

Universidad Nacional Del Altiplano

Facultad De Ingeniería Mecánica Eléctrica, Electrónica Y

Sistemas

Escuela Profesional De Ingeniería De Sistemas



PLANIFICACIÓN DE “SISTEMA DE RECONOCIMIENTO DE

IMÁGENES”

CURSO:

ALGORITMOS Y ESTRUCTURAS DE DATOS

DOCENTE:

MG. ALDO HERNAN ZANABRIA GALVEZ.

INTEGRANTES:

CODIGO: 216984

FECHA: 28/04/2025

SEMESTRE:

I. Introducción

En la era digital actual, el procesamiento y análisis de imágenes ha adquirido una relevancia sin precedentes, siendo aplicado en múltiples campos como la seguridad, la salud, la industria y la automatización. El reconocimiento de imágenes, una rama fundamental de la inteligencia artificial y la visión por computadora, permite a los sistemas identificar y clasificar objetos, rostros, señales u otros elementos visuales dentro de imágenes digitales. Gracias a los avances en algoritmos de aprendizaje profundo (deep learning) y el uso de bibliotecas como TensorFlow, Keras y OpenCV, es posible desarrollar soluciones precisas y eficientes que imitan la percepción visual humana.

Este proyecto tiene como objetivo diseñar e implementar un software en Python capaz de realizar reconocimiento de imágenes utilizando modelos de redes neuronales convolucionales (CNN). A través de este desarrollo, se busca demostrar la aplicabilidad de estas tecnologías en la resolución de problemas reales, brindando una herramienta funcional que automatice tareas visuales complejas y reduzca la intervención humana.

II. Objetivo General

Desarrollar una aplicación de software en Python capaz de reconocer y clasificar objetos o rostros en imágenes utilizando técnicas de visión por computadora e inteligencia artificial. El sistema permitirá procesar imágenes desde archivos locales o en tiempo real mediante una cámara, brindando resultados precisos y rápidos.

III. Alcance del Proyecto

El sistema estará enfocado en el reconocimiento de imágenes para tareas específicas como clasificación de objetos, reconocimiento facial, o lectura de caracteres. Permitirá cargar imágenes desde un archivo o capturar en tiempo real. El usuario podrá visualizar los resultados del reconocimiento en una interfaz gráfica sencilla.

IV. Planteamiento del Problema

En la actualidad, muchas organizaciones enfrentan dificultades para analizar grandes volúmenes de imágenes de manera eficiente. El reconocimiento manual de elementos visuales es costoso, propenso a errores humanos y poco escalable. Por ello, se plantea la necesidad de un sistema automatizado de reconocimiento de imágenes que facilite la detección precisa de patrones visuales con el apoyo de algoritmos de aprendizaje profundo..

V. Herramientas y Tecnologías

- Lenguaje de programación: Python 3.x
- Librerías: OpenCV, TensorFlow, Keras, NumPy, Pandas, scikit-learn, Matplotlib

- Entornos de desarrollo: Goggle colab, Pycharm, VS Code
- Framework de interfaz: Tkinter o Streamlit
- Formato de entrega: Instalador ejecutable o aplicación web local

VI. Fases del Proyecto y Cronograma

Fase	Tareas Principales	Herramientas	Duración Estimada
1. Planificación y análisis	Definición del problema, recopilación de requisitos, análisis del dominio	Word, Notion, Reuniones	3 días
2. Investigación técnica	Estudio de librerías, revisión de modelos existentes (CNN, MobileNet, VGG16)	Documentación oficial, Jupyter	3 días
3. Recolección y preparación de datos	Descargar datasets, etiquetar datos, preprocesamiento (resize, normalización)	OpenCV, NumPy, Pandas	7 días
4. Desarrollo del modelo de reconocimiento	Diseño y entrenamiento de un modelo CNN o uso de uno preentrenado	TensorFlow / Keras	7-10 días
5. Desarrollo de la interfaz de usuario	Diseño de interfaz gráfica (GUI) para cargar imágenes y mostrar resultados	Tkinter / Streamlit / Flask	5 días
6. Pruebas y validación	Pruebas funcionales, métricas de rendimiento (accuracy, precision, recall)	scikit-learn, Matplotlib	4 días
7. Documentación y presentación final	Redacción de manual de usuario, informe técnico, presentación de resultados	Word, PowerPoint	3 días

VII. Casos de Uso Potenciales

- Reconocimiento facial para autenticación de usuarios.
- Clasificación de objetos en imágenes para inventario o monitoreo.
- Detección de señales de tránsito para asistencia a conductores.
- Imágenes de radiodiagnóstico.
- OCR para lectura automática de documentos o matrículas.

VIII. Temática del Proyecto

El presente proyecto se enfoca en el desarrollo de un software de reconocimiento de imágenes, que emplea técnicas de inteligencia artificial y aprendizaje profundo (deep learning) para identificar y clasificar objetos o rostros en imágenes digitales. El enfoque estará centrado en aplicaciones prácticas como seguridad, automatización y análisis visual.

IX. Metodología de Desarrollo

Para el desarrollo del software se utilizará la metodología ágil SCRUM. Este enfoque promueve el trabajo colaborativo, iterativo y basado en entregas parciales y funcionales. El proyecto será dividido en Sprints de una semana, cada uno con objetivos claros, seguimiento mediante reuniones diarias (Daily Scrum) y revisión al finalizar cada ciclo (Sprint Review).

X. Asignación de Tareas a los Miembros del Equipo

- **Product Owner:** Encargado de definir los requerimientos y prioridades del proyecto.
- **Scrum Master:** Responsable de coordinar el equipo, facilitar las ceremonias SCRUM y eliminar obstáculos.
- **Desarrollador 1:** Responsable del preprocesamiento de datos y entrenamiento del modelo CNN.
- **Desarrollador 2:** Responsable de la implementación de la interfaz de usuario e integración del modelo.
- **Tester:** Encargado de realizar pruebas funcionales y validar el rendimiento del sistema.
- **Documentador:** Responsable de la documentación técnica y del informe final.

XI. Resultados Esperados

Al finalizar el proyecto, se espera contar con un software funcional que sea capaz de reconocer con precisión objetos o rostros en imágenes proporcionadas por el usuario, y que ofrezca una interfaz fácil de usar. El modelo deberá alcanzar un nivel de precisión adecuado según los datos utilizados y el caso de uso definido.

XII. Bibliografía Recomendada

- *Building a Python Image Recognition System* / Cloudinary. (2025, 2 mayo). Cloudinary.
<https://cloudinary.com/guides/image-effects/building-a-python-image-recognition-system>
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly.
- Chollet, F. (2017). Deep Learning with Python. Manning Publications.
- Sitios web: tensorflow.org, opencv.org, keras.io