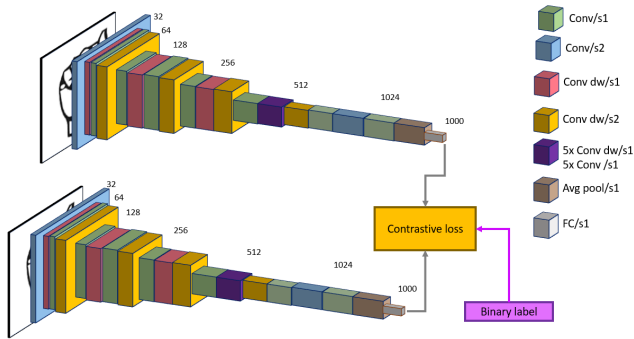


Figura 15 Arquitectura Mobilenet siamesa. La etiqueta binaria indica la similitud del par de imágenes. La pérdida de contraste hace que las redes generen incrustaciones de imágenes de acuerdo con la similitud del par de imágenes. La última capa, coloreada en gris, se usa como vector de características de la imagen

Cuadro 4 Hyper parameters used for Siamese Mobilenet training

Hyper parameter	MNIST Sign Language Datas	ASL Alphabet Dataset
Input shape	64x64x1	64x64x3
Number of epochs	15	
Batchsize	4	
Optimizer	RMSprop	
Initial Learning Rate	1E-4	
Learning rate decay factor	1E-6	
Loss function	Contrastive loss	

Resultados experimentales

El reconocimiento del alfabeto ASL se realiza mediante una tarea de clasificación, y para ello se utilizaron los modelos generados luego 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 mide utilizando la matriz de confusión y las métricas más

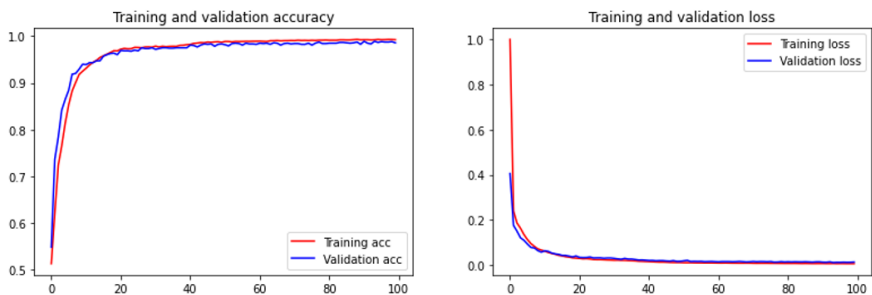


Figura 16 Hola Curvas de entrenamiento de Mobilenet siamés en el conjunto de datos de lenguaje de señas del MNIST. el rojo y el azul las curvas corresponden a conjuntos de entrenamiento y validación, respectivamente.

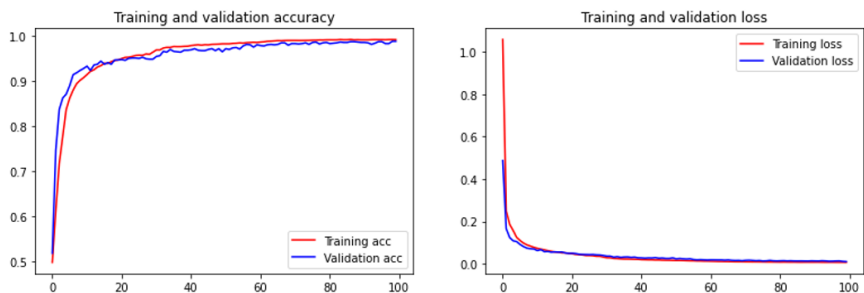


Figura 17 Curvas de entrenamiento de Mobilenet siamés en ASL Alphabet Dataset. el rojo y el azul las curvas corresponden a conjuntos de entrenamiento y validación, respectivamente.

comunes utilizadas para la evaluación de la clasificación, como la exactitud, la precisión, la recuperación y la puntuación F1. En la Figura 18 y la Figura 19 se pueden observar algunas clasificaciones erróneas para ambos conjuntos de datos usando ambas arquitecturas, esto se debe a las altas similitudes entre clases y las altas variaciones intra-entre la misma clase. El rendimiento general de VGG16 y Mobilenet en ambos conjuntos de datos se presenta en Table 5.

Por otro lado, para las arquitecturas siamesas, la tarea de clasificación se realiza por incrustaciones de imágenes cuyos elementos dependen de la similitud de la imagen de entrada de entrenamiento. Para la evaluación se alimentó al modelo con las imágenes de entrenamiento se para obtener un centroide de clase calculando el promedio de todos estos vectores, posteriormente para la predicción de clase, se utiliza una imagen para una red para obtener su imagen incrustada. Finalmente se hizo el cálculo de la distancia entre esta imagen incrustada y cada centroide de clase utilizando la misma distancia métrica del entrenamiento. La clase de centroide más cercana a la imagen incrustada

determina la clase predicha (consulte la Figura 20). Los resultados de las arquitecturas siamesas se presentan en la Figura 21 y la Figura 22. Además, en Table ??, se presenta la evaluación usando métricas.

Los resultados obtenidos por versiones siamesas de arquitecturas CNN muestran que el aprendizaje de similitud semántica mejora el reconocimiento del alfabeto ASL.

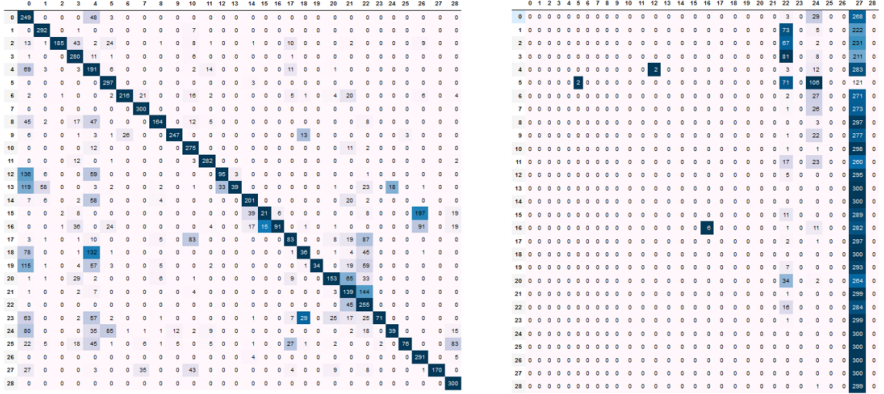


Figura 18 Matriz de confusión VGG16 (izquierda) y Mobilenet (derecha) utilizando el conjunto de datos de lenguaje de señas MNIST. Se puede observar que VGG16 funciona mejor que Mobilenet.

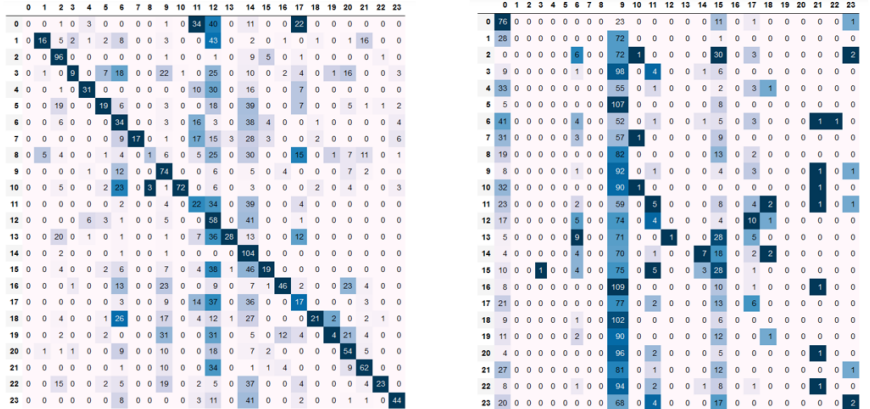
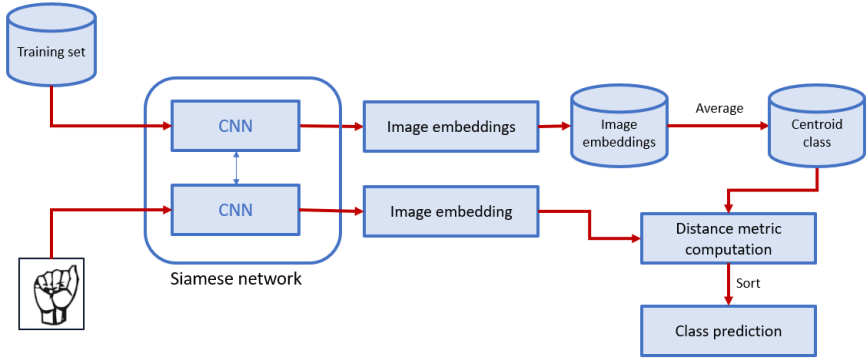


Figura 19 Matriz de confusión VGG16 (izquierda) y Mobilenet (derecha) utilizando ASL Alphabet Dataset. Se puede observar que VGG16 funciona mejor que Mobilenet.

Con el fin de mejorar la hipótesis de que el aprendizaje de similitud semántica podría mejorar el reconocimiento del alfabeto ASL, se realizó un análisis estadístico utilizando t-SNE (incrustación de vecinos estocásticos distribuidos

Cuadro 5 Classification report of single Architectures

Metric	VGG16		Mobilenet	
	MNIST Dataset	ASL Dataset	MNIST Dataset	ASL Dataset
Accuracy	0.58	0.30	0.08	0.04
Precision	0.71	0.47	0.08	0.07
Recall	0.58	0.30	0.08	0.04
F1-score	0.56	0.28	0.04	0.01

**Figura 20** Predicción de clases para arquitecturas siamesas. La clase de centroide se calcula promediando todas las incrustaciones de imágenes de entrenamiento. Dada una imagen de prueba, la distancia euclidiana se mide entre su incrustación de imagen y cada clase de centroide, la más cercana determina la clase predicha.**Cuadro 6** Classification report of Siamese Architectures

Metric	Siamese VGG16		Siamese Mobilenet	
	MNIST Dataset	ASL Dataset	MNIST Dataset	ASL Dataset
Accuracy	0.99	0.89	1.00	0.91
Precision	0.99	0.88	1.00	0.91
Recall	0.99	0.89	1.00	0.91
F1-score	0.99	0.89	1.00	0.91

en t). La técnica t-SNE es un método de reducción de dimensionalidad no lineal muy adecuado para incrustar datos de alta dimensión para su visualización en un espacio de baja dimensión (en este caso, en un espacio euclidiano 2D). En la Figura 23 se puede observar la distribución en un espacio euclidiano 2D de la última capa de arquitectura VGG16 (izquierda) y la distribución de las incrustaciones de imágenes generadas por siamese Vgg16, en el mismo espacio euclidiano (derecha). Cada punto 2D del espacio euclidiano representa una representación de imagen de 1000 dimensiones para cada muestra del conjunto de prueba del conjunto de datos del lenguaje de señas del MNIST.

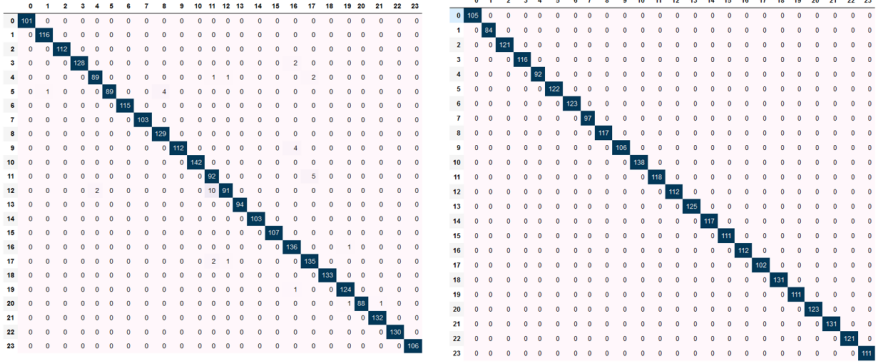


Figura 21 Matriz de confusión de Siamese VGG16 (izquierda) y Siamese Mobilenet (derecha) utilizando el conjunto de datos de lenguaje de señas del MNIST. Se puede observar un buen desempeño en la clasificación del alfabeto ASL.

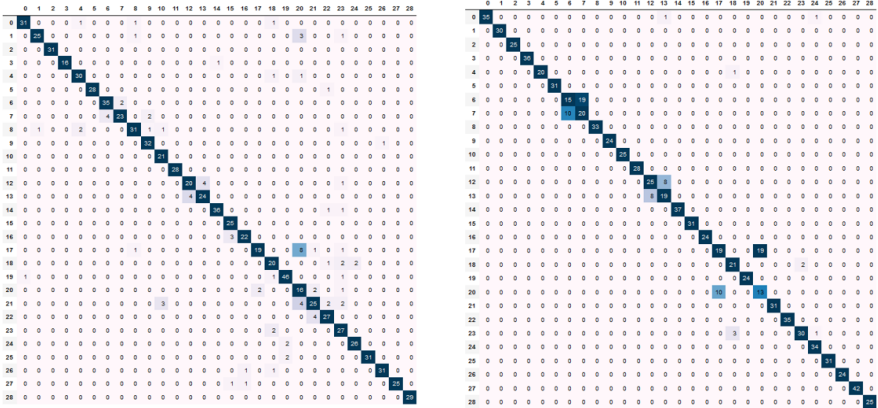


Figura 22 Matriz de confusión de Siamese VGG16 (izquierda) y Siamese Mobilenet (derecha) utilizando ASL Alphabet Dataset. Se pueden observar muy pocos errores de clasificación en comparación con los resultados de un solo VGG16 y un solo Mobilenet.

A partir de aquí, se puede observar que el siamés VGG16 crea vectores de características de acuerdo con la similitud del par de imágenes, lo que facilita el reconocimiento del alfabeto ASL. El mismo caso para VGG16 y VGG16 siamés en el conjunto de datos del alfabeto ASL, Mobilenet y Siamese Mobilenet en el conjunto de datos del lenguaje de señas MNIST y, Mobilenet y siamés Mobilenet en el conjunto de datos del alfabeto ASL en la Figura 24, Figura 25 y Figura 26, respectivamente.

Los resultados de las matrices de confusión que se muestran en la Figura 18, Figura 19, Figura 21, Figura 22 muestran que la hipótesis de que la arquitectura siamesa podría mejorar el reconocimiento del alfabeto ASL era cierto y la razón por la cual se puede encontrar en la Figura 23, Figura 24, Figura 25 y Figura 26, donde se puede observar que las arquitecturas siamesas

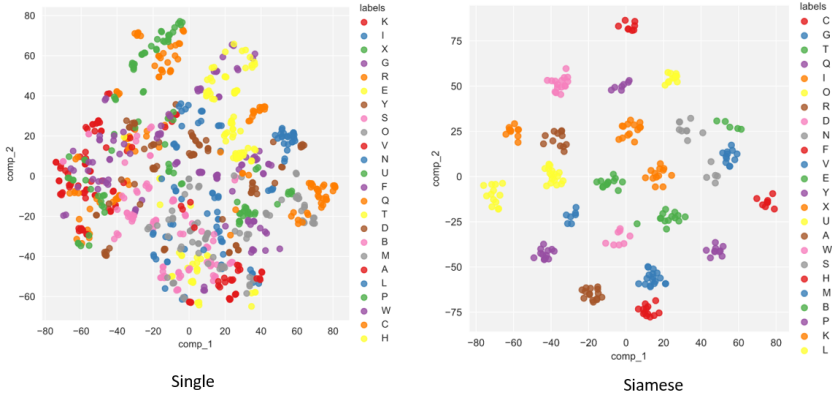


Figura 23 Análisis estadístico usando t-SNE en vectores de características de la última capa de VGG16 (izquierda) y las incrustaciones de imágenes generadas por siamés VGG16 (derecha) del conjunto de prueba del conjunto de datos de lenguaje de señas MNIST. Cuanto más separables sean las representaciones de imágenes, más fácil será la tarea de clasificación.

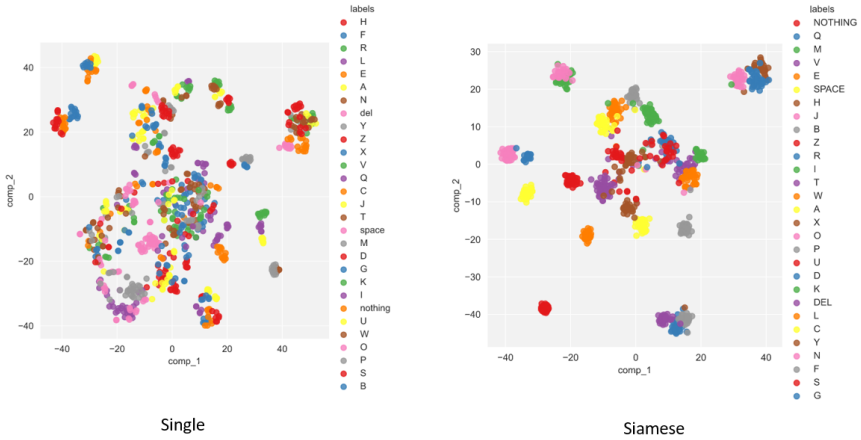


Figura 24 Análisis estadístico utilizando t-SNE en vectores de características de la última capa de VGG16 (izquierda) y las incrustaciones de imágenes generadas por siamés VGG16 (derecha) del conjunto de prueba del conjunto de datos del alfabeto ASL. Cuanto más separables sean las representaciones de imágenes, más fácil será la tarea de clasificación.

intente separar los vectores de características de las imágenes que pertenecen a diferentes clases e intente juntar los vectores de características de las imágenes de la misma clase. Esta recopilación facilita el reconocimiento de ASL porque las muestras son más separables.

Conclusiones

El lenguaje de señas es la forma en que las personas sordas se comunican con los demás, sin embargo, la mayoría de las veces, las personas oyentes

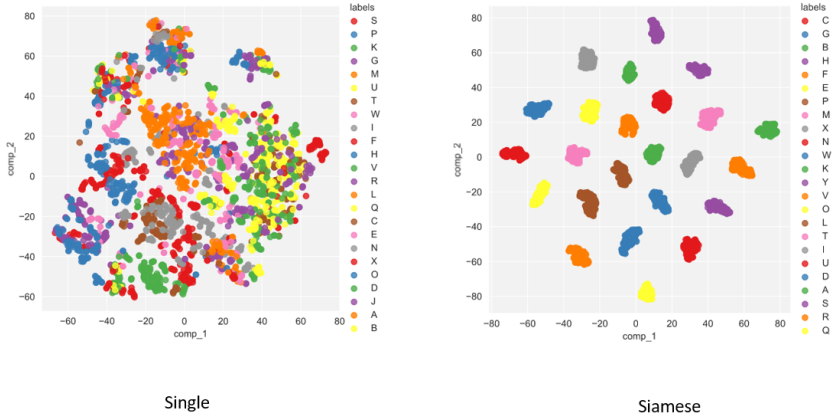


Figura 25 Análisis estadístico utilizando t-SNE en vectores de características de la última capa de Mobilenet (izquierda) y las incrustaciones de imágenes generadas por siamese Mobilenet (derecha) del conjunto de prueba del conjunto de datos de lenguaje de señas MNIST. Cuanto más separables sean las representaciones de imágenes, más fácil será la tarea de clasificación.

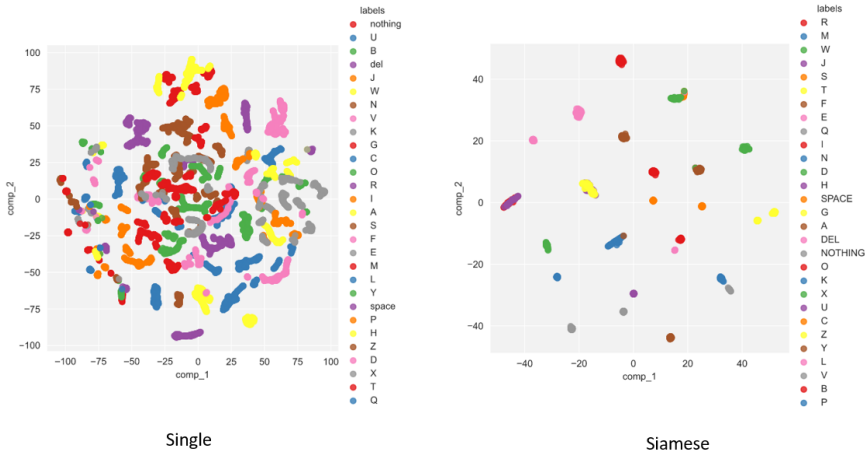


Figura 26 Análisis estadístico utilizando t-SNE en vectores de características de la última capa de Mobilenet (izquierda) y las incrustaciones de imágenes generadas por siamese Mobilenet (derecha) del conjunto de prueba del conjunto de datos ASL Alphabet. Cuanto más separables sean las representaciones de imágenes, más fácil será la tarea de clasificación.

no conocen este idioma. 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 novedoso para el reconocimiento del alfabeto ASL. Uno de los mayores desafíos en el reconocimiento del alfabeto ASL es la gran variación intraclass y la gran similitud entre clases entre las imágenes. Los métodos tradicionales luchan por encontrar patrones para la clasificación alfabética de ASL. Para solucionar esto, se planteó la hipótesis de que si era

posible generar patrones según 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 vectores de características generados por las arquitecturas siamesas son mejores representaciones de 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 sola CNN, las redes siamesas no presentaron sobreajuste, esto se debe a la ayuda del aprendizaje de un disparo. El aprendizaje de una sola vez 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 requisitos de memoria de hardware de las redes siamesas son los mayores inconvenientes porque involucra pares cuadráticos de muestras para aprender.