Universidad Rafael Landivar

Inteligencia Artifical

Sección 1

Ing. Mario Rolando Valdez Argueta

Proyecto II

Diego Oswaldo Orellana Morales – 1163722

Kenny Uziel Sical Menjivar – 1273922

Diego Andres Azurdia Ortiz – 2528119

Cesar Daniel Bocel Morales – 1094921

Guatemala, 19 de mayo de 2025

Link al repositorio

Link: <https://github.com/diegoazurdia1998/Proyecto2IA2025.git>

Introducción al problema

El actual proyecto para la clase de Inteligencia Artificial de la carrera ingeniería en sistemas plantea la utilización de diferentes datasets, siendo uno de ellos del de rasgos faciales en personas, el objetivo de este proyecto es el crear un modelo de inteligencia artificial capaz de procesar, interpretar y modelarse a partir de los conocimientos que posee debido a este dataset y de esa manera poder desarrollarse.

Para este proyecto se utilizarán diversos conceptos estadísticos y fundamentos, como el de visión artificial o el de visión por computador, por el cual los datos de entrada van a ser pre procesados, predichos y últimamente determinando el tipo de emoción que se está expresando en las fotografías en cuestión.

El desarrollo de dicha herramienta está distribuido a través de diversos archivos .py como el archivo main, predictor y preprocess. En donde cuyos parámetros que habrán sido recibidos por medio de una página web serán procesados en tiempo real para dar como resultado la emoción que está expresando el individuo en cuestión al ser grabado.

Planteamiento del problema

En el día a día el ser humano experimenta diversa cantidad de emociones, siendo una de sus formas más constantes de expresarse, los rasgos faciales, los cuales son elementales para el entendimiento humano, sin embargo. Esto se debe a que los humanos son seres sensibles, al contrario de las máquinas, por lo que puede llegar a ser un tanto confuso para ellas la identificación correcta de dichos rasgos.

Este tipo de discrepancias por parte de las máquinas puede llegar a causar problemas al algunas de estas llegar a ser utilizadas para el identificar el estado de los empleados o incluso identificar ciertos rasgos físicos que puedan llegar a ser difícil de identificar para los seres humanos, y grandes empresas pueden llegar a tener una tarea abrumadora a la hora de identificar ese tipo de emociones en los empleados para poder capacitarlos en la forma correcta, por lo que puede ser una herramienta útil para mantener la salud mental de los empleados e identificar cuales requieren asistencia.

Objetivos generales

Desarrollar una herramienta la cual no solamente sea capaz de identificar la expresión de una foto entrante, sino que también de hacerlo en una grabación a tiempo real, mediante el uso del uso de la visión artificial.

Objetivos Específicos

La aplicación apropiada del modelo de red neuronal de forma apropiada para la detección de emociones.

Desarrollar una manera de poder pre procesar y predecir distintos datos entrantes en base al dataset a disposición.

Implementar una manera en que no solo mediante fotografías, sino que también a través de forma dinámica se detecte la emoción del individuo.

Descripción del dataset

El dataset está compuesto por diversas fotografías de personas clasificadas por la expresión que están mostrando, están categorizadas por medio de: **angry, disgust, fear, happy, neutral, sad, surprise.**

Siendo cada imagen de un tamaño de 48 x 48 píxeles con el objetivo de un mucho más sencillo procesamiento a la hora de evaluar y comparar las distintas emociones de manera mucho más apropiada y eficiente, especialmente al momento de realizarse de forma en vivo.

La distribución de datos a lo largo de las categorías es de:  
**Angry:** 958 imágenes con gente mostrando emoción de enojo.

**Disgust:** 111 imágenes con gente mostrando emoción de disgusto.

**Fear:** 1024 imágenes con gente mostrando emoción de miedo.

**Happy:** 1774 imágenes con gente mostrando emoción de alegría.

**Neutral:** 1233 imágenes de gente mostrando una expresión neutra.

**Sad:** 1247 imágenes de gente mostrando expresión de tristeza

**Surprise:** 831 imágenes de gente mostrando expresión de sorprendidos.

Con ello sumando un total de 7178 imágenes dentro del set de datos, en donde no solo se encuentran expresiones de personas reales, sino que también dibujos, siendo capaz de esta forma reconocer incluso expresiones de dibujos que tengan un tipo de semblanza con los humanos.

Explicación del preprocesamiento

La tarea de preprocesamiento es realizada por el archivo nombrado “preprocess.py”, lo primero que realiza es cargar la imagen ingresada la cual fue guardada dentro del proyecto luego de ser recibida, luego procede a redimensionar la imagen en cuestión a un 48x48 píxeles, lo cual hace que sea compatible a la comparación y predicción de las demás imágenes dentro del set de datos.

Luego de ello procede a normalizar los píxeles, volviéndolos a float32, y son escalados entre 0.0 y 1.0 dividiendo entre 255 para estabilizar el entrenamiento, luego de ello se vuelve la imagen en un formato de escala de grises con el objetivo de que sea mucho más sencillo para el procesado y compatibilidad con el set de datos.

Esto aplica para cuando se inserta un archivo, siendo cuando es en un uso en vivo, se realiza el preprocesamiento cada frame que avanza y al detectar un rostro, lo recorta y aplica el mismo procedimiento explicado anteriormente.

Implementación del modelo usado

Posterior al procesamiento se pasa al modelo de red neuronal convolucional (CNN) el cual se lleva principalmente en el archivo nombrado “trainer.py”, primero se importan librerías necesarias necesarias como Keras y TensorFlow, luego de ello se inicializa un modelo secuencial el cual permite agregar capas una a una.

La estructura del modelo se construye de forma secuencial y es compilado a través de las capas que realiza la normaización de píxeles de 0-255 a 0-1, en donde se introducen variaciones como la rotación, zoom y volteo.

Posteriormente comienza el entrenamiento del modelo el cual es entrenado desde las carpetas train/ y test/, junto a los callbacks para TensorBoard y ModelCheckpoint, y luego de ello comienza la elección aleatoria del conjunto de prueba y a partir de esas predicciones se visualizan las predicciones en donde se muestra en pantalla la etiqueta real y se imprimen las probabilidades de cada emoción posible.

Evaluación del modelo con métricas y análisis de resultados

La evaluación del modelo se lleva a cabo en el archivo evaluar\_modelo.py, en donde es una evaluación cuantitativa la cual mide el accurracy, precisión, recall y F1-score por cada clase para determinar la confiabilidad de los valores, el cual es utilizado recibiendo el modelo y datos de prueba preprocesados y convierte sus etiquetas one-hot en etiquetas de clase y de esa forma predice los resultados y genera las métricas para luego imprimir la matriz de confusión como mapa de calor.

También se encuentra el archivo graficar\_resultado.pý, cuyo objetivo es el que analiza el comportamiento del modelo durante el entrenamiento, mostrando la curva de pérdida vs epochs y la curva de precisión vs epochs, siendo alimentado por el objeto history por model.fit(), ayudando de esta manera la detección del overfitting y el bajo aprendizaje del modelo usado.

Adicionalmente se encuentra la evaluación visual post-entrenamiento, en donde dentro del método train() se realiza la toma de un lote de imágenes para la prueba y empiezan las pruebas con imágenes aleatorias junto a su etiqueta real y las probabilidades de predicción para últimamente imprimir las probabilidades en consola en el siguiente formato:

Probabilidades por emoción:

angry: 12.34%

happy: 72.10%

etc.

Por último se encuentra main.py, el cual se encarga de la evaluación en tiempo real y la predicción frame por frame, haciendo posible que se capturen los frames a utilizar para el procesamiento de imagen.

Arquitectura de la solución

**Frontend:**index.php: Contiene la estructura de la página principal.

Camara\_predict.html: Activa la cámara, captura imágenes y lanza predicciones.

Predecir\_ajax.php: Recibe las imágenes y las envía al modelo Python.

Subir\_imasgen.php y procesar\_imagen.php: Dedicadas para cuando el usuario decida subir fotos de forma manual.

**Backend:**

Trainer.py: Encargado del entrenamiento del modelo CNN con FER2013.

Evaluar\_modelo.py: Encargado de la evaluación de resultados.

graficar\_resultado.py: Graficación de resultados.

Predecir emoción.py: Se encarga de la predicción de emociones según los píxeles.

Predecir\_ajax.py: realiza inferencias con imágenes.

Entrenar\_modelo.py: Permite el entrenamiento manual.

Casos de uso

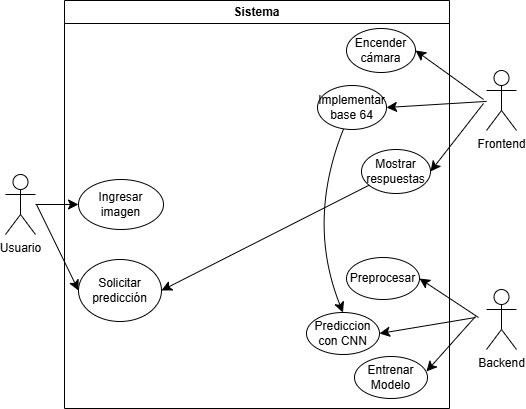
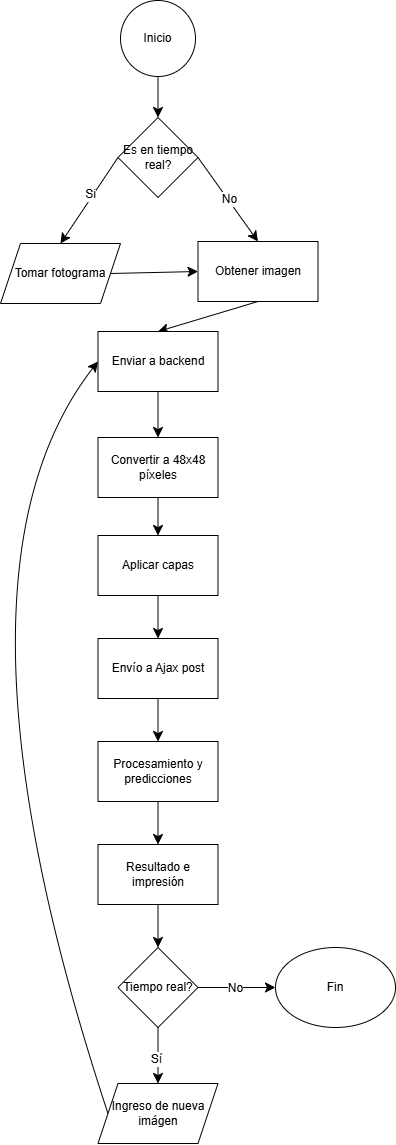
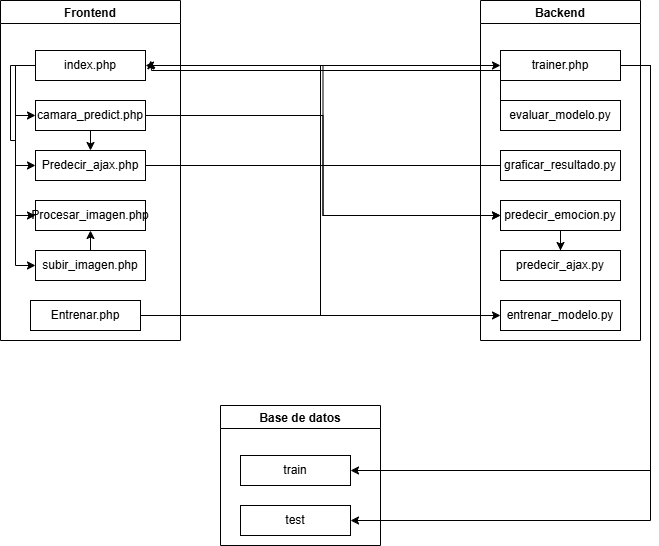


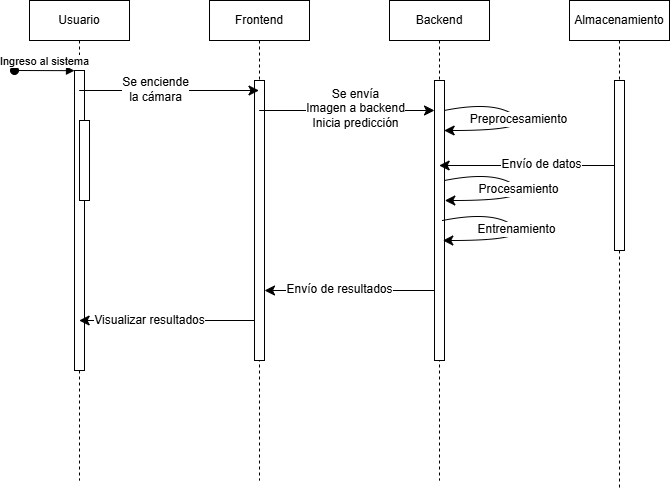
Diagrama de flujo general



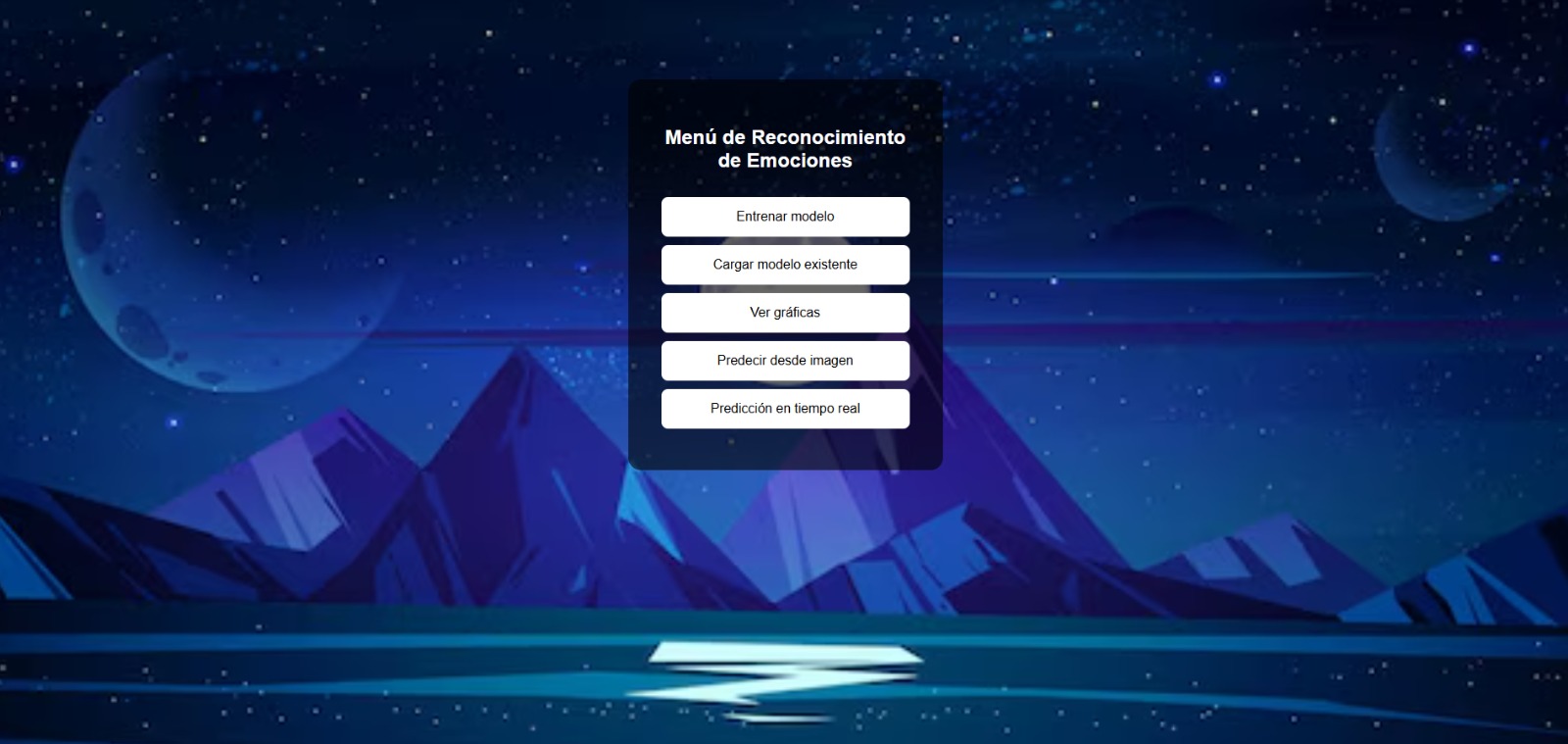
Componentes

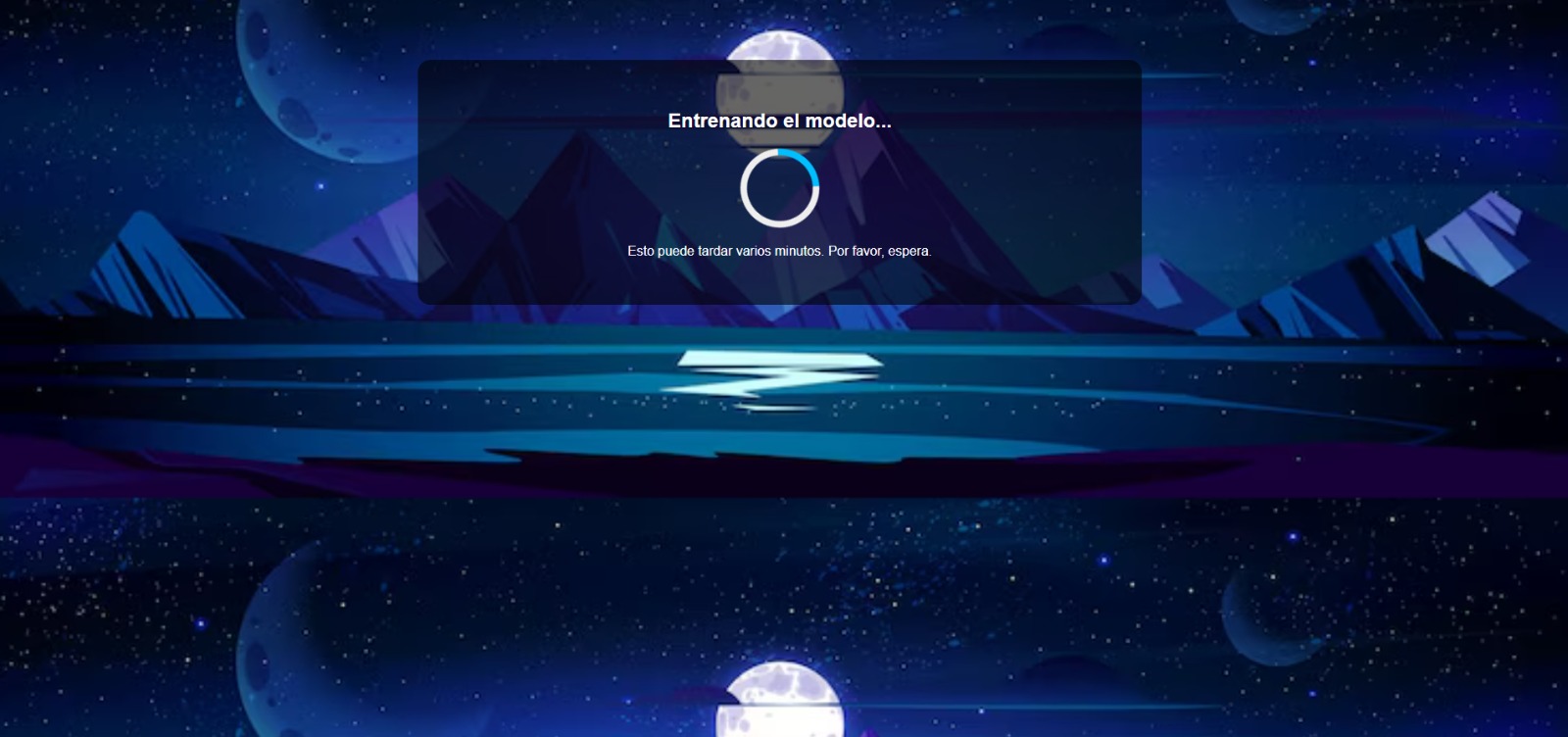


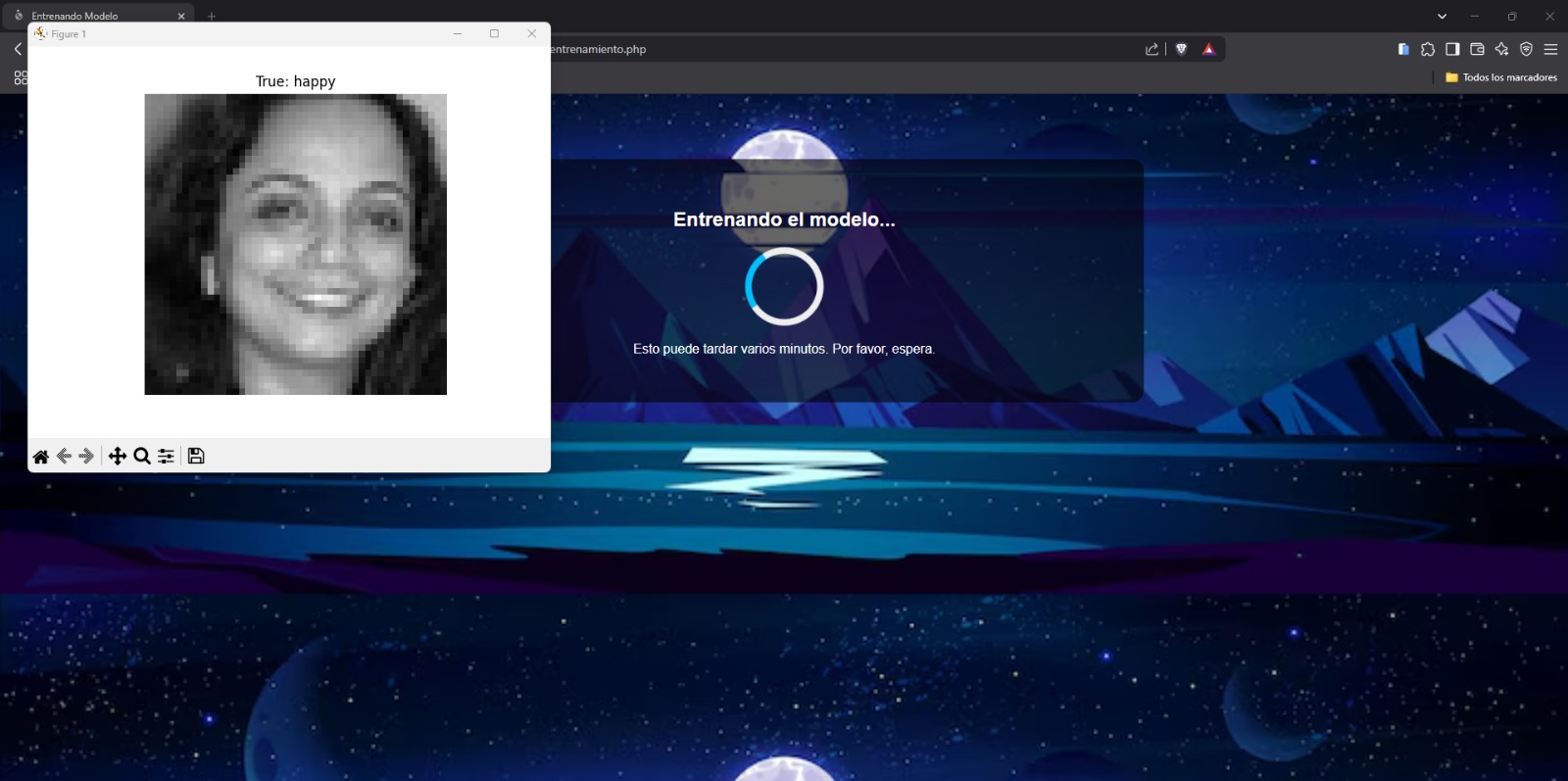
Secuencia

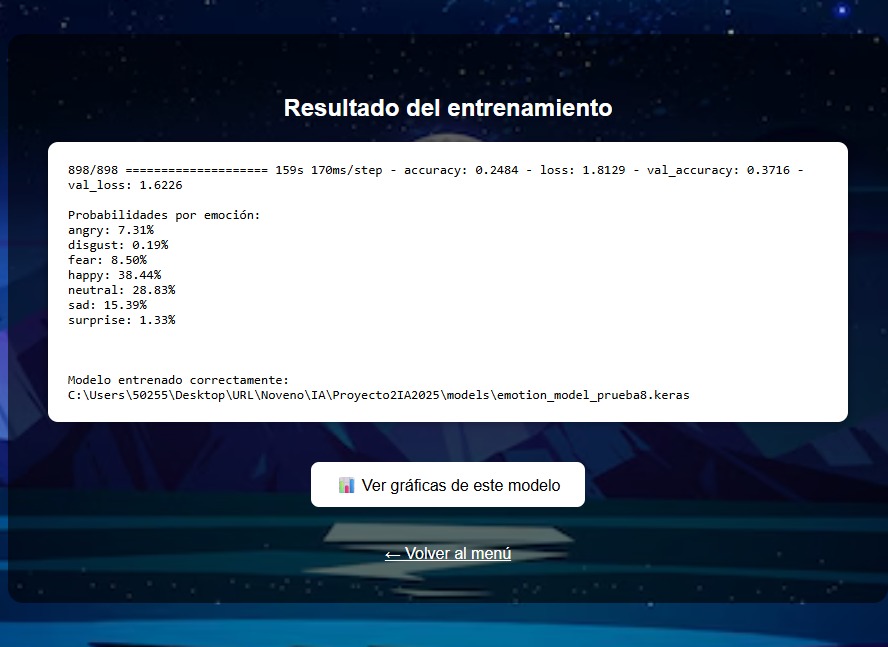


Evidencias de funcionamiento





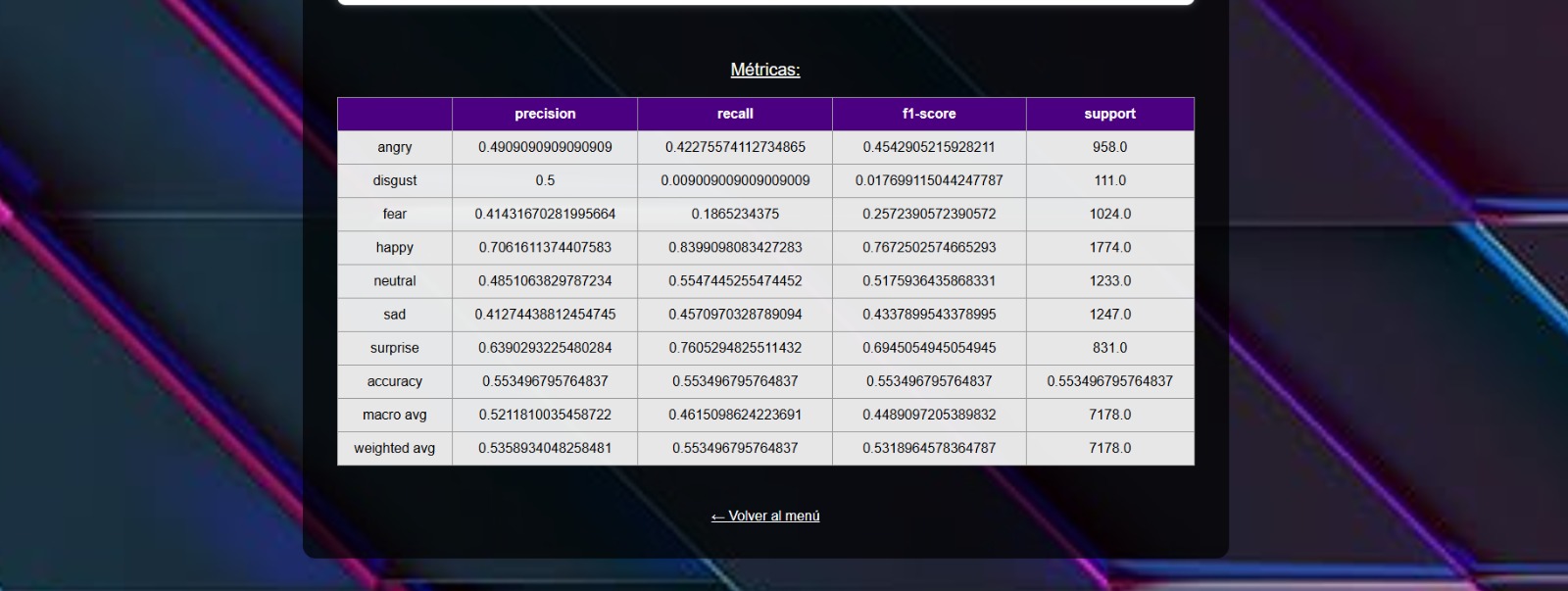












Conclusión

Las conclusiones que obtuvimos de este proyecto es en como funcionan los distintos modelos y la importancia de la consistencia en los sets de datos para unos resultados correctos y apropiados al momento de tomar las distintas métricas a evaluar. Esto además de lo aprendido y aplicado como las distintas capas para la correcta obtención de datos por medio de las distintas convoluciones.

Por lo que en conclusión final es importante la normalización de datos y la identificación correcta de los parámetros para la siguiente comparación.