Revista ELECTRO, Vol. 46, 2024, pp. 333 - 338
https://itchihuahua.mx/revista_electro
ISSN Electrónico: 3061-774X

COSTO COMPUTACIONAL DE REDES NEURONALES PROFUNDAS PARA RECONOCIMIENTO DE EMOCIONES MEDIANTE EXPRESIONES FACIALES

Amparán Ortega Nirvana Cinnereth[1] Corral Sáenz Alma Delia[1] Ramírez Quintana Juan Alberto[1]
[1]Tecnológico Nacional de México/ I.T. Chihuahua
División de Estudios de Posgrado e Investigación
Av. Tecnológico 2909, Chihuahua, Chih., México, 31310, Tel. +52 (614) 2012000
L23061080@chihuahua.tecnm.mx, alma.cs@chihuahua.tecnm.mx, juan.rq@chihuahua.tecnm.mx

RESUMEN.

El reconocimiento de emociones es un área destacada en el contexto de la Inteligencia Artificial (IA) ya que tiene una amplia aplicación en computación afectiva. El enfoque que ha generado mejores resultados en el reconocimiento de emociones es la aplicación de las redes neuronales profundas en análisis de expresiones faciales; sin embargo, está asociado con una carga computacional considerable. Este estudio compara el costo computacional de las arquitecturas VGG16, ResNet50 e InceptionV3 usando la base de datos JAFFE. Se evalúa el rendimiento en dos entornos de hardware: Jetson TX2 y una computadora con procesador Intel Xeon E5-1603 v3, centrado en la velocidad de procesamiento (FPS) y consumo de memoria. Los resultados muestran una precisión del 81.25%, 71.88% y 43.75% para VGG16, InceptionV3 y ResNet50, respectivamente. La computadora con procesador Intel Xeon presentó un rendimiento superior a la tarjeta Jetson TX2, que no utilizó la GPU para la inferencia.

Palabras clave: VGG16, ResNet50, InceptionV3, FPS, redes neuronales profundas.

ABSTRACT.

Emotion recognition is a prominent area in the context of Artificial Intelligence (AI) due to its wide application in affective computing. The approach that has yielded the best results in emotion recognition is the application of deep neural networks in facial expression analysis; however, it is associated with a considerable computational load. This study compares the computational cost of the VGG16, ResNet50, and InceptionV3 architectures using the JAFFE database. The performance is evaluated in two hardware environments: Jetson TX2 and a computer with an Intel Xeon E5-1603 v3 processor, focusing on processing speed (FPS) and memory consumption. The results show an accuracy of 81.25%, 71.88%, and 43.75% for VGG16, InceptionV3 and ResNet50, respectively. The computer with the Intel Xeon processor demonstrated superior performance compared to the Jetson TX2, which did not utilize the GPU for inference.

Keywords: VGG16, ResNet50, InceptionV3, FPS, deep neural networks.

1. INTRODUCCIÓN

Las emociones son episodios de cambios coordinados de activación neuropsicológica, expresiones motoras y sentimientos subjetivos en respuesta a eventos externos o internos, como el comportamiento de otras personas, cambio de situación actual o un encuentro con un nuevo estímulo [1]. Los cambios emocionales son acompañados de cambios en señales físicas y/o fisiológicas. Las señales físicas son manifestaciones externas como expresiones faciales, tono de voz o escritura; y las señales fisiológicas corresponden a una forma interna de expresión como la actividad eléctrica cerebral, el ritmo cardíaco, la conductividad de la piel, la actividad muscular, entre otras. Es por esto que, las emociones humanas pueden ser clave para detectar condiciones fisiológicas de riesgo como fatiga, somnolencia y depresión. Las emociones activan diversos músculos faciales, generando expresiones, que son esenciales para la interacción humano-robot y la evaluación cerebro-computadora, donde las máquinas se diseñan para comportarse como humanos [1].

Ekman et al. [2] realizaron un estudio sobre expresiones faciales, descubriendo que la activación de músculos faciales está asociada con cambios correspondientes en la actividad del sistema nervioso autónomo. Una combinación específica de unidades de acción está involucrada en emociones particulares. El reconocimiento de emociones mediante expresiones faciales tiene un gran potencial en diversas aplicaciones. Por ejemplo, muchas empresas utilizan esta tecnología para obtener retroalimentación de clientes e identificar clientes potenciales. En el ámbito educativo, se emplea para ajustar la estrategia de enseñanza según las emociones detectadas. En hospitales, ayuda a evaluar y detectar el estado emocional de los pacientes inmovilizados [3].

Los métodos tradicionales de clasificación de emociones se han complementado con enfoques de aprendizaje profundo por medio de expresiones faciales. Las técnicas de extracción de rostros, características geométricas, texturas y patrones binarios han sido efectivas, al igual que los modelos de redes neuronales y módulos de atención. Las bases de datos son fundamentales para cualquier técnica de aprendizaje automático en el reconocimiento de emociones con expresiones faciales, CK+ y JAFFE son las más utilizadas [4].

Las redes neuronales profundas, han demostrado un alto poder de representación que permite abordar tareas de reconocimiento de emociones con un rendimiento cada vez más preciso. Sin embargo, esta capacidad viene acompañada de una carga computacional considerable, lo que plantea desafíos en términos de eficiencia y rendimiento, especialmente en entornos con recursos limitados como dispositivos móviles o embebidos. Los sistemas de reconocimiento de emociones se basan en el aprendizaje supervisado y constan de clasificadores binarios o

multiclase. Las entradas a estos clasificadores son varias señales y las etiquetas de clase de salida corresponden a un estado emocional (es decir, diferentes tipos y niveles de emoción).

Los primeros estudios utilizaron métodos de clasificación tradicionales para detectar emociones, como Análisis Discriminante Lineal (LDA), Análisis Discriminante Cuadrático (QDA), k-vecinos más cercanos (kNN), Bosques Aleatorios (RF) y Máquinas de Vectores de Soporte (SVM). Con los avances en los algoritmos de aprendizaje profundo, técnicas como Perceptrón Multicapa (MLP), Redes Neuronales Convolucionales (CNN) y Memoria a Largo Plazo (LSTM) han sido empleadas para el reconocimiento emocional. MLP, una red neuronal artificial, se destacó inicialmente por su precisión superior en esta tarea comparada con los algoritmos tradicionales. Las CNN han logrado éxitos notables en el ámbito de reconocimiento de patrones en imágenes y, recientemente, se han aplicado a señales fisiológicas como EEG, EMG y ECG. Las LSTM, un tipo especial de red neuronal recurrente (RNN) capaz de aprender dependencias a largo plazo, han sido comúnmente utilizadas en combinación con CNN para el reconocimiento emocional, aprovechando la capacidad de la CNN como extractor de características [4].

Existen arquitecturas de redes neuronales convolucionales profundas que han sido populares en el estado del arte debido al desempeño que obtuvieron en el desafío ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Las más conocidas son VGG16, Inception y ResNet. Diversos estudios han desarrollado arquitecturas modernas basados en estas redes debido a su eficacia en tareas de clasificación, adaptándolas al reconocimiento de emociones [5]. Aunque algunos de estos modelos han sido empleados para el reconocimiento de emociones, no se ha encontrado en la literatura un análisis del costo computacional de estos modelos en el reconocimiento de emociones con enfoque en expresiones faciales.

La importancia de comprender el costo computacional radica en la necesidad de seleccionar adecuadamente la arquitectura de red neuronal profunda en función de los recursos disponibles en la plataforma de implementación. Es por ello que, en este trabajo, se evalúa el rendimiento de las mencionadas arquitecturas en dos entornos de hardware diferentes: una tarjeta embebida NVIDIA Jetson TX2, conocida por sus capacidades de procesamiento eficiente de IA en dispositivos embebidos, y una computadora Dell Precision Tower 5810 con un procesador Intel Xeon E5-1603 v3 a 2.80 GHz y 4 núcleos, con 8 GB de memoria RAM y opera con Windows 10 Pro de 64 bits (versión 10.0, compilación 19045), representativa de hardware de alto rendimiento en entornos de escritorio. El análisis se centra en dos métricas principales: la velocidad de procesamiento medida en cuadros por segundo (FPS) y el consumo de memoria. Estas métricas son cruciales tanto para aplicaciones en tiempo real donde se requiere un procesamiento rápido, como para aplicaciones que operan en dispositivos con limitaciones de memoria.

Al comparar el rendimiento de las redes neuronales profundas en estos dos entornos, se busca proporcionar información para desarrolladores y diseñadores de sistemas de IA, permitiendo tomar decisiones informadas sobre la selección de arquitecturas de redes neuronales profundas para aplicaciones específicas en función de las restricciones de hardware y los requisitos de rendimiento.

El resto del articulo presenta la siguiente forma: en la sección 2 se describen la base de datos, y las arquitecturas de las redes neuronales utilizadas en este trabajo. La sección 3 describe el desarrollo, implementación y evaluación de los modelos. Finalmente, en la sección 4 se presentan los resultados y conclusiones obtenidos.

2. MATERIALES Y MÉTODOS

2.1. Base de Datos JAFFE.

El Conjunto de Datos de Expresión Facial Femenina Japonesa (JAFFE) consta de 213 imágenes de diferentes expresiones faciales de 10 mujeres japonesas diferentes. A cada sujeto se le pidió que hiciera 7 expresiones faciales (6 expresiones faciales básicas y 1 neutral). Cada imagen tiene una resolución de 256x256 pixeles y se encuentra en escala de grises de 8 bits en formato TIFF [6]. Una muestra aparece en la Figura 1.

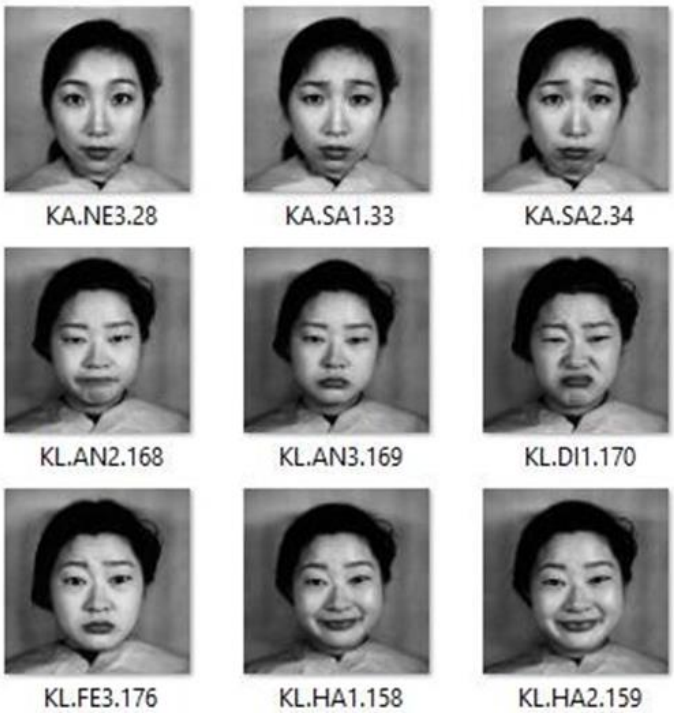

Figura 1. Ejemplo de la base de datos JAFFE.

2.2. Redes Neuronales Profundas.

Las Redes Neuronales Profundas (DNN) son un tipo de modelo de aprendizaje profundo que se inspira en la estructura y función del cerebro humano. La arquitectura de una DNN puede variar según la tarea específica, pero comúnmente se utilizan capas convolucionales, de agrupamiento y totalmente conectadas como se muestra en la Figura 2. Entre las arquitecturas más populares

se encuentran VGG16, ResNet50 e Inception, cada una con características y propósitos particulares.

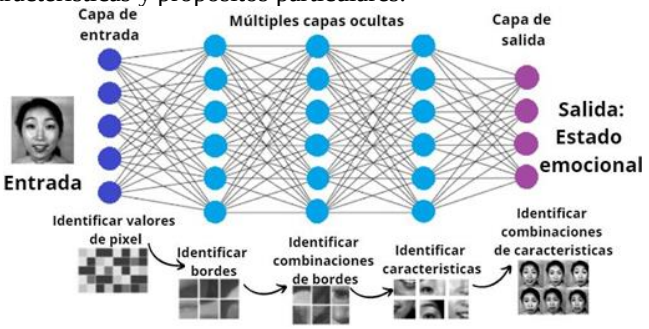

Figura 2. Ejemplo de estructura de una DNN.

2.3. Red VGG16.

Es una arquitectura estándar de Red Neuronal Convolucional Profunda (DCNN). El VGG-16 consta de trece capas convolucionales (Conv) y tres capas completamente conectadas (FC) [7]. Acepta un tamaño de entrada de imagen de 224×224 pixeles. Cada capa convolucional utiliza filtros de 3x3 que permite capturar las características locales en las imágenes de entrada, como bordes y texturas. Las capas de agrupamiento (Max Pool) reducen las dimensiones espaciales de los mapas de características Todas las capas ocultas de la red utilizan la función de activación ReLU. La arquitectura de esta red se muestra en la Figura 3.

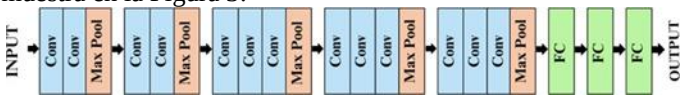

Figura 3. Arquitectura de la red VGG16 [8].

2.4. Red ResNet50.

ResNet50 es una arquitectura compuesta por 50 capas. La red incluye cuatro partes: capas Conv, bloques de identidad, bloques convolucionales y capas FC. Las capas Conv extraen características, seguidas de normalización por lotes (Batch Norm), activación ReLU y Max Pool para reducir dimensiones. Los bloques de identidad y convolucionales (ID block y Conv block) procesan y transforman características para añadir aprendizaje residual de entrada a salida. Finalmente, la agrupación promedio (Avg Pool) reduce dimensiones espaciales promediando valores dentro de un filtro. Se añaden ceros alrededor de la imagen para mantener el tamaño durante las convoluciones (Zero Padding). La capa de aplanado (Flattening) convierte un mapa de rasgos (matriz) en un vector unidimensional para las capas totalmente conectadas. Las capas FC realizan la clasificación final, con la salida alimentada a una función de activación Softmax para producir las probabilidades finales de clase [9]. La arquitectura se muestra en la Figura 4.

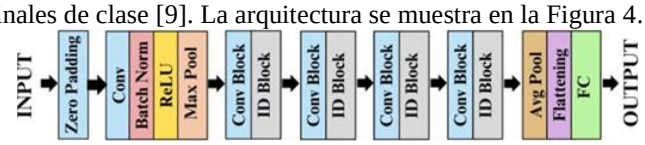

Figura 4. Arquitectura de la red ResNet50 [10].

2.5. Red Inception.

La arquitectura Inception introduce el concepto de "bloques Inception", que combina diferentes tamaños de filtros en una sola capa para extraer información a varias escalas en paralelo, aumentando la profundidad y el ancho de la red sin incrementar el costo computacional. Estas salidas se concatenan (Concat) para formar los bloques de salida [11]. La arquitectura ha sido refinada en varias versiones, incluyendo V2, V3 y V4. La Figura 5 presenta la arquitectura de bloques de InceptionV3.

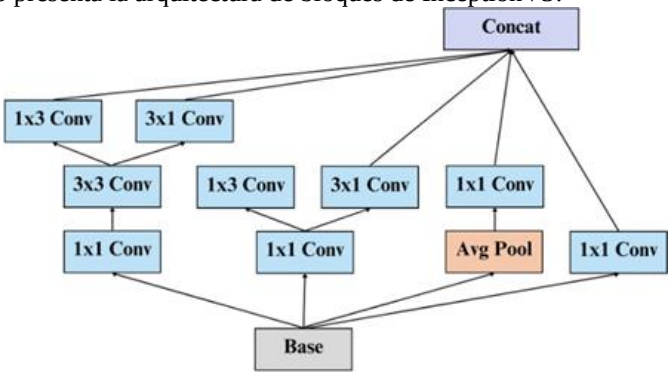

Figura 5. Estructura de un “bloque InceptionV3”.

3. DESARROLLO

3.1. Implementación de modelos.

Entre las librerías utilizadas en Python para el entrenamiento de los modelos se encuentra TensorFlow con Keras para importar las arquitecturas de VGG16, ResNet50 e InceptionV3 y que también permite crear, compilar y entrenar los modelos de la red neuronal. Las emociones se mapearon a valores numéricos (0-6) para su clasificación.

Luego, se accedió al directorio de la base de datos para leer cada imagen en escala de grises y se redimensionaron al tamaño que acepta la red a utilizar (224x224 pixeles para VGG16 y ResNet50, 299x299 pixeles para InceptionV3). A la vez que se cargó cada imagen, se extrajo del nombre de la imagen la etiqueta de la emoción correspondiente a esa expresión facial, la etiqueta se mapeo al valor numérico correspondiente a la emoción y se guardó en un arreglo.

Una vez que se obtuvo el arreglo de las 213 etiquetas e imágenes, estas se convirtieron en arreglos NumPy, el cual es el formato adecuado para el entrenamiento de los modelos con TensorFlow. Los datos se dividieron en 70% para el entrenamiento del modelo, 15% para la validación y 15% para pruebas. En la Figura 6 se muestra un diagrama con los pasos realizados.

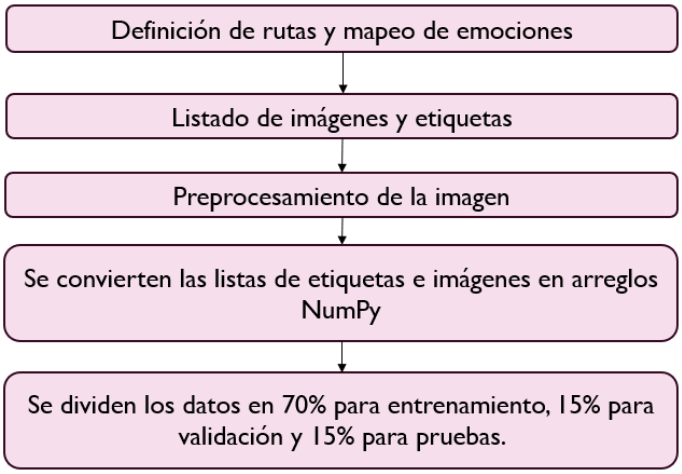

Figura 6. Diagrama para procesamiento de los datos.

Después se cargaron los modelos de VGG16, ResNet50 e InceptionV3 por medio de TensorFlow y Keras. Se utilizaron los pesos pre entrenados de las redes con ImageNet, sin embargo, se omitió la parte final del modelo original para adaptar el modelo a la nueva tarea. Las capas del modelo base se congelaron, es decir, no se entrenaron, para conservar el conocimiento aprendido durante el entrenamiento. Luego, se añadieron capas personalizadas que permiten al modelo aprender las características específicas del conjunto de datos JAFFE. En la Figura 7 se muestra esta configuración.

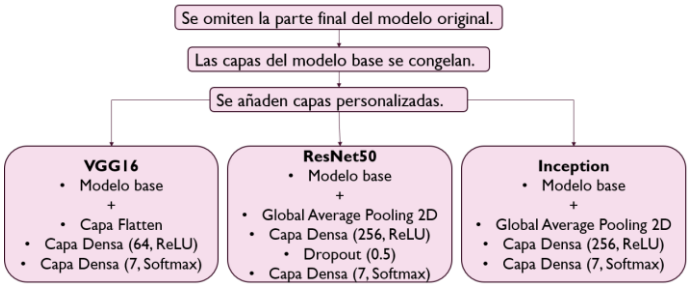

Figura 7. Diagrama para configuración de los modelos.

Durante el entrenamiento se utilizaron funciones para almacenar las pérdidas y tasas de aprendizaje correspondientes a cada época. Estas funciones consisten en monitorear la pérdida de entropía cruzada para determinar si el modelo ha mejorado y solo se guarda el modelo si la métrica ha mejorado con respecto a la mejor puntuación observada hasta ese momento.

3.2. Evaluación de modelos.

Se cargó el modelo guardado durante el entrenamiento que obtuvo el mejor rendimiento y se mide el tiempo de inferencia para calcular los FPS. El modelo realizó predicciones sobre las imágenes del conjunto de datos de prueba y se compararon las predicciones con las etiquetas reales de las imágenes para determinar si el modelo predijo correctamente.

Cabe destacar que, para realizar el entrenamiento de cada modelo, se generó un archivo para cada uno, y cada modelo se guardó en un formato H5 para TensorFlow. Hasta este punto solo se ha tratado sobre configuración, entrenamiento y validación de los modelos.

3.3. Cargar modelos.

En este estudio se optó por utilizar el formato ONNX (Open Neural Network Exchange) para simplificar la portabilidad de los modelos, dado que es un formato de intercambio abierto que permite a los desarrolladores usar el mismo modelo en diferentes frameworks y plataformas de aprendizaje automático. Para ejecutar los modelos en la tarjeta NVIDIA Jetson TX2 fue necesario convertir los modelos a formato ONNX. Una vez con los modelos en este formato se realizó la carga de cada modelo y se realizó el mismo procedimiento de las secciones anteriores: se accede al directorio de la base de datos de manera similar que en la sección 3.1 para cargar cada imagen, obtener la etiqueta de la emoción real, pre procesar la imagen y cargarla al modelo para realizar la predicción. Se midió el tiempo que tarda el modelo en clasificar todas las imágenes para calcular los FPS. Para medir el rendimiento de la GPU y uso de memoria se utilizó la librería de GPUtil para el entorno de la computadora Intel Xeon con GPU NVIDIA GeForce RTX 3060 y Tegrastats para la tarjeta Jetson TX2.

3.4. Configuración de Jetson TX2.

Se siguieron los pasos mencionados en [12] para la configuración de Jetson TX2.
Primero se instaló el sistema operativo de Ubuntu en la PC, para después pasar a conectar la tarjeta Jetson TX2 e instalar el SDK. El SDK para la tarjeta Jetson TX2 es un conjunto de herramientas y bibliotecas proporcionadas por NVIDIA diseñadas para facilitar el desarrollo y despliegue de aplicaciones de inteligencia artificial y aprendizaje automático en los dispositivos Jetson. Para correr los modelos en la tarjeta Jetson TX2 se tuvo que realizar un entorno virtual con una versión de Python superior a 3.8 e instalar las demás librerías, entre ellas onnxruntime, para ejecutar el archivo que carga el modelo ONNX.

4. RESULTADOS

4.1. Rendimiento en FPS.

La computadora Intel Xeon con GPU NVIDIA GeForce RTX 3060 presentó un rendimiento significativamente superior en términos de FPS para todos los modelos. Los resultados con fueron de 45.51 FPS para VGG16, 15.84 FPS para Inception y 21.35 FPS para ResNet50. En comparación, la tarjeta Jetson TX2 alcanzó FPS de 0.82, 0.91 y 1.26 para los modelos VGG16, Inception y ResNet50, respectivamente.

4.2. Procesamiento y uso de memoria.

El uso del formato ONNX permitió utilizar los modelos entrenados con TensorFlow y Keras en la computadora para su

ejecución en la tarjeta Jetson TX2. En la computadora Intel Xeon con GPU NVIDIA GeForce RTX 3060 la carga de GPU aumentó después de la inferencia y el consumo de memoria se mantuvo alrededor de los 11160MB de los 12288 MB disponibles. La tarjeta Jetson mostro una utilización limitada de la GPU, con la frecuencia GR3D manteniéndose en 0% durante las inferencias.

4.3. Comparación de arquitecturas.

VGG16: la arquitectura es más simple y menos profunda que ResNet50 e InceptionV3, sin embargo, mostró una notable diferencia en rendimiento de FPS entre ambos dispositivos.

ResNet50: la arquitectura es más compleja y profunda y aunque mostró una mejora marginal en la tarjeta Jetson TX2 en comparación con la computadora Intel Xeon con GPU NVIDIA GeForce RTX 3060 en cuestión de precisión, su desempeño fue significativamente inferior en términos de FPS.

InceptionV3: también es una arquitectura compleja que, aunque fue eficiente en la computadora Intel Xeon con GPU NVIDIA GeForce RTX 3060, en la tarjeta Jetson TX2 mostró un rendimiento inferior en comparación con VGG16.

5. CONCLUSIONES

En este artículo se presentó un análisis del costo computacional de las arquitecturas VGG16, ResNet50 e InceptionV3 utilizando la base de datos JAFFE. Se evaluó el rendimiento en los entornos de hardware Jetson TX2 y una computadora con procesador Intel Xeon E5-1603 v3 equipada con GPU NVIDIA GeForce RTX 3060 centrado en la velocidad de procesamiento (FPS) y consumo de memoria.

De acuerdo con los resultados, la computadora Intel Xeon con GPU NVIDIA GeForce RTX 3060 obtuvo un rendimiento superior en comparación con la tarjeta Jetson TX2, demostrando su capacidad para manejar modelos más grandes y complejos. Además, utilizó su GPU de manera eficiente para la inferencia. Sin embargo, presentó un alto consumo de memoria antes de la inferencia, lo que puede ser una limitación en entornos con recursos de memoria críticos.

Con los resultados obtenidos con la tarjeta Jetson TX2, se observó un menor consumo de recursos en comparación con la RTX 3060, lo cual es un beneficio para aplicaciones donde el consumo de energía y memoria es crítico. Sin embargo, la incapacidad de utilizar la GPU para la inferencia es una limitación.

Como trabajo futuro se buscará utilizar la GPU de la tarjeta Jetson TX2 de manera eficiente optimizando controladores y software para permitir la inferencia por GPU explorando técnicas para mejorar la velocidad de procesamiento sin comprometer la precisión.

En la Tabla 1 se muestran los datos cuantitativos de estos resultados.

Tabla 1. Resultados en la computadora Intel Xeon con GPU NVIDIA GeForce RTX 3060

Modelos	VGG16	ResNet50	InceptionV3
Precisión	81.25%	43.75%	71.88%
FPS	45.51	21.35	15.84
GPU antes inferencia	Load: 23%, GPU mem: 11160MB / 12288MB	Load: 45%, GPU mem: 11212MB / 12288MB	Load: 29%, GPU mem: 11237MB / 12288MB
GPU después inferencia	Load: 64%, GPU mem: 11160MB / 12288MB	Load: 100%, GPU mem: 11212MB / 12288MB	Load: 100%, GPU mem: 11205MB / 12288.00MB

Tabla 2. Resultados en Jetson TX2.

Modelos	VGG16	ResNet50	InceptionV3
Precisión	81%	44%	72%
FPS	0.82	1.26	0.91
GPU antes inferencia	RAM 2638MB / 7858MB, GR3D_FREQ 0%	RAM 2638MB / 7858MB, GR3D_FREQ 0%	RAM 2638MB / 7858MB, GR3D_FREQ 0%
GPU después inferencia	RAM 2671MB / 7858MB, GR3D_FREQ 0%	RAM 2693MB / 7858MB, GR3D_FREQ 0%	RAM 2693MB / 7858MB, GR3D_FREQ 0%

6. AGRADECIMIENTOS

Se agradece al Tecnológico Nacional de México por el financiamiento del proyecto “Modelo de reconocimiento de emociones para cuidado de salud mental basado en ciencia de datos y redes neuronales profundas” con número 19182.24-P.

7. REFERENCIAS

[1] S. K. Khare, V. Blanes-Vidal, E. S. Nadimi, and U. R. Acharya, “Emotion recognition and artificial intelligence: A systematic review (2014–2023) and research recommendations,” Information Fusion, vol. 102, p. 102019, Feb. 2024, doi: https://doi.org/10.1016/j.inffus.2023.102019.

[2] P. Ekman, R. W. Levenson, and W.V. Friesen, “Automatic nervous system activity distinguishes among emotions”, Science, vol. 221, no. 4616, pp. 1208-1210, Sep. 1983.

[3] P. Naga, S. D. Marri, and R. Borreo, “Facial emotion recognition methods, datasets and technologies: A literature survey,” Materials Today: Proceedings, Jul. 2021, doi: https://doi.org/10.1016/j.matpr.2021.07.046.

[4] Y. S. Can, B. Mahesh, E. André, “Approaches, applications, and challenges in physiological emotion recognition—a tutorial overview,” Proceedings of the IEEE, pp. 1-27, 2023

[5] N. Bukhari, S. Hussain, M. Ayoub, Y. yu, and A. Khan, “A Deep Learning-based Framework for Emotion Recognition using Facial Expression”, Pakistan Journal of Engineering and Technology, vol. 5, no. 3, pp. 51-57, 2022

[6] M. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, “Coding facial expressions with Gabor wavelets”, Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 200-205, 1998.

[7] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Computer Science, 2014, doi: https://doi.org/10.48550/arXiv.1409.1556

[8] S. Bangar, “VGG-Net Architecture Explained,” Medium, Jun. 28, 2022. [Online]. Available: https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” arXiv (Cornell University), Dec. 2015, doi: https://doi.org/10.48550/arxiv.1512.03385

[10] Nayan Chaure, “Variants of ResNet: A Comparative Analysis - Nayan Chaure - Medium,” Medium, Apr. 27, 2024. https://medium.com/@nayanchaure601/variants-of-resnet-a-comparative-analysis-63fdc1573b34

[11] C. Szegedy et al., “Going Deeper with Convolutions,” arXiv (Cornell University), Sep. 2014, doi: https://doi.org/10.48550/arxiv.1409.4842

[12] J.J. Muñoz, “Manual de usuario y herramientas para introducción al uso y desarrollo de redes neuronales en el sistema embebido NVIDIA Jetson TX2”, Instituto Tecnológico de Chihuahua, pp. 1-28, 2024