

# Qué es ButterflAI?



#### **DETECTOR Y CLASIFICADOR:**

- Es una aplicación de detección y clasificación de mariposas a tiempo real.
- Con la capacidad de detectar hasta 100 clases de mariposas distintas y darte información de ellas.
- Con un Chatbot maravilloso.
- Hemos desarrollado dos vertientes:
  - Página web
  - Aplicación movil





## Desarrollo del proyecto

Planteamiento de la idea

Obtención de los datos

Limpieza y preparación de los datos

Entrenamiento de los modelos

Desarrollo de la web

Desarrollo de la app

NLP Pruebas



## Obtención de datos

DATASET

Butterfly & Moths Image Classification 100 species

- 13.594 imágenes
- 100 especies



Random Images for Image classification

- 10.000 imágenes
- Avión, vela, muñeco de nieves, etc.



## Scraping

#### **BUTTERFLY CONSERVATION**

Página inicial con los enlaces a las subpágina de cada tipo de mariposa.



En cada subpágina hay información de cada mariposa como el nombre científico, hábitat, etc.

Subida de datos a S3.

#### Adonis Blue

Polyommatus benargus

This beautiful species of butterfly is one of the most characteristic of unimproved southern chall downland, where it can be seen flying low over shortly grazed turf [typically speep, south-facing shorest.

have been twen brilliand sky blave wings, while the formales are chocolate brown and for less solcution. Both series have distinctive black lines that enter or cross the white tringes of the eings.

The white, textured discarbaged eggs are laid singly under young, unshaded Horseshire Vetch leaves in May June and August September. They can be found most easily in September where unshaded Horseshire Vetch is growing on short turf.

The Adon's titlur overwinters as a caterpiliar; it is is green with short, sellow stripes, which camouflage it while it feeds on Hosseshoe Vetch during the day. It is most correnonly seen durin April and late July as it searches for ants to milk its sugary secretions.

In April-May and July-August the caterpillar forms into a chrysalis in small crevices or hollows an is then buried by ants in earth chambers connected to the ant nest. The ants constantly attend





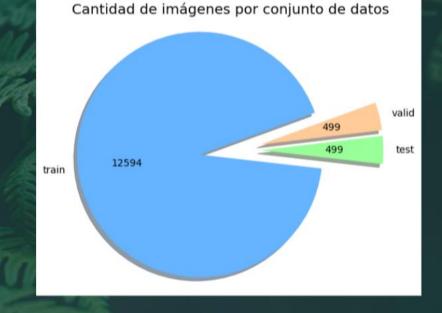




### Visualización de los datos

#### DATASET

|       | class id |    | filepaths                   | labels          | data set |  |  |
|-------|----------|----|-----------------------------|-----------------|----------|--|--|
|       | 0        | 0  | train/ADONIS/001.jpg        | ADONIS          | train    |  |  |
| NA PA | 1        | 0  | train/ADONIS/002.jpg        | ADONIS          | train    |  |  |
|       | 2        | 0  | train/ADONIS/003.jpg        | ADONIS          | train    |  |  |
|       | 3        | 0  | train/ADONIS/004.jpg        | ADONIS          | train    |  |  |
|       | 4        | 0  | train/ADONIS/005.jpg        | ADONIS          | train    |  |  |
|       |          |    |                             |                 |          |  |  |
|       | 13589    | 99 | valid/ZEBRA LONG WING/1.jpg | ZEBRA LONG WING | valid    |  |  |
|       | 13590    | 99 | valid/ZEBRA LONG WING/2.jpg | ZEBRA LONG WING | valid    |  |  |



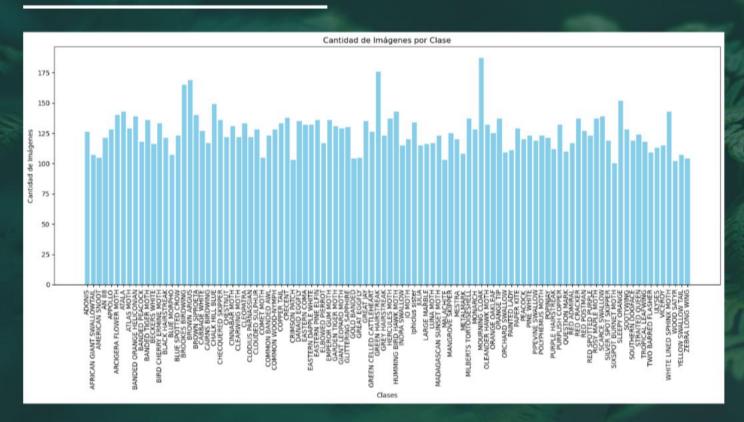
• Image: columna con las imágenes en array normalizadas [0, 1]

Sin valores nulos



## Visualización de los datos

#### REPARTO DE IMÁGENES



#### **Entrenamiento:**

- Max. → 187
- Min. → 100

#### Validación

• 5 por especie

#### Prueba

• 5 por especie



### Preparación de los datos

1. Separación en conjuntos.

train\_df = df[df['data set'] == 'train'][['class id', 'image']]

2. Mezcla de filas.

train\_df = train\_df.sample(frac=1).reset\_index(drop=True)

3. Separación en los conjuntos X e y.

X\_train, y\_train = train\_df['image'], train\_df['class id']

4. Conjunto y a One-hot

y\_train = tf.keras.utils.to\_categorical(y\_train, num\_classes=num\_classes)



### Creación del modelo

· Tamaño de entrada

Input((224, 224, 3)),

Capa de salida

Dense(100, activation='softmax')

→ Total de capas: 7

Primera capa

```
Conv2D(64, (3, 3), activation='relu', kernel_regularizer=regularizers.12(0.00005)), BatchNormalization(), MaxPooling2D(2, 2),
```

Capas ocultas

```
Conv2D(256, (3, 3), activation='relu', kernel_regularizer=regularizers.12(0.00005)), BatchNormalization(), MaxPooling2D(2, 2), Dropout(0.2),
```

Mostrar código



## Compilación del modelo

```
model.compile(
    optimizer=keras.optimizers.Adam(learning_rate=0.001),
    loss='categorical_crossentropy',
    metrics=['accuracy']
)
```



••••

## Data Augmentation

Rotación: 30°

Translación vertical: 20%

• Translación horizontal: 20%

• Cizallamiento: 20%

• Zoom: 20 %

Volteo horizontal

 Rellena los píxeles que faltan con los valores más cercanos.

Mostrar código



### Creación de Callbacks

Early Stopping

```
early_stopping = EarlyStopping(
    monitor='val_loss',
    patience=7,
    restore_best_weights=True
)
```

Reduce LR

```
reduce_lr = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.5,
    patience=3,
    min_lr=1e-6
)
```

### Entrenamiento del modelo

```
history = model.fit(
    datagen.flow(X_train, y_train, batch_size=64),
    epochs=100,
    batch_size=64,
    validation_data=(X_valid, y_valid),
    callbacks=[early_stopping, reduce_lr]
)
```

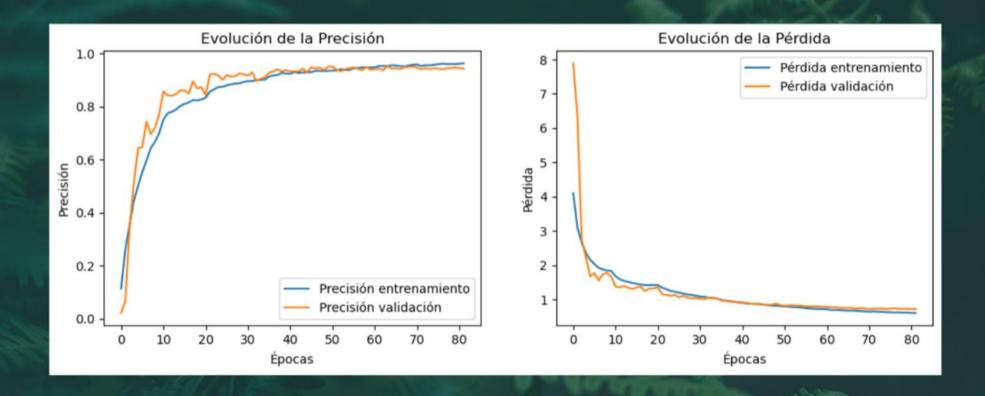
82 épocas

• Train: 0.9621

• Valid: 0.9420



### Métricas del rendimiento





•••

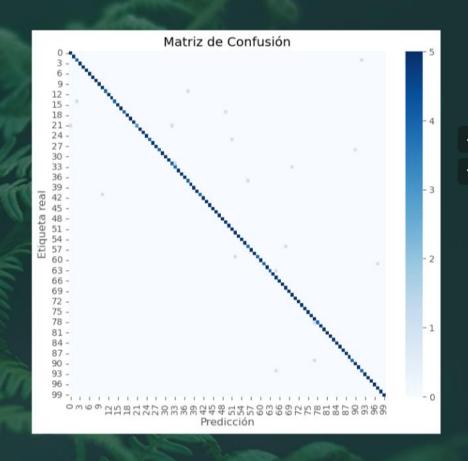
### Rendimiento del clasificador

• Precisión: 0.9620

Aciertos: 481

• Fallos: 19

| Especie                      | Precisión | Recall | F1-<br>Score | Soporte |
|------------------------------|-----------|--------|--------------|---------|
| ADONIS                       | 0.83      | 1.00   | 0.91         | 5       |
| AFRICAN GIANT<br>SWALLOWTAIL | 1.00      | 1.00   | 1.00         | 5       |
| AMERICAN SNOOT               | 0.80      | 0.80   | 0.80         | 5       |
| AN 88                        | 1.00      | 1.00   | 1.00         | 5       |
| APPOLLO                      | 1.00      | 1.00   | 1.00         | 5       |
|                              |           |        |              |         |
| accuracy                     |           |        | 0.96         | 500     |
| macro avg                    | 0.97      | 0.96   | 0.96         | 500     |
| weighted avg                 | 0.97      | 0.96   | 0.96         | 500     |







### Preparación de los datos

DATASET DE KAGGLE A YOLO

Usando el dataset de Kaggle que tiene las mariposas con sus tipos correspondientes.

Butterfly & Moths Image Classification 100 species

Data Card Code (109) Discussion (1) Suggestions (0)

Data Explorer Version 12 (478.64 MB)

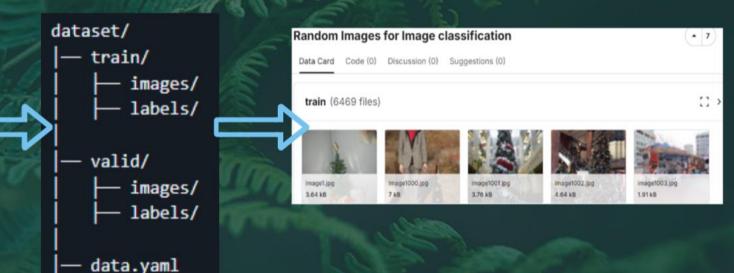
Data Explorer Version 12 (478.64 MB)

Data Explorer Version 12 (478.64 MB)

Data Explorer Oversion 13 (478.64 MB)

Usamos **Roboflow** para poder pasar los datos a formato **YOLO v8**.

Añadimos las imagenes de los **objetos aleatorios**.





## Fine-tuning

#### CONFIGURACIÓN DEL MODELO

- Al ser un modelo preentrenado no tenemos que crearlo desde cero.
- Usando el dataset que hemos creado y que está configurado en el data.yaml.
- Usamos la GPU para hacerlo lo más óptimo posible.
- Se genera una carpeta con todos los gráficos y el modelo.

Uso de la GPU

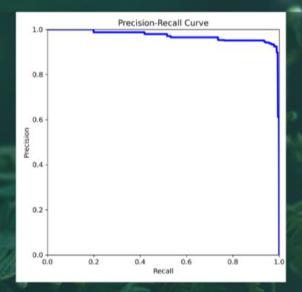
```
model.train(
    data="modificado_copy/data.yaml",
    epochs=50,
    imgsz=640,
    batch=8,
    project="butterfly_detection",
    name="yolov8_butterfly",
    device=0.
    half=True,
    lr0=1e-4.
    workers=8,
    augment=True,
    freeze=10,
    cos_lr=True,
    patience=10,
    val=True
```

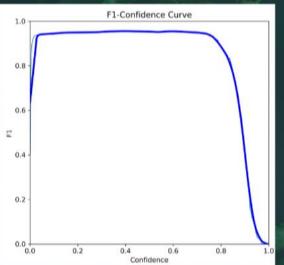
### Rendimiento del detector

GRÁFICA PRECISIÓN-RECALL CURVE

Estas gráficas muestran que el modelo tiene un buen rendimiento, con un F1-score de 0.96 y una alta precisión en la detección de mariposas.

GRÁFICA F1-CONFIDENCE CURVE





Precisión: 0.969

• F1-score: 0.96

Umbral: 0.395







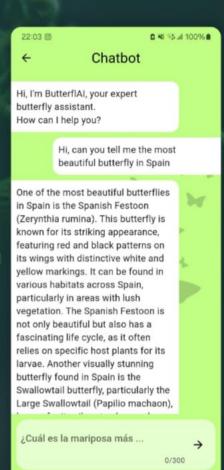
### NLP

#### PROCESAMIENTO DE LENGUAJE NATURAL

Para llevar a cabo este punto hemos realizado un Chatbot usando la API de ChatGPT.

Como añadido hemos puesto text-to-spech.







## Conclusiones

¿QUÉ HEMOS LOGRADO?

- Desarrollo de dos modelos de IA:
  - Detector de mariposas
  - Clasificador de mariposas

- Web en Agular
- App Android con Flutter

PLATAFORMAS

NLP

- Chatbot usando la API de ChatGPT
- Texto-to-spech para las respuestas del bot

NUESTRA OPINION



