Programación Web 3

UNLaM - Tecnicatura en Desarrollo Web

Trabajo Práctico de Investigación

*Reconocimiento de emociones en .NET*

[**Integrantes**](#_heading=h.30j0zll) **[1](#_heading=h.30j0zll)**

[**Objetivo**](#_heading=h.3znysh7) **1**

[**Situación Actual**](#_heading=h.2et92p0) **2**

[**Desarrollo de la Investigación**](#_heading=h.tyjcwt) **2**

[**Conclusiones**](#_heading=h.3dy6vkm) **2**

# 

# Integrantes

*<Listado de integrantes del equipo>*

# Objetivo

*<Descripción del trabajo práctico de investigación. La descripción debe incluir el tema a desarrollar y el punto de análisis sobre el que se trabajará>*

El objetivo de este trabajo práctico es desarrollar una **aplicación en .NET que utilice inteligencia artificial (IA)** para el reconocimiento de emociones faciales. Este proyecto tiene como finalidad explorar y analizar las **capacidades y limitaciones de los algoritmos de IA** para el procesamiento y análisis de imágenes faciales con el fin de identificar y clasificar diferentes emociones del rostro en base al modelo que se entrenará. Además, se evaluará la **eficacia de un modelo entrenado**, así como su grado de **precisión** y se le permitirá al usuario probar esta funcionalidad con una fotografía tomada en el momento.

El punto de análisis principal de este trabajo será la precisión y eficacia del modelo de reconocimiento de emociones faciales desarrollado en .NET. Se investigarán diversos factores que influyen en el rendimiento del modelo, tales como:

* **Calidad y tamaño del dataset** utilizado para el entrenamiento.
* **Robustez** del modelo frente a variaciones en las condiciones de iluminación y poses faciales.
* Tiempo de **respuesta y rendimiento** en tiempo real de la aplicación.

# Situación Actual

<Descripción de la situación actual de la tecnología, framework o concepto en relación al desarrollo de software utilizando tecnologías dotnet.>

**Introducción a .NET y su Aplicación en Inteligencia Artificial**

.NET es un marco de desarrollo de software creado por **Microsoft** que ofrece un entorno unificado para construir aplicaciones en **múltiples plataformas y lenguajes** de programación. Desde su lanzamiento, .NET se ha convertido en una opción popular debido a su flexibilidad, rendimiento y amplio conjunto de herramientas.

La integración de inteligencia artificial (IA) en aplicaciones .NET se ha vuelto cada vez más importante en diversos campos. Con bibliotecas como **ML.NET y TensorFlow.NET**, los desarrolladores pueden implementar modelos de **machine learning y deep learning** directamente en sus aplicaciones .NET. Esto abre nuevas oportunidades para **automatizar tareas, analizar datos y crear experiencias** de usuario más personalizadas y sofisticadas.

Al aprovechar las capacidades de IA en .NET, las organizaciones pueden mejorar la **eficiencia operativa**, descubrir conocimientos ocultos en **grandes volúmenes de d**atos y ofrecer productos y servicios **más inteligentes y adaptables** a las necesidades del usuario. En resumen, la combinación de .NET e IA representa un poderoso conjunto de herramientas para el desarrollo de aplicaciones modernas e innovadoras.

En el desarrollo de nuestra aplicación de reconocimiento de emociones faciales utilizando inteligencia artificial, ya avanzamos significativamente en varias **áreas clave**. A continuación, se detalla el estado actual del proyecto, incluyendo los logros alcanzados, las herramientas empleadas y las **próximas etapas** de desarrollo.

**1. Herramientas y Tecnologías Utilizadas**

**.NET:**

Hemos utilizado el marco de trabajo .NET, que proporciona un entorno de desarrollo robusto y escalable para la creación de aplicaciones de alta calidad. .NET ha sido fundamental para la integración de componentes y servicios, facilitando la implementación de nuestro modelo de IA en la aplicación.

**Visual Studio:**

Visual Studio ha sido nuestro entorno de desarrollo integrado (IDE) principal, proporcionando un conjunto completo de herramientas para codificación, depuración y pruebas. Su integración con Git y otras herramientas de control de versiones ha mejorado nuestra eficiencia en la gestión del código y la colaboración en el equipo.

**GitHub:**

GitHub ha sido nuestra plataforma principal para el control de versiones y la colaboración en el desarrollo del proyecto. Utilizando Git, hemos gestionado eficientemente el código fuente, facilitado la colaboración entre los miembros del equipo, y mantenido un historial detallado de los cambios realizados. Además, GitHub Actions nos ha permitido automatizar procesos de integración continua y despliegue.

**ML.NET:**

Se ha utilizado ML.NET, una biblioteca de machine learning de código abierto para .NET, que facilita la integración de modelos de aprendizaje automático en aplicaciones .NET. ML.NET nos ha permitido entrenar y evaluar modelos de manera eficiente, integrándolos directamente en nuestra aplicación.

**TensorFlow:**

TensorFlow se ha empleado para la creación y entrenamiento de modelos de aprendizaje profundo. Aprovechamos su capacidad para manejar grandes volúmenes de datos y su robustez en la implementación de redes neuronales convolucionales (CNN) para el reconocimiento de emociones faciales.

**2. Hitos Alcanzados (en orden cronológico)**

**Modelo de Reconocimiento de Emociones Entrenado:**

Completamos el entrenamiento del modelo de reconocimiento de emociones faciales, utilizando datasets bien establecidos. El modelo ha mostrado una precisión satisfactoria en la identificación de diferentes emociones, tales como felicidad, tristeza, enojo, sorpresa y neutralidad.

**Interfaz Visual Mínima:**

Se ha desarrollado una interfaz de usuario mínima que permite a los usuarios cargar imágenes y obtener resultados del reconocimiento de emociones. La interfaz es funcional y proporciona una experiencia básica para la interacción con el modelo.

**Integración del Modelo:**

El modelo entrenado ha sido integrado exitosamente en la aplicación .NET, permitiendo el procesamiento de imágenes y la clasificación de emociones en tiempo real.

**3. Próximas Etapas de Desarrollo (futuros hitos alcanzados)**

**Afinamiento de la Interfaz de Usuario**:

Mejoraremos la interfaz de usuario para que sea más intuitiva y visualmente atractiva. Esto incluye la optimización del diseño, la mejora de la experiencia del usuario y la incorporación de elementos visuales que faciliten la interpretación de los resultados.

**Generación de Frases Correspondientes a las Emociones Detectadas (memes):**

Añadiremos una característica que generará frases correspondientes a las emociones detectadas. Por ejemplo, si el modelo reconoce una expresión de felicidad, la aplicación podría mostrar una frase como "Cuando el profe te tira flores en la revisión del trabajo".

Esta funcionalidad requerirá la implementación de un sistema de mapeo entre emociones y frases predefinidas, asegurando que las respuestas sean contextualmente apropiadas y enriquecedoras para la experiencia del usuario.

**Clasificación del Reconocimiento a manos del usuario:**

Implementaremos funcionalidades adicionales para clasificar el reconocimiento de emociones de manera más detallada. Queremos implementar un sistema de “rating” para poder presentar las frases que más gusten a los usuarios en base a su emoción.

**4. Desafíos y Consideraciones generales**

**Optimización del Rendimiento:**

A medida que añadimos nuevas funcionalidades, es crucial garantizar que el rendimiento de la aplicación no se vea afectado.

**Precisión y Robustez del Modelo:**

Continuaremos afinando el modelo para mejorar su precisión y robustez frente a variaciones en las condiciones de iluminación, poses faciales y expresiones complejas.

**Consideraciones Éticas y de Privacidad:**

Con las posibles mejoras y evolución de las ideas del proyecto es muy posible que surjan dudas en base a la manipulación de los datos de los usuarios, siendo un tema delicado debido a que la función principal de la aplicación requiere del rostro del mismo

Hasta ahora, el desarrollo de nuestra aplicación de reconocimiento de emociones faciales ha progresado de manera significativa, logrando un modelo funcional y una interfaz de usuario básica. Con la próxima fase de afinamiento y adición de funcionalidades, estamos en una posición sólida para mejorar aún más la experiencia del usuario y la precisión del reconocimiento emocional. Nos enfocaremos en asegurar que cada nueva característica se integre de manera eficiente. Optimizando tanto el rendimiento técnico como la satisfacción del usuario

# Desarrollo de la Investigación

<Contenido de la investigación realizada incluyendo citas, links, código de ejemplo consistente con el objetivo definido.>

**Diferencias entre Deep Learning y Machine Learning y su Aplicación en el Proyecto**

**Deep Learning vs. Machine Learning**

**Machine Learning (ML):**

Se centra en algoritmos que pueden **aprender de datos y hacer predicciones** o tomar decisiones basadas en esos datos.

Utiliza técnicas de **aprendizaje supervisado, no supervisado y por refuerzo**.

Requiere la **extracción manual** de características de los datos.

Adecuado para problemas con **conjuntos de datos más pequeños y características bien definidas.**

**Deep Learning (DL):**

Es una subcategoría de ML que se basa en redes neuronales profundas con múltiples capas ocultas.

Puede aprender **automáticamente a extraer características de los datos**, eliminando la necesidad de extracción manual.

Es altamente efectivo para **problemas complejos y con grandes volúmenes de datos**.

**Aplicación en el Proyecto**

En nuestro proyecto de reconocimiento de emociones faciales, hemos optado por utilizar **Machine Learning**, específicamente técnicas de Machine Learning tradicionales, debido a las siguientes razones:

**Naturaleza del Problema:**

El reconocimiento de emociones faciales puede ser abordado eficazmente con técnicas de Machine Learning tradicionales, especialmente cuando se trata de clasificación de imágenes y reconocimiento de patrones visuales.

**Disponibilidad de Datos:**

Contamos con conjuntos de datos etiquetados adecuados para el entrenamiento de modelos de Machine Learning, lo que nos permite desarrollar y evaluar modelos con precisión satisfactoria.

**Eficiencia y Tiempo de Entrenamiento:**

Los modelos de Machine Learning tradicionales pueden entrenarse de manera eficiente y son menos intensivos en recursos computacionales en comparación con los modelos de Deep Learning. Esto nos permite desarrollar y desplegar modelos de manera más rápida y con menos recursos.

**Interpretabilidad:**

Los modelos de Machine Learning tradicionales suelen ser más interpretables que los modelos de Deep Learning, lo que significa que podemos entender mejor cómo se toman las decisiones y explicar el funcionamiento del sistema a los usuarios y partes interesadas.

En resumen, al utilizar técnicas de **Machine Learning** **tradicionales** en nuestro proyecto, podemos desarrollar un sistema de reconocimiento de emociones faciales **efectivo y eficiente**, que proporciona resultados **precisos e interpretables**, adecuados para nuestras necesidades y recursos disponibles.

A continuación, enumeramos de manera muy general las tareas a realizar (o realizadas) para la formación y crecimiento del proyecto:

**1. Investigación sobre técnicas de reconocimiento de emociones faciales:**

* Análisis de métodos tradicionales vs. métodos basados en aprendizaje profundo. (En nuestro caso fuimos por métodos tradicionales debido a la sencillez de aplicación)
* Revisión de datasets disponibles para el entrenamiento de modelos de reconocimiento de emociones.

2. **Desarrollo de la aplicación en .NET:**

* Diseño de la arquitectura del sistema.
* Implementación de la interfaz de usuario.
* Integración de bibliotecas y frameworks de IA compatibles con .NET (ML.NET, TensorFlow.NET).

3. **Entrenamiento y evaluación del modelo:**

* Preprocesamiento de datos y etiquetado de emociones (felicidad, sorpresa, tristeza, etc).
* Entrenamiento del modelo utilizando nuestro dataset específico (carpetas clasificadas por emoción con miles de imágenes dentro).
* Evaluación del rendimiento del modelo (porcentaje de reconocimiento exitoso)

4. **Análisis de resultados y optimización:**

* Interpretación de los resultados obtenidos.
* Identificación de posibles mejoras en el modelo y ajustes en el dataset.
* Comparación del rendimiento en base a nuestro objetivo.

# Conclusiones

<Conclusión de la investigación realizada en relación al objetivo definido>

# Referencias/Bibliografía

<Listado de referencias bibliográficas para el desarrollo del contenido del trabajo de investigación>

<https://learn.microsoft.com/es-es/dotnet/>

<https://learn.microsoft.com/en-us/dotnet/machine-learning/>

<https://www.tensorflow.org/?hl=es>

<https://aws.amazon.com/es/compare/the-difference-between-machine-learning-and-deep-learning/>

<https://chatgpt.com>