# Object Detection for Estrous Staging (ODES) Guide

This script uses a pre-trained YOLOv8 model to detect and classify cells in images as Leukocyte, Cornified, or Nucleated. Based on the cell counts and ratios, it predicts the estrous stage (Estrus, Diestrus, Proestrus, or Metestrus) for each image.

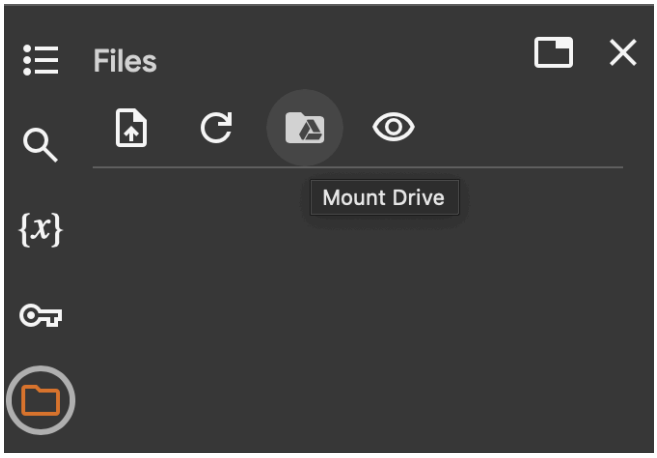
**Before Running ODES:**

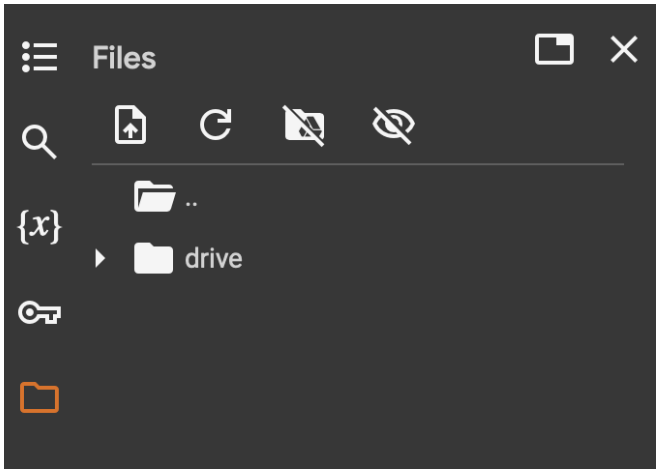
1. Upload the weight file to your Google Drive
2. Download/save the image directory to be used on your Google Drive

**How to Run ODES**

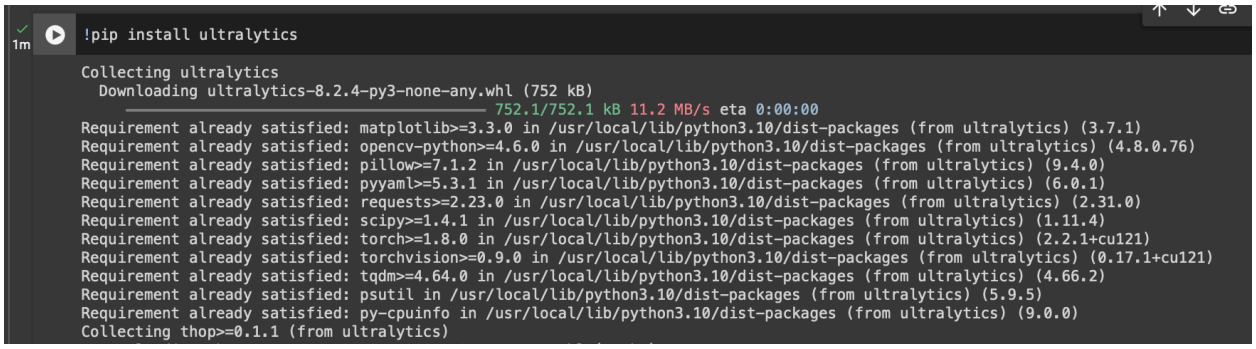
1. Open a Google Colab Notebook

2. **Mount Drive and Press "Connect to Google Drive**: Provides the notebook access to your Google Drive



2. **!pip install ultralytics**: Installs necessary packages to run the script. Wait until complete.



```
!pip install ultralytics

Collecting ultralytics
  Downloading ultralytics-8.2.4-py3-none-any.whl (752 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 752.1/752.1 kB 11.2 MB/s eta 0:00:00
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (3.7.1)
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.8.0.76)
Requirement already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (9.4.0)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (6.0.1)
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.31.0)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (1.11.4)
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (2.2.1+cu121)
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (0.17.1+cu121)
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.10/dist-packages (from ultralytics) (4.66.2)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from ultralytics) (5.9.5)
Requirement already satisfied: py-cpuinfo in /usr/local/lib/python3.10/dist-packages (from ultralytics) (9.0.0)
Collecting thop>=0.1.1 (from ultralytics)
```

3. **Run this script** and **enter the required paths** when prompted
(Required paths: path to the weight file in your drive and path to your image directory in your drive)

```python
# Load necessary libraries
from ultralytics import YOLO
import torch
import pandas as pd
import os
import time


# Get model path and images directory from user
model_path = input("Enter the path to the weight file (e.g.,
/content/drive/MyDrive/Folder/ODESweight.pt): ")
images_directory = input("Enter the path to the image directory (e.g.,
/content/drive/MyDrive/images): ")


# Load trained model
model = YOLO(model_path)

# Function to detect estrous stage
def determine_estrus_stage(boxes, names):

  class_indices = boxes.data[:, -1].cpu().to(dtype=torch.int32).numpy()
  total_cells = len(class_indices)
  leukocytes_count = sum(names[i] == "Leukocyte" for i in class_indices)
  cornified_count = sum(names[i] == "Cornified" for i in class_indices)
```

```
  nucleated_count = sum(names[i] == "Nucleated" for i in class_indices)

  # Calculate percentages
  leukocytes_percentage = (leukocytes_count / total_cells) * 100 if
total_cells else 0
  cornified_percentage = (cornified_count / total_cells) * 100 if
total_cells else 0
  nucleated_percentage = (nucleated_count / total_cells) * 100 if
total_cells else 0

  # Determine if low cell count
  is_low_cell_count = '***' if total_cells <= 35 else ''

  # Determine estrous stage based on cell counts and percentages
  if total_cells > 35:
      if cornified_percentage >= 80 or (cornified_percentage >= 70 and
leukocytes_percentage <= 10):
          predicted_stage = "Estrus"
      elif leukocytes_percentage >= 75 or (leukocytes_percentage >= 60 and
cornified_percentage <= 10):
          predicted_stage = "Diestrus"
      elif nucleated_percentage >= 35 or nucleated_percentage >= 70:
          predicted_stage = "Proestrus"
      else:
          predicted_stage = "Metestrus"
  else:  # For low cell count (<=35)
      if leukocytes_count < 5:
          if cornified_count > nucleated_count:
              predicted_stage = "Estrus"
          elif nucleated_count > cornified_count:
              predicted_stage = "Proestrus"
      elif leukocytes_count >= 5:
          if leukocytes_count > 0.7 * total_cells:
              predicted_stage = "Diestrus"
          elif nucleated_count >= 1 and cornified_count >= 1:
              predicted_stage = "Metestrus"

  return predicted_stage, total_cells, leukocytes_percentage,
cornified_percentage, nucleated_percentage, is_low_cell_count
```

```
def process_image_and_get_stage(image_path):
  result = model.predict(image_path, conf=0.25, max_det=1000, stream=True)
  for res in result:
      return determine_estrus_stage(res.boxes, res.names)


# List to store results
results = []


# Loop through each image in the directory and process
for image_name in os.listdir(images_directory):
  if image_name.lower().endswith(('.png', '.jpg', '.jpeg', '.bmp')):
      image_path = os.path.join(images_directory, image_name)
      (predicted_stage, total_cells, leukocytes_percentage,
cornified_percentage, nucleated_percentage, is_low_cell_count) =
process_image_and_get_stage(image_path)

      # Append result for current image with percentages and low cell
count indicator
      result = {
          'ImageName': image_name,
          'Total Cells': total_cells,
          'Leukocytes Percentage': f"{leukocytes_percentage:.2f}%",
          'Cornified Percentage': f"{cornified_percentage:.2f}%",
          'Nucleated Percentage': f"{nucleated_percentage:.2f}%",
          'Predicted Stage': predicted_stage,
          'Low Cell Count': is_low_cell_count
      }
      results.append(result)


      # Print result for current image with percentages and low cell count
indicator
      print(f"Processed: {image_name}, Predicted Stage: {predicted_stage},
Total Cells: {total_cells}, Leukocytes: {leukocytes_percentage:.2f}%,
Cornified: {cornified_percentage:.2f}%, Nucleated:
{nucleated_percentage:.2f}%, Low Cell Count: {is_low_cell_count}")
```

```
# Save the results after processing all images
df_results = pd.DataFrame(results)
df_results.to_csv('/content/results.csv', index=False)
```

4. The results will be saved as a results.csv under the “Files” section.
Make sure to download or save it on your Google Drive. (Note: If it doesn’t initially appear, press the refresh button to the left of the mount drive icon)