

IMPLEMENTACIÓN DE UN ALGORITMO DE RECONOCIMIENTO FACIAL DE EMOCIONES UTILIZANDO LA TARJETA JETSON NANO DE NVIDIA PARA REDES NEURONALES ARTIFICIALES

Chávez Guzmán Alexis Daniel ¹, Espinoza Barraza Leticia Sarahi ¹, López Ramírez Miguel Ángel ¹, Mora Rosas Norma Elizabeth ²

Tecnológico Nacional de México ¹

Instituto Tecnológico de Tijuana ¹

Departamento de Sistemas y Computación ¹

Calzada Del Tecnológico S/N, Fraccionamiento Tomas Aquino. Tijuana, Baja California. C.P. 22414 ¹

SportMed Av. Patria 1401 Zapopan, Jalisco, México C.P. 45136 ²

+52 (664) 607 8400 ¹

+52 (33) 3656 6999 ²

alexis.chavez17@tectijuana.edu.mx ¹

leticia.espinoza16@tectijuana.edu.mx ¹

mlopez@tectijuana.edu.mx ¹

moraras@gmail.com ²

RESUMEN.

El presente trabajo consiste en determinar las emociones por medio de las expresiones faciales de pacientes en terapia de rehabilitación deportiva, esto, mediante un dispositivo previamente entrenado con un banco de imágenes de rostros de diferentes personas del mundo, siendo capaz de reconocer en tiempo real aquellas emociones presentadas en pacientes en terapia. Las emociones, específicamente las expresiones faciales, muestran la pauta para lograr el avance en la recuperación del deportista al realizar sus terapias de rehabilitación de forma que el dispositivo capte el estado emocional, de llegar a ser una emoción negativa, se neutralice por medio de un terapeuta. Se utilizaron dos dispositivos uno de ellos Movidius y el otro Jetson Nano que son comparados para determinar cuál de ellos tiene mayor precisión y es más apto para lograr el propósito deseado. Por último, se presenta una comparativa y conclusiones de la investigación presentada.

Palabras Clave: Reconocimiento facial, emociones, expresiones faciales, terapias de rehabilitación, redes neuronales, procesamiento digital de imágenes.

ABSTRACT.

The present work consists of determining the emotions by means of the facial expressions of the patient in sports rehabilitation therapy, this, by means of a device previously trained with a bank of images of faces of different people of the world, being able to recognize in real time those emotions presented in patients in therapy. Emotions specifically facial expressions show the pattern to achieve progress in the recovery of the athlete to perform their rehabilitation therapies so that the device captures the emotional state, to become a negative emotion, neutralized by a therapist. Two devices were used, one of them Movidius and the other Jetson Nano that are compared to determine which one has greater precision and is more apt to achieve the desired purpose. Finally, a comparison and conclusions of the presented research is presented.

Keywords: Facial recognition, emotions, facial expressions, rehabilitation therapies, neural networks, digital image processing.

1. INTRODUCCIÓN

Los gestos son una de las formas de comunicación que tiene el ser humano, mediante estos, pueden llegar a expresarse cientos de mensajes. El rostro puede realizar varias expresiones de acuerdo a la emoción que se esté sintiendo en cierta situación. Las emociones están estrechamente relacionadas con las expresiones que realizamos y esto se convierte en un determinante para conocer al ser humano sobre su estado de ánimo. El estado de ánimo influye de cierta manera con nuestra salud y más aún cuando hablamos de recuperación por medio de la terapia de rehabilitación de un paciente, se pretende que cada paciente ingrese a su terapia de rehabilitación con el mejor estado de ánimo posible para que en conjunto se logre en un avance significativo, aunque esto no es lo que siempre sucede ya que esto depende de las circunstancias bajo las cuales se encuentre el paciente. Tomando en cuenta estos datos se ha decidido realizar la implementación de un algoritmo en un dispositivo que sea capaz de detectar aquellas emociones que no favorezcan al paciente en su terapia de rehabilitación y al ser detectadas que una persona capacitada sea capaz de neutralizar y así canalizarlo de nuevo a su terapia de rehabilitación o agendar una nueva cita.

2. TRABAJOS RELACIONADOS

Como lo muestra el siguiente trabajo se pretende tener como objetivo la implementación de un dispositivo de redes neuronales en una Raspberry Pi 3 para la detección de objetos específicos. Se utilizó un dispositivo de redes neuronales artificiales llamado Movidius con un modelo ya entrenado para detección de objetos [1]. A partir de estos puntos se detectan los movimientos musculares de la cara (unidades de acción) y en función de ellos un clasificador reconoce la emoción. Dicho clasificador es entrenado previamente a partir de las combinaciones más frecuentes de unidades de acción en cada una de las emociones básicas (ira, asco, miedo, felicidad, tristeza

y sorpresa) [2]. El objetivo de la presente investigación es analizar la relación existente entre Inteligencia Emocional y Motivación de Logro en deportistas de alto rendimiento y si ello influye en el desempeño del deportista [3]. Las emociones juegan un rol muy importante en la vida de cada sujeto y podemos conocer cada una de ellas por medio de las expresiones faciales [4].

3. HERRAMIENTAS DE TRABAJO

Para la implementación del reconocimiento facial de emociones se usará un algoritmo llamado GoogLeNet¹ y VGGNet² en conjunto con algunas herramientas de software descritas a continuación:

El Algoritmo usado para la red neuronal fue GoogLeNet en su versión ILSVRC³ que fue una competencia de redes neuronales y su rendimiento. La configuración para el entrenamiento fue tomar 20 fotos, analizar, entrenar y volver a repetir el ciclo, con un ratio de aprendizaje 0.001 por vuelta, con 85000 iteraciones o “vueltas” que da a las fotos, con el tipo de caída de aprendizaje de “step” el cual multiplica este ratio de aprendizaje por la cantidad de fotos elevado a la iteración en la que está, para finalizar se trabajó mediante la Tarjeta Gráfica del sistema (GTX 1060).

3.1. Caffe

Caffe es un marco de aprendizaje profundo creado teniendo en cuenta la expresión, la velocidad y el modularidad. Es desarrollado por Berkeley AI Research (BAIR) y por colaboradores de la comunidad. Yangqing Jia creó el proyecto durante su doctorado en UC Berkeley. Caffe se lanza bajo la licencia BSD 2-Clause [5].

Se utilizará para realizar el entrenamiento de la red neuronal mediante capas que serán las que leerán las imágenes para después predecir de que se trata lo que está viendo.

Caffe es el software con el que se creó el modelo de red neuronal capaz de mostrar las predicciones a las diversas emociones y expresiones aprendidas.

3.2. Nvidia Jetson Nano

NVIDIA® Jetson Nano™ Developer Kit es una computadora pequeña y poderosa que permite ejecutar múltiples redes neuronales en paralelo para aplicaciones como clasificación de imágenes, detección de objetos, segmentación y procesamiento de voz. Todo en una plataforma fácil de usar que funciona en tan solo cinco vatios [6].

El mismo JetPack SDK se utiliza en toda la familia de productos NVIDIA Jetson™. Además, es compatible con la plataforma de IA líder en el mundo de NVIDIA para la capacitación y el despliegue del software de AI, lo que reduce la complejidad y el esfuerzo para los desarrolladores.

Jetson Nano, es como si se tuviera una Raspberry en conjunto con Movidius todo en uno, con la desventaja de la memoria RAM (4 Gigabytes) y algunos problemas de calentamiento.

3.3. Movidius

Es una herramienta de desarrollo para inferencia de aprendizaje profundo con un consumo mínimo. La herramienta permite el desarrollo y prototipado de aplicaciones de inteligencia artificial (IA) en una amplia gama de dispositivos de última generación con un cómodo formato USB [7]. Las VPU⁴ de Intel® Movidius™ manejan las cargas de trabajo exigentes de la visión moderna de la computadora y las aplicaciones de inteligencia artificial a una potencia ultra baja. Al acoplar la computación programable altamente paralela con la aceleración de hardware específica de la carga de trabajo y la ubicación conjunta de estos componentes en un tejido de memoria inteligente común, Movidius logra un equilibrio único de eficiencia energética y alto rendimiento [8]. Movidius logra cargas bastante decentes con la negativa de que tiene varias limitaciones como el de solo poder cargar modelos de hasta 320 Megabytes.

3.4. Keras

Keras es una API⁵ de redes neuronales de alto nivel, escrita en Python y capaz de ejecutarse sobre TensorFlow, CNTK o Theano. Fue desarrollado con un enfoque en permitir la experimentación rápida. Poder pasar de la idea al resultado con el menor retraso posible es clave para hacer una buena investigación.

Keras es una biblioteca de aprendizaje profundo que:

- Permite una creación de prototipos fácil y rápida (a través de la facilidad de uso, el modularidad y la extensibilidad).
- Admite redes convolucionales y redes recurrentes, así como combinaciones de las dos.
- Funciona a la perfección en CPU y GPU [9] [10].

Keras es otro creador de modelos que trabaja en colaboración de Movidius.

3.5. OpenCV

Open Source Computer Vision Library, es una biblioteca de software de visión de computadora y de aprendizaje automático

¹ GoogLeNet La red utilizó una red neuronal convolucional (CNN) inspirada en LeNet, pero implementó un elemento novedoso que se denomina módulo de inicio. Se utilizó la normalización por lotes, distorsiones de imagen y RMSprop.

² VGGNet consta de 16 capas convolucionales y es muy atractiva debido a su arquitectura muy uniforme.

³ Imagenet large scale visual recognition competition

⁴ Unidad de procesamiento de visión (VPU) es un tipo de microprocesador destinado a acelerar el aprendizaje automático y las tecnologías de inteligencia artificial.

⁵ Application Programming Interface

de código abierto, construido para proporcionar infraestructura común para las aplicaciones de visión artificial y para acelerar el uso de la percepción de la maquina en los productos comerciales siendo este de licencia BSD [11].

OpenCV es utilizado para la creación de la base de datos de las imágenes con las cuales se entrenó el modelo de red neuronal.

3.6. CUDA

Es una plataforma de computación paralela y un modelo de programación desarrollado por NVIDIA para computación general en unidades de procesamiento grafico GPU: con CUDA, los desarrolladores pueden acelerar drásticamente las aplicaciones informáticas aprovechando el poder de las GPU [12].

Estos son los procesadores usados para entrenar al modelo de la red neuronal, modelo que es funcional solo con la tarjeta gráfica Nvidia.

3.7. TensorFlow

Es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral flexible de herramientas, bibliotecas y recursos de la comunidad que permite a los investigadores impulsar el estado del arte en ML⁶ y los desarrolladores puedan desplegar fácilmente aplicaciones potenciadas por ML [13].

TensorFlow es utilizado para crear la división de las emociones para que sean reconocidas por medio de Caffe y así aprenda a distinguirlas.

4. IMPLEMENTACIÓN DE MODELO EN DISPOSITIVOS

El objetivo principal es tener una respuesta con un buen porcentaje en aciertos de cuál es la emoción que está sintiendo el paciente en terapia de rehabilitación por medio de sus expresiones faciales para esto se utilizó un modelo pre entrenado con objetos y que de esta manera confirme la instalación y correcto funcionamiento del dispositivo en este caso una Raspberry Pi 3+ con Movidius, OpenCV y NVIDIA Jetson Nano [14].

El Algoritmo utilizado para la realización de la red neuronal fue GoogLeNet en su versión de la ILSVRC que fue una competición de redes neuronales y su rendimiento. La configuración para el entrenamiento fue que toma 20 fotos las analiza, entrena y vuelve a repetir el ciclo, con una ratio de aprendizaje 0.001 por vuelta, con 85000 iteraciones o “vueltas” que le da a las fotos, con el tipo de caída de aprendizaje de “step” el cual multiplica este ratio de aprendizaje por la cantidad de fotos elevado a la iteración en la que está y para finalizar todo esto se trabajó mediante la Tarjeta Gráfica del sistema (GTX 1060).

4.1. Movidius

El modelo basado en Keras se pre entrenó con objetos de la vida cotidiana con fines de prueba [15] [16] [17] [18] . En la Figura 1 se puede apreciar la detección de un objeto en este caso una silla de escritorio con una coincidencia del 100%, esto mediante el entrenamiento con Movidius. En la Figura 2 se muestra la detección al 100% de un perro.

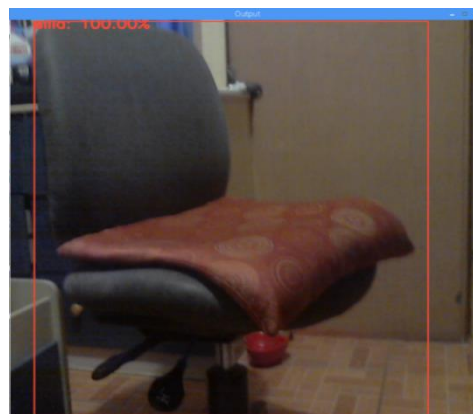


Figura 1. Detección de una silla al 100% con Movidius

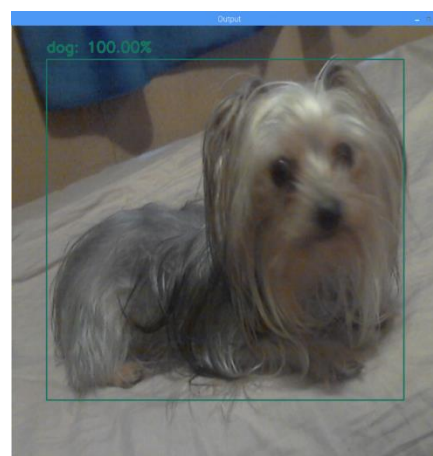


Figura 2. Detección de un perro al 100% con Movidius

Las predicciones del modelo pre entrenado están realizadas en el software de Caffe hecho por Google con un entrenamiento de 18,000 imágenes de diferentes objetos para tener una presión del 95%, con estos datos se obtuvo el resultado de cinco fotogramas por segundo como se muestra en la Figura 3.

⁶ Machine Learning

```

94, 265))
[INFO] Prediction #1: class=tvmonitor, confidence=0.568359375, boxpoints=((13, 3
6), (241, 297))
[INFO] Prediction #0: class=dog, confidence=0.998046875, boxpoints=((10, 48), (1
96, 274))
[INFO] Prediction #0: class=dog, confidence=0.998046875, boxpoints=((6, 46), (19
4, 268))
[INFO] Prediction #0: class=dog, confidence=0.9990234375, boxpoints=((8, 47), (1
95, 269))
[INFO] Prediction #1: class=tvmonitor, confidence=0.52880859375, boxpoints=((9,
18), (285, 297))
[INFO] Prediction #0: class=dog, confidence=0.9990234375, boxpoints=((9, 47), (1
95, 269))
[INFO] Prediction #0: class=dog, confidence=0.9990234375, boxpoints=((4, 42), (1
89, 264))
[INFO] Prediction #0: class=dog, confidence=1.0, boxpoints=((4, 46), (184, 265))
[INFO] Prediction #1: class=tvmonitor, confidence=0.71044921875, boxpoints=((0,
16), (236, 294))
[INFO] Prediction #0: class=dog, confidence=1.0, boxpoints=((0, 40), (175, 260))
[INFO] Prediction #0: class=dog, confidence=1.0, boxpoints=((0, 40), (166, 253))
[INFO] Prediction #0: class=dog, confidence=1.0, boxpoints=((0, 40), (155, 251))
^C[INFO] elapsed time: 36.18
[INFO] approx. FPS: 5.33
pi@raspberrypi:~/prueba/google/realtime-object-detection $
    
```

Figura 3. Entrenamiento mostrando fotogramas por segundo

4.2. Nvidia Jetson Nano

Nuevamente se utiliza al objeto silla y al objeto perro para realizar una prueba, esta vez en con NVIDIA Jetson Nano.

En la Figura 4 se muestra la detección y coincidencia que se encontró utilizando Jetson Nano mientras en la Figura 5 se encuentra el resultado de la coincidencia del perro.



Figura 4. Detección de una silla al 94.53% con NVIDIA Jetson Nano

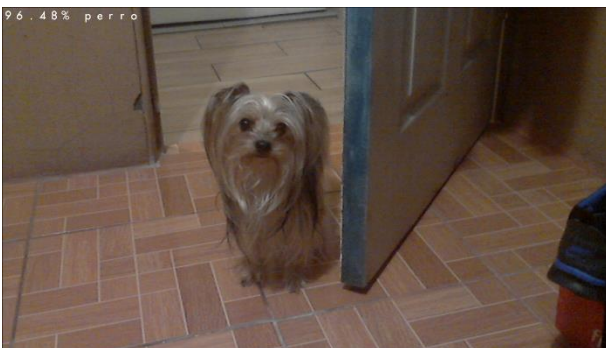


Figura 5. Detección de un perro al 96.48% con NVIDIA Jetson Nano

El modelo realizado con Jetson Nano tiene problema alguno detectando al objeto perro ni al objeto silla.

Un dato interesante es que Jetson Nano responde con el doble de velocidad en FPS⁷ que Movidius 10 FPS.

4.3. Comparativa de dispositivos

A continuación, se muestra la comparativa entre Movidius y Jetson Nano, una de sus diferencias se entra en los Fotogramas por segundo. En la tabla 1 se pueden observar los datos de FPS.

Tabla 1. Comparación de precisión en FPS de dos objetos entre las tarjetas de Movidius y Jetson Nano.

Tarjeta	Fotogramas Por Segundo	Objeto	
		Silla de Escritorio	Perro
Movidius	5.6 FPS	100%	100%
Jetson Nano	10.1 FPS	94.53%	96.48%

NVIDIA Jetson Nano tiene el doble de fotogramas por segundo, procesa muchas más veces la misma imagen arrojando una precisión más exacta que Movidius, este tiene la mitad de fotogramas por segundo por lo cual arroja una precisión que simula que no hay error alguno.

5. ENTRENAMIENTO

Para alcanzar el objetivo el entrenamiento [19] [20] , se realizó con 1596 imágenes de rostros de todo el mundo a fin de que se logre un mejor reconocimiento facial y se arroje un porcentaje más favorable, los rostros de las imágenes contienen siete emociones las cuales son: “Enojo”, “Disgusto”, “Miedo”, “Felicidad”, “Neutral”, “Triste” y “Sorprendido”. Las imágenes van de un tamaño de 224 por 224 pixeles [21].

Esto realizado en una computadora diseñada para trabajos pesados con:

- Procesador Intel i7 4790 a 4.00Ghz
- Tarjeta gráfica Nvidia GTX 1060 de 6Gb con 1280 procesadores CUDA
- Memoria RAM de 16 GB

Como se puede apreciar en la Figura 6 se muestra una corrida del entrenamiento en consola que indica que se están haciendo comprobaciones en la red neuronal para después comenzar con el aprendizaje [22].

⁷ FPS Fotogramas Por Segundo


```

servidor@servidor-Z97X-SLI: ~/Downloads/CFEE
File Edit View Search Terminal Help
I0603 15:11:19.820896 2399 net.cpp:198] conv1 needs backward computation.
I0603 15:11:19.820894 2399 net.cpp:200] label_data_1_split does not need backwa
rd computation.
I0603 15:11:19.820896 2399 net.cpp:200] data does not need backward computation
.
I0603 15:11:19.820899 2399 net.cpp:242] This network produces output accuracy
I0603 15:11:19.820900 2399 net.cpp:242] This network produces output loss
I0603 15:11:19.820910 2399 net.cpp:255] Network initialization done.
I0603 15:11:19.820966 2399 solver.cpp:56] Solver scaffolding done.
I0603 15:11:19.820991 2399 caffe.cpp:248] Starting Optimization
I0603 15:11:19.820996 2399 solver.cpp:272] Solving caffeNet
I0603 15:11:19.820997 2399 solver.cpp:273] Learning Rate Policy: step
I0603 15:11:19.876502 2399 solver.cpp:330] Iteration 0, Testing net (#0)
I0603 15:15:52.480181 2403 data_layer.cpp:73] Restarting data prefetching from
start.
I0603 15:21:55.497507 2403 data_layer.cpp:73] Restarting data prefetching from
start.
I0603 15:27:36.569077 2403 data_layer.cpp:73] Restarting data prefetching from
start.
I0603 15:33:40.296363 2403 data_layer.cpp:73] Restarting data prefetching from
start.
I0603 15:39:43.503557 2403 data_layer.cpp:73] Restarting data prefetching from
start.
    
```

Figura 6. Muestra del inicio de aprendizaje del entrenamiento.

El porcentaje de precisión que tiene cada dispositivo se debe a diversos factores uno es la cantidad de imágenes entre mayor número de imágenes cuente el entrenamiento mayor será la precisión, agregando que la imagen sea de personas con diferentes rasgos del mundo. Otro factor importante son los fotogramas por segundo como lo mostro la Tabla 1.

6. DETECCIÓN DE EMOCIONES EN TIEMPO REAL

Una vez realizado el entrenamiento se obtuvo una precisión de 75% estimada, esta precisión permitió obtener resultados sorprendentemente buenos.

7. PRUEBAS

Las pruebas realizadas son la base para conocer el rendimiento que tiene el modelo y así se tenga una certeza de que los resultados obtenidos son confiables para que se pueda tomar una decisión sobre si neutralizar al paciente a recibir su terapia de rehabilitación o proceder con la terapia. Lo importante es ayudar a que los deportistas recuperen su condición en el mejor tiempo posible cuidando su salud física y mental.

A continuación, se mostrarán imágenes pasadas por el modelo entrenado Jetson Nano para el cual se realizaron bajas las siete emociones mencionadas anteriormente.

En la Figura 7 se muestra la emoción neutra en la que el modelo Jetson Nano realizo una detección del 99.98%. En figura 8 se apreciar la emoción del enojo siendo detectada con un 66.07% y la Figura 9 muestra un porcentaje del 97.83 para la emoción de Disgusto.

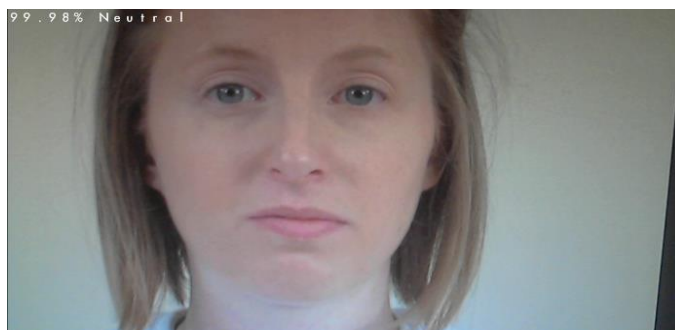


Figura 7. Emoción Neutral siendo detectado por el modelo Jetson Nano en un 99.98%

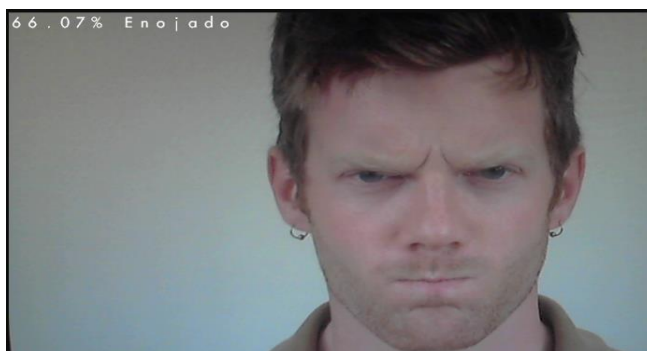


Figura 8. Emoción Enojo siendo detectado por el modelo Jetson Nano en un 66.07%

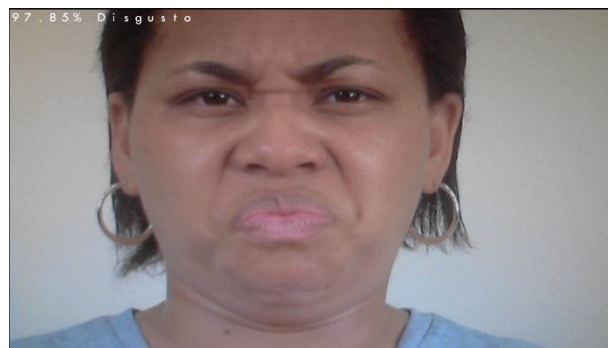


Figura 9. Emoción Disgusto siendo detectado por el modelo Jetson Nano en un 97.85%

Tabla 2. Resultados de pruebas realizadas con el banco de datos del entrenamiento.

Emoción	Imágenes de Entrenamiento	Imágenes de Prueba	Promedio de Precisión
Disgusto	230	5	91.246%
Enojo	230	5	75.536%
Miedo	230	5	98.59%
Feliz	230	5	97.99%
Neutral	230	5	94.02%
Sorprendido	230	5	78.664%
Triste	230	5	76.06%
		\bar{X}	87.44%

Al modelo utilizado con Jetson Nano se le aplicaron varias pruebas para determinar la precisión en la detección de expresiones, una vez que el modelo ya tenía un avance en su aprendizaje se procedió a realizar las pruebas a las siete emociones con cinco muestras por cada emoción. En la Tabla 2 se podrá observar el resultado de las pruebas realizadas mostrando el promedio de las muestras por emoción.

Tal como se observa en la Tabla 2, las emociones no responden con el mismo porcentaje de coincidencia esto porque alguna de las imágenes a las que se le hizo las pruebas marcaban más sus expresiones que otras y esto también debió a que el modelo aún tiene que aprender, entre más tiempo pase entrenándose mejor precisión tendrá a la hora de querer identificar una emoción.

8. CONCLUSIONES

Los datos que se han obtenido en la investigación han sido satisfactorios para la primera fase que fue la de realizar el entrenamiento y pruebas con imágenes de rostros que muestran alguna expresión, en la segunda, se desea que el modelo se utilice directamente en los deportistas en terapia de rehabilitación.

El entrenamiento realizado en cada uno de los dispositivos nos arrojó importantes datos, hablando de Movidius fue una guía para manejar las redes neuronales e implementarla en el modelo, así obtuvimos resultados como el de que su entrenamiento es más preciso cuando se trata de objetos, sea a larga o corta distancia mientras que en el dispositivo Jetson Nano, la detección de objetos y de emociones tiene un porcentaje de detección mayor y este nos agregó una amplia visión en el campo de la inteligencia artificial y el aprendizaje en un entorno neural. Otro dato a resaltar es que NVIDIA Jetson Nano comparado con Movidius (en conjunto con la Raspberry Pi 3+) tiene mejor capacidad de cargar modelos más pesados y de dar buenos fotogramas por segundo permitiendo al mismo tiempo una visualización más rápida de lo que muestra la cámara.

Se demostró que lo implementado en este documento puede ser capaz de ayudar a detectar las emociones por medio de las expresiones faciales de rostros, esto comprobado con una media de 87.44% en precisión que tiene el modelo utilizado con el dispositivo Jetson Nano.

8.1. Referencias.

- [1] A. B. R. Rojas, «Implementación de un dispositivo de redes neuronales en una Raspberry Pi 3,» Instituto Tecnológico de Tijuana, Tijuana, B. C., 2018.
- [2] D. Z. Olmedillo, «Herramienta de reconocimiento facial de emociones en Android,» E.T.S.I. y Sistemas de Telecomunicación (UPM), Madrid, 2016.
- [3] A. M. A. Melgarejo, «Relación entre inteligencia emocional y motivación de logro en deportistas de alto rendimiento,» Universidad Argentina de la Empresa, Ciudad Autónoma de Buenos Aires, 2015.
- [4] L. R. Suárez, «La influencia de las emociones en las expresiones faciales según Paul Ekman,» *Contribuciones a las Ciencias Sociales*, p. 7, 2016.
- [5] Y. a. S. E. a. D. J. a. K. S. a. L. J. a. G. R. a. G. S. a. D. T. Jia, «Caffe: Convolutional architecture for fast feature embedding,» *Proceedings of the 22nd ACM international conference on Multimedia*, pp. 675-678, 2014.
- [6] A. Rosebrock, «PyImageSearch,» Getting started with the NVIDIA Jetson Nano, 6 Mayo 2109. [En línea]. Available: <https://www.pyimagesearch.com/2019/05/06/getting-started-with-the-nvidia-jetson-nano/>. [Último acceso: 29 mayo 2019].
- [7] Interempresas Media, S.L.U. - Grupo Nova Àgora. «Interempresas net,» Dispositivo de computación neuronal: en formato USB, 2019. [En línea]. Available: [https://www.interempresas.net/Agricola/FeriaVirtual/Producto-Dispositivo-de-computacion-neuronal-Intel-Movidius-Neural-Compute-Stick-\(NCS\)-160023.html](https://www.interempresas.net/Agricola/FeriaVirtual/Producto-Dispositivo-de-computacion-neuronal-Intel-Movidius-Neural-Compute-Stick-(NCS)-160023.html). [Último acceso: mayo 2029].
- [8] A. Rosebrock, «Real-time object detection on the Raspberry Pi with the Movidius NCS,» PyImageSearch, 19 Febrero 2019. [En línea]. Available: <https://www.pyimagesearch.com/2018/02/19/real-time-object-detection-on-the-raspberry-pi-with-the-movidius-ncs/>. [Último acceso: 20 Marzo 2019].
- [9] Keras Documentation, «Keras Documentation,» 2019. [En línea]. Available: <https://keras.io/>. [Último acceso: 21 mayo 2019].
- [10] A. Rosebrock, «PyImageSearch,» Keras – Save and Load Your Deep Learning Models, 10 Diciembre 2018. [En línea]. Available: <https://www.pyimagesearch.com/2018/12/10/keras-save-and-load-your-deep-learning-models/>. [Último acceso: 29 Mayo 2019].
- [11] OpenCV team, «OpenCV,» 2019. [En línea]. Available: <https://opencv.org/>. [Último acceso: 13 mayo 2019].
- [12] NVIDIA Corporation, «NVIDIA Developer,» 2019. [En línea]. Available: <https://developer.nvidia.com/cuda-zone>. [Último acceso: 18 mayo 2019].
- [13] TensorFlor, «TensorFlor,» 2019. [En línea]. Available: <https://www.tensorflow.org>. [Último acceso: 30 mayo 2019].
- [14] G. Levi, «Github,» 16 Junio 2016. [En línea]. Available: <https://gist.github.com/GilLevi/54aee1b8b0397721aa4b>. [Último acceso: 24 abril 2019].
- [15] A. Rosebrock, «PyImageSearch,» Raspberry Pi: Deep learning object detection with OpenCV, 16 Octubre 2017. [En línea]. Available: <https://www.pyimagesearch.com/2017/10/16/raspberry-pi-deep-learning-object-detection-with-opencv/>. [Último acceso: 5 abril 2019].
- [16] A. Rosebrock, «PyImageSearch,» Getting started with the Intel Movidius Neural Compute Stick, 12 FEBRERO 2018. [En línea]. Available: <https://www.pyimagesearch.com/2018/02/12/getting-started-with-the-intel-movidius-neural-compute-stick/>. [Último acceso: 9 abril 2019].
- [17] A. Rosebrock, «PyImageSearch,» Keras: Multiple outputs and multiple losses, 4 junio 2018. [En línea]. Available: <https://www.pyimagesearch.com/2018/06/04/keras-multiple-outputs-and-multiple-losses/>. [Último acceso: 11 abril 2019].
- [18] A. Rosebrock, «PyImageSearch,» OpenVINO, OpenCV, and Movidius NCS on the Raspberry Pi, 8 abril 2019. [En línea]. Available: <https://www.pyimagesearch.com/2019/04/08/openvino-opencv-and-movidius-ncs-on-the-raspberry-pi/>. [Último acceso: 2 abril 2019].
- [19] D. C-H-D, «GitHub,» 8 julio 2018. [En línea]. Available: https://github.com/C-H-D/tensorRT-Caffe?fbclid=IwAR1BBDvbyYBecbkN_NqfSiqEuE_77_Ky6NNiZR8j2KpyxSPfz2QwLsLoT5Tc. [Último acceso: 23 mayo 2019].
- [20] J. Long, «JK Jung's blog,» How to Do Real-time Object Detection with SSD on Jetson TX2, 30 julio 2018. [En línea]. Available: https://jkjung-avt.github.io/camera-ssd-threaded/?fbclid=IwAR1oddgOb5ndlmInhhDo71RoKEaux7XWdxc_w2sbCB43AyntkSpqqtRwe8I. [Último acceso: 29 mayo 2019].
- [21] J. Herrera, «Github,» 21 Noviembre 2018. [En línea]. Available: <https://github.com/jorgher/faceRec>. [Último acceso: 27 abril 2019].
- [22] A. Kumar, «Machine Learning in Action,» Demonstration of Facial Emotion Recognition on Real Time Video Using CNN : Python & Keras, 28 Noviembre 2018. [En línea]. Available: <https://appliedmachinelearning.blog/2018/11/28/demonstration-of-facial-emotion-recognition-on-real-time-video-using-cnn-python-keras/>. [Último acceso: 7 mayo 2019]