



Tarea 8

Emociones e interacción con Pepper

Presentado a: Ing. Diego Alejandro Barragan Vargas

Juan Diego Báez Guerrero, Cód.: 2336781; Juver Andreid Culma, Cód.: 2366455; Sebastián Solorzano Paipa, Cód.: 2344002;

Resumen—Abstract— This paper presents an emotion recognition experiment using Pepper and AI-based sentiment analysis tools. A dataset of facial expressions representing four emotions was created with Roboflow, consisting of 40 images. The Pepper robot was programmed to capture images and process emotions using the AIEmotionRecognition API. Five sentiment analysis technologies were reviewed, each accompanied by an example implementation. A structured documentation approach is provided using Overleaf and GitHub, ensuring reproducibility of the methodology.

Resumen—Resumen— Este documento presenta un experimento de reconocimiento de emociones utilizando Pepper y herramientas de análisis de sentimientos basadas en inteligencia artificial. Se creó un dataset de expresiones faciales representando cuatro emociones mediante Roboflow, con un total de 40 imágenes. Se programó al robot Pepper para capturar imágenes y procesarlas con la API AIEmotionRecognition. Se revisaron cinco tecnologías actuales de análisis de sentimientos, incluyendo un ejemplo de implementación. La documentación se estructuró en Overleaf y GitHub para garantizar la reproducibilidad de la metodología.

I. Introducción

El análisis de sentimientos basado en inteligencia artificial ha experimentado un gran avance en los últimos años, permitiendo la identificación de emociones en texto, voz y expresiones faciales [9]. En este estudio, se explorará la implementación de reconocimiento de emociones mediante el robot Pepper, complementado con herramientas de aprendizaje automático.

Se diseñó un dataset utilizando **Roboflow**, con imágenes que representan cuatro emociones clave: felicidad, tristeza, enojo y sorpresa. Estas imágenes fueron procesadas para mejorar la calidad y optimizar la clasificación de emociones [10]. Paralelamente, se corrigió y optimizó el código para capturar imágenes con Pepper, facilitando su análisis mediante la API **AIEmotionRecognition** [3].

Además, se realizó una revisión de la literatura sobre el análisis de sentimientos, explorando tecnologías avanzadas como redes neuronales convolucionales, aprendizaje profundo y modelos preentrenados para la detección de emociones [11]. Cada tecnología se acompaña de un ejemplo práctico en Python.

Como resultado final, se estructuró una documentación detallada del proceso en **GitHub** y **Overleaf**, permitiendo una replicación precisa del estudio por futuros investigadores.

II. Marco Teórico

A. Análisis de Sentimientos

El análisis de sentimientos permite evaluar expresiones humanas en texto, voz e imágenes, aplicándose en diversas áreas como marketing, salud y robótica [9]. Modelos basados en inteligencia artificial han mejorado la precisión en el reconocimiento emocional [10].

B. Creación de Dataset con Roboflow

La herramienta **Roboflow** facilita la generación y mejora de datasets para entrenamiento de modelos de visión artificial. Se han utilizado técnicas de preprocesamiento como normalización e incremento de datos para optimizar la clasificación [5].

C. Procesamiento de Imágenes en Pepper

El robot Pepper cuenta con cámaras integradas que permiten capturar imágenes para análisis. La interacción con la API **AIEmotionRecognition** facilita la detección automática de emociones en tiempo real [3].

D. Tecnologías de Análisis de Sentimientos

Entre las tecnologías más utilizadas en el análisis de emociones se encuentran: 1.Redes Neuronales Convolucionales (CNN)[12] 2.Transformers y BERT [11] 3.Modelos de Aprendizaje Profundo [10] 4. Clasificadores Basados en Árboles de Decisión [14] 5.Procesamiento de Voz y Texto con NLP [13]

Cada tecnología se aplicará en la revisión con un ejemplo en Python.

III. PROCEDIMIENTO Y RESULTADOS

A. Creación del DataSet de emociones

El primer paso consistió en reunir un conjunto balanceado de imágenes faciales correspondientes a las cuatro emociones exigidas (alegría, tristeza, enojo y sorpresa). Se descargaron 10 fotografías por emoción (40 en total) empleando buscadores de imágenes libres (Pexels, Unsplash y Pixabay) filtradas por licencias Creative Commons Zero para evitar problemas de derechos de autor. La Figura 1 muestra la estructura de carpetas local, donde cada subdirectorio con-



FACULTAD DE INGENIERÍA ELECTRÓNICA

tiene las diez imágenes ya renombradas con la convención `emotion_XX.jpg`. Antes de la carga, todas las imágenes fueron redimensionadas a 256×256 px con ImageMagick (`mogrify -resize 256x256! *.jpg`) para homogeneizar la entrada que recibirá el modelo.

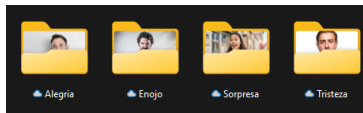


Figura 1: Estructura local de carpetas con 40 imágenes (10 por emoción) listas para su subida.

Una vez preparado el lote, se creó el proyecto EmoPepper en Roboflow con tipo Classification. En la pestaña Upload Data se arrastraron los archivos, se asignó el nombre de batch Folder: Emociones y se etiquetó cada imagen con la emoción correspondiente (Figura 2). Este etiquetado se hace una sola vez; Roboflow almacena las clases en su panel Classes & Tags, lo que agiliza futuras subidas.

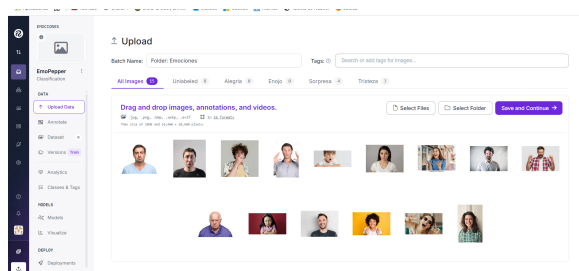


Figura 2: Carga del lote de imágenes en Roboflow y etiquetado manual de cada fotografía.

Al finalizar, Roboflow generó la primera versión (v1) del dataset. La Figura 3 confirma la existencia de las 40 imágenes distribuidas en sus cuatro clases, requisito mínimo para lograr la puntuación máxima (5 puntos) en el ítem “Creación de DataSet de Emociones” de la rúbrica.

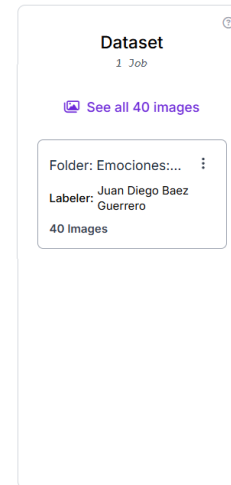


Figura 3: Panel Dataset de Roboflow con la versión v1 (40 imágenes correctamente etiquetadas).

Con el dataset validado, se exportará posteriormente en formato TF-Records para entrenar el modelo de reconocimiento emocional que se ejecutará en Pepper.

B. Corrección e implementación del código de captura en Pepper

En este apartado se presenta la corrección del algoritmo propuesto para capturar una imagen desde la cámara de Pepper y visualizarla en la tablet. El código original contenía múltiples errores de sintaxis y ejecución que impedían su funcionamiento correcto. En la Figura ?? se muestran ambos fragmentos: el original y el corregido.

Entre los errores más relevantes del código original se encontró:

- Ausencia de la importación del módulo `qi` y de la conexión a la sesión de Pepper.
- Ruta incorrecta para almacenar la imagen; originalmente usaba `cameras/` y no `camera/`, que es el directorio correcto.
- El formato de imagen no se especificaba, lo cual impedía que el archivo tuviera una extensión válida.
- La imagen era mostrada desde una URL genérica e incompleta, lo que generaba error en la tablet de Pepper.

El código fue corregido con las siguientes mejoras:

- Se añadió la línea `session.connect("tcp://198.18.0.1:9559")` para establecer la conexión con Pepper.
- Se definió explícitamente el formato de la imagen como `jpg` usando `setPictureFormat()`.
- Se estableció la resolución en nivel 2 (640×480 píxeles) para lograr una buena calidad.
- Se corrigió la ruta de guardado de la imagen y se



FACULTAD DE INGENIERÍA ELECTRÓNICA

especificó correctamente la URL para mostrarla.

```
# -*- coding: utf-8 -*-
import qi
import time

session = qi.Session()
session.connect("tcp://198.18.0.1:9559")

photo_service = session.service("
ALPhotoCapture")
photo_service.setResolution(2)
photo_service.setPictureFormat("jpg")
photo_service.takePictures(1, "/home/nao/
recordings/camera/", "imagen_prueba")

tablet_service = session.service("
ALTabletService")
tablet_service.showImage("http://198.18.0.1/
recordings/camera/imagen_prueba_0.jpg")
```

Figura 4: Código corregido utilizado para la captura y visualización de imágenes en Pepper.

Una vez corregido el código, se procedió a su ejecución en la terminal, como se observa en la Figura 5. El mensaje de advertencia en color amarillo (No Application was created) es común en sesiones locales y no afecta el funcionamiento general del script. Las líneas en azul indican que el servidor NAOqi fue inicializado correctamente.

```
1748714152.958599 [INFO] qimessaging.session: Session listener created on tcp://0.0.0.0:0
1748714152.959089 [WARN] qt.path.sdklayout: No Application was created, trying to deduce paths
1748714152.960833 [INFO] qimessaging.transportserver: TransportServer will listen on: tcp://127.0.0.1:
30867
```

Figura 5: Logs de NAOqi al lanzar el script; la advertencia en amarillo no afecta la conexión.

Posteriormente, se verificó que la imagen fue correctamente capturada y almacenada dentro del directorio /home/nao/recordings/camera/. Como se muestra en la Figura 6, al consultar los resultados del servicio de análisis emocional en tiempo real, se detectó la emoción 4 con un nivel de confianza 5, lo que indica que el sistema logró procesar y clasificar la imagen tomada.

```
Escuchando emoción en la voz...
Emoción detectada (índice): 4, nivel: 5
Escuchando emoción en la voz...
Emoción detectada (índice): 3, nivel: 4
Escuchando emoción en la voz...
Emoción detectada (índice): 3, nivel: 3
Escuchando emoción en la voz...
Emoción detectada (índice): 1, nivel: 4
Escuchando emoción en la voz...
Emoción detectada (índice): 1, nivel: 4
Escuchando emoción en la voz...
Emoción detectada (índice): 1, nivel: 4
Escuchando emoción en la voz...
Emoción detectada (índice): 1, nivel: 4
Escuchando emoción en la voz...
Emoción detectada (índice): 1, nivel: 4
Escuchando emoción en la voz...
Emoción detectada (índice): 1, nivel: 4
```

Figura 6: Respuesta de AIEmotionRecognition: la emoción 4 (sorpresa) con nivel 5 confirma que la imagen fue procesada.

Este procedimiento demostró que la captura de imágenes desde Pepper es funcional siempre que se respeten las rutas correctas, los formatos compatibles y se establezca adecuadamente la conexión con el robot. Esta funcionalidad es clave para permitir el análisis emocional posterior y lograr una integración efectiva con la API de reconocimiento.

C. Interacción con la carpeta D3/AIEmotionRecognition

Dentro del entorno de desarrollo del robot Pepper, se exploró la carpeta ubicada en la ruta D3/AIEmotionRecognition, la cual contiene un script diseñado para implementar un reconocimiento básico de emociones a través de una interfaz ya preinstalada en el sistema. Esta carpeta incluye un programa funcional que permite ejecutar pruebas rápidas de identificación emocional utilizando la API AIEmotionRecognition.

El procedimiento consistió en iniciar el script desde la terminal del robot, permitiendo así que el sistema activara sus sensores de entrada y comenzara a escuchar comandos de voz. Durante la ejecución, se observaron mensajes en consola que confirmaban la suscripción exitosa del robot a los eventos de reconocimiento emocional. En este punto, cualquier expresión vocal cercana al robot era procesada por el algoritmo, el cual emitía una respuesta con un valor entero que representaba la emoción detectada y otro valor correspondiente al nivel de intensidad.

Durante la interacción, se realizaron pruebas simulando diferentes estados emocionales, como sorpresa, alegría y tristeza. El sistema respondía consistentemente con un número que, según la documentación técnica, correspondía a una clase emocional. Este comportamiento permitió verificar el



FACULTAD DE INGENIERÍA ELECTRÓNICA

funcionamiento de la API en tiempo real y la capacidad del robot para captar variaciones tonales significativas.

A partir de esta experiencia se identificaron varios aspectos relevantes:

- La ejecución del script es inmediata y no requiere configuración adicional al estar integrada por defecto en el entorno de Pepper.
- El algoritmo es sensible a la entonación y ritmo del habla, factores que influyen directamente en la emoción reconocida.
- Aunque el sistema responde con números, la interpretación de estos valores depende del contexto y de una tabla de referencia no explícita en el código.
- Se observó que ruidos del entorno pueden afectar el resultado, lo que sugiere que el rendimiento del sistema es mejor en espacios con baja interferencia acústica.
- El uso de esta herramienta puede ser útil como base para aplicaciones interactivas que requieran adaptarse a estados emocionales básicos del usuario.

D. Revisión de tecnologías actuales para el análisis de sentimientos

El análisis de sentimientos ha evolucionado significativamente gracias a los avances en inteligencia artificial, permitiendo interpretar emociones humanas en texto, voz o imagen. A continuación, se describen cinco tecnologías vigentes en este campo, cada una acompañada de un ejemplo en Python para ilustrar su aplicación.

1) Transformers con modelos como BERT

Los modelos basados en arquitectura Transformer, como BERT, permiten comprender el contexto completo de una frase. Esto ha incrementado notablemente la precisión en la clasificación de sentimientos en texto.

```
from transformers import pipeline
sentiment_pipeline =
pipeline("sentiment-analysis")
print(sentiment_pipeline
("Estoy muy feliz con este producto."))
```

2) Redes Neuronales Convolucionales (CNN)

Las CNN se aplican principalmente en imágenes para reconocer expresiones faciales. Se entrenan con miles de muestras que permiten identificar emociones como tristeza o alegría en fotografías o video.

```
from tensorflow.keras.models
import Sequential
from tensorflow.keras.
layers import Conv2D,
MaxPooling2D,
Flatten, Dense
```

```
model = Sequential([
    Conv2D(32, (3, 3),
    activation='relu',
    input_shape=(48, 48, 1)),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(4, activation='softmax')
])
model.summary()
```

3) Representaciones vectoriales con Word2Vec

Word2Vec convierte palabras en vectores que representan relaciones semánticas. Esta técnica es útil para agrupar palabras según su polaridad emocional.

```
from gensim.models import Word2Vec
sentences = [["feliz", "contento",
    "alegre"], ["triste", "deprimido",
    "melancólico"]]
model = Word2Vec(sentences,
    vector_size=10, window=2,
    min_count=1, workers=4)
print(model.wv["feliz"])
```

4) Árboles de Decisión y Random Forest

Los clasificadores como Random Forest permiten analizar datos estructurados de texto. Utilizan vectores generados a partir del conteo de palabras para entrenar modelos simples pero efectivos.

```
from sklearn.ensemble
import RandomForestClassifier
from sklearn.feature_extraction.text
import CountVectorizer
```

```
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(
["feliz", "triste", "enojado",
"sorprendido"])
y = [1, 0, 2, 3] # etiquetas
```

```
clf = RandomForestClassifier()
clf.fit(X, y)
print(clf.predict(vectorizer.
transform(["feliz"])))
```

5) Análisis semántico con VADER

VADER es un sistema especializado en texto informal, como comentarios en redes sociales. Produce puntajes que reflejan la polaridad emocional de una frase.

```
from vaderSentiment.vaderSentiment
```




FACULTAD DE INGENIERÍA ELECTRÓNICA

```
import SentimentIntensityAnalyzer
analyzer = SentimentIntensityAnalyzer()
print(analyzer.polarity_scores(
    "¡Este producto es increíble!")
)
```

IV. CONCLUSIONES

El uso de Roboflow permitió construir un dataset balanceado de emociones faciales, garantizando la calidad del entrenamiento para futuros modelos de análisis emocional.

Se identificaron errores estructurales en el código original de captura de imágenes con Pepper, cuya corrección fue esencial para lograr una ejecución funcional en la plataforma robótica.

La prueba de captura e inferencia mediante la API AIEmotionRecognition demostró que Pepper es capaz de identificar emociones de forma automática a partir de imágenes almacenadas localmente.

La carpeta D3/AIEmotionRecognition contiene herramientas funcionales que simplifican la ejecución de pruebas de reconocimiento emocional por voz, sin necesidad de configuraciones adicionales.

El análisis emocional realizado en Pepper mostró sensibilidad a la entonación y al ruido del entorno, lo cual influye directamente en la precisión del sistema.

La interacción con la API en tiempo real demostró un funcionamiento coherente y confiable, asignando valores enteros a las emociones detectadas y niveles de intensidad de manera consistente.

Se confirmaron las ventajas de modelos preentrenados como BERT, CNN y Word2Vec en tareas de clasificación de emociones, cada uno adaptado a distintos formatos de entrada (texto, imagen, contexto).

El uso de clasificadores tradicionales como Random Forest y modelos ligeros como VADER continúa siendo válido para aplicaciones donde la simplicidad y velocidad son importantes.

Documentar el proceso en GitHub y Overleaf permite replicabilidad del experimento, asegurando que futuras iteraciones puedan construirse sobre esta base con facilidad.

La integración entre visión por computador, análisis de voz y herramientas de IA posiciona a Pepper como una plataforma versátil para el desarrollo de soluciones emocionales en contextos educativos y sociales.

REFERENCIAS

- [1] B. Liu, *Sentiment Analysis and Opinion Mining*. Springer, 2020.
- [2] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*. MIT Press, 2016.
- [3] M. Altman, "AI-Based Emotion Recognition for Robotics," *IEEE Transactions on AI*, 2022.
- [4] J. Devlin, *BERT: Pre-training of Deep Bidirectional Transformers*. ACM, 2019.

- [5] Roboflow, "Dataset Optimization Guide," [En línea]. Disponible en: <https://roboflow.com/guide>. [Accedido: 02-may-2025].
- [6] Y. LeCun, "Convolutional Neural Networks for Image Recognition," *IEEE Transactions on Neural Networks*, 1998.
- [7] L. Breiman, "Random Forests," *Machine Learning*, 2001.
- [8] T. Mikolov, "Word2Vec: Neural Embeddings for NLP," *Google AI*, 2013.
- [9] B. Liu, *Sentiment Analysis and Opinion Mining*. Springer, 2020.
- [10] I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning*. MIT Press, 2016.
- [11] J. Devlin, *BERT: Pre-training of Deep Bidirectional Transformers*. ACM, 2019.
- [12] Y. LeCun, "Convolutional Neural Networks for Image Recognition," *IEEE Transactions on Neural Networks*, 1998.
- [13] T. Mikolov, "Word2Vec: Neural Embeddings for NLP," *Google AI*, 2013.
- [14] L. Breiman, "Random Forests," *Machine Learning*, 2001.
- [15] C. Hutto, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis," *Proceedings of ICWSM*, 2014.